**DESIGN LAB REPORT**

**“Martian Parachute”**

**Spring 2021**

**PHYS 172 Modern Mechanics**

Lab Section 059

Group 03

Isaac Abram
Alexander Hardin
Alexandra Hasenour
Sarah Lanie
Samantha Thompson

March 31, 2021

## I. EXECUTIVE SUMMARY

NASA has contracted our team to design a parachute system containing a 207 kg rover and a 828 kg lander to safely enter Mars atmosphere and safely decelerate from entry velocities of 10,000 - 12,000 mph to speeds around 200 mph. The criteria include a virtual model showing the parachute systems' descent and a computational model in which the system decelerates to a safe speed of around 200 mph before it reaches Mars' surface. The constraint is a time limit of 1 month. Space exploration is of growing importance as scientists seek to gain new knowledge about foreign space objects, and our teams' system can help with the initial Mars' landing. Our proposed solution is a parachute added to the lander. Materials used for the parachute would be similar to materials used with prior space vehicles such as Pathfinder, Opportunity, and Spirit (Protective Fabric). Airbags for Mars have to be strong enough to cushion the space craft. They must also allow the spacecraft to bounce across the surface of Mars. In addition to this, the airbags are required to be inflated seconds before touchdown and must be deflated once safely on the ground. The Mars airbags are made from Vectran which is a synthetic material that was used on Mars Pathfinder (Mars exploration rover mission: The mission). Vectran performs better than other synthetic materials. The material needs to be lightweight and ultratough to adequately protect the vehicle from Mar's jagged surface. The material successfully dissipated the energy from a 45 mph impact. The rover used four airbags with six lobes each. The bag system must be flexible and responsive to group pressure. The fabric of the airbags is attached to the rover with these ropes. The ropes give the bags shape, which make inflation work better. Three gas generators are used for inflation.A parachute successfully fulfills our design requirements. We see this through our code. Once we add in the code for the parachute, the lander successfully slows down to speeds around 200 mph and has a steady descent. The code successfully complies, which means that we developed a computational model. Our team met the time frame of one month for completing the project. Finally, our virtual model has a descent similar to the computational model. A potential limitation of a parachute could be choosing wrong materials for the parachute strings. If the strings don't have the proper tension, they could break and the parachute would fail. Other limitations include not having prior experience with coding python. Having prior experience with coding python would have made coding something complicated like this lab easier and would have saved us time and effort, allowing us to further develop our design. Our final limitation is a limitation of supplies and testing methods. We can only test and develop our prototype through code. We don't have access to the supplies or time that would be needed to sufficiently test our prototype outside of coding.

**Contents**


EXECUTIVE SUMMARY **2**

TEAM MEMBER ROLES **4**

PROBLEM SCOPING **6**

- Problem Statement 6
- Design Requirements 6
- Design Stakeholders 6
- Assumptions about the Problem 7

SOLUTION GENERATING **7**

- Functional Decomposition 8
- Exploring Prior Art 8

PROTOTYPING **9**

ITERATION #1 **12**

ITERATION #2 **13**

FINAL DESIGN OVERVIEW **14**

REFERENCES **17**

APPENDICES **18**

SIGNATURES OF ALL PARTICIPATING STUDENTS **18**

## II. TEAM MEMBER ROLES

Our project team included 5 members (See Figure 1). We allocated roles and responsibilities to each team member as described in Table 1.

*Table 1. Team member contributions*

| **Project milestones** | **Isaac Abram** | **Alexander Hardin** | **Alexandra Hasenour** | **Sarah Lanie** | **Samantha Thompson** |
|---|---|---|---|---|---|
| Problem-scoping | Took the leadership role, shared his screen as well as wrote most of the answers to the problem scoping questions | Helped develop criteria and constraints | Helped with the problem statement formulation | Helped identify stakeholders and assumptions | Helped identify assumptions and stakeholders |
| Solution-generating | Helped with the functional decomposition generation | Took the leadership role, shared his screen as well as wrote most of the answer to the solution generating questions | Helped with the functional decomposition generation | Helped with the prior art generation | Helped with the prior art generation. Wrote the solution generation breakdown and created a corresponding table. |
| Prototyping | Helped with the mathematical theory behind the prototype | Helped with the mathematical theory behind the prototype | Took the leadership role, shared her screen as well as wrote the answers to most of the prototyping questions. | Helped with the actual code implementation | Helped with the actual code implementation |
| Iteration 1 | Helped find outside | Helped with the | Helped find outside | Took the leadership | Helped with the |

| | | | | | |
|---|---|---|---|---|---|
| | sources to develop our iteration involving air density | mathematical theory behind the iteration | sources to develop our iteration involving air density | role, shared her screen as well as wrote most of the code for iteration 1 | mathematical theory behind the iteration |
| Iteration 2 | Helped find outside sources to develop our iteration involving air density | Helped find outside sources to develop our iteration involving air density | Helped with the mathematical theory behind the iteration | Helped with the mathematical theory behind the iteration | Took the leadership role, shared her screen as well as wrote most of the code dealing with iteration 2 |
| Final report | Helped write parts of the final report | Helped write parts of the final report | Helped write parts of the final report | Helped write parts of the final report | Helped write parts of the final report |



*Figure 1. Project Team*

## III. PROBLEM SCOPING

### a. Problem Statement

NASA has contracted our team to design a parachute system containing a 207 kg rover and a 828 kg lander to safely enter Mars atmosphere and safely decelerate from entry velocities of 10,000 - 12,000 mph to speeds around 200 mph. The criteria include a virtual model showing the parachute systems' descent and a computational model in which the system decelerates to a safe speed of around 200 mph before it reaches Mars' surface. The constraint is a time limit of 1 month. Space exploration is of growing importance as scientists seek to gain new knowledge about foreign space objects and our teams' system can help with the initial Mars' landing.

### b. Design Requirements

*Table 2. Design Requirements*

| # | Criteria/Constraints | Metrics (units) |
|---|---|---|
| 1 | Enable safe and effective landing | Success vs. Failure (does it reach Mars' surface with a speed of less than 200 mph?) |
| 2 | Be completed within 1 month | Success vs. Failure (did the team meet the time frame?) |
| 3 | Develop computational model | Success vs. Failure (does the model's code compile?) |
| 4 | Virtual Model | Success vs. Failure (does the model show the system's descent similarly to how the computational model would predict?) |

The design requirements, both constraints and criteria, come with their inherent trade-offs that had to be managed and assessed throughout the design process. One trade-off we had to take into consideration was the quality of the design versus the time constraint; given we only had a month to develop a working solution that fit the requirements of the client, the quality of the design was less than it would have been had the deadline been further out. Another trade-off we were faced with was materials versus time, the team was given minimal materials to complete this project, which enabled us to effectively develop a final solution that was finished within the deadline and met the client's requirements.

### c. Design Stakeholders

*Table 3. Design Stakeholders*

| Client/User | Method | Questions | Answers |
|---|---|---|---|
| The European Space Agency | N/A - only talked with NASA | | |

| | | | |
|---|---|---|---|
| NASA | Weekly Meetings | -Air density equations and general approach for computational model incorporating air density<br>-If the teams' code quality met standards | -Directed the team to a NASA website documenting the Mars' air density equation<br><br>-Yes, our code met all standards |
| Roscosmos State Corporation | N/A - only talked with NASA | | |

**d. Assumptions about the Problem**

One of the assumptions the team needs to make is that we should ignore the time it takes the parachute to open because the time is insignificant in this case. Furthermore, the computational model will not have to account for spin, since the thrusters located on the lander will prevent rotation. Another assumption made was that the gravity on Mars is 3.711 m/s^2.

## IV. SOLUTION GENERATING

Our original solution generation consisted of multiple different ideas (listed below).

1. Parachute with longer strings
2. Parachute with different material
3. Parachute with a larger surface area
4. Parachute with a smaller surface area
5. Parachute with shorter strings
6. Parachute with thrusters
7. Two parachutes in parallel
8. Two parachutes in series
9. An airbag attached to the rover with a parachute

After our initial solution generation we were able to break the ideas down into four categories: simple parachute design (ideas 1-5), a system with a parachute and thrusters (idea 6), parachutes in parallel/series (idea 7&8), and an airbag system(idea 9).

*Table 4. Solution Generation and Breakdown*

| Category | Simple Parachute Design | Parachute and Thrusters | Parachutes in Parallel/Series | Airbag System |
|---|---|---|---|---|
| **System** | Rosalind<br>Kazachok<br>Parachute | Rosalind<br>Kazachok<br>Parachute<br>Thrusters | Rosalind<br>Kazachok<br>Parachutes | Rosalind<br>Kazachok<br>Parachute<br>Airbag |
| **Surroundings** | Mars | Mars | Mars | Mars |

| | | | | |
|---|---|---|---|---|
| **Principles of Physics** | Momentum Principle, Energy Principle, Position Update Equation, Kinetic Energy, Drag Force, Gravitational Potential Energy, Velocity Update Equation, Newton's 2nd Law, Work Equation. | Momentum Principle, Energy Principle, Position Update Equation, Kinetic Energy, Drag Force, Gravitational Potential Energy, Velocity Update Equation, Newton's 2nd Law, Work Equation. | Momentum Principle, Energy Principle, Position Update Equation, Kinetic Energy, Drag Force, Gravitational Potential Energy, Velocity Update Equation, Newton's 2nd Law, tension in parallel/series, Work Equation. | Momentum Principle, Energy Principle, Position Update Equation, Kinetic Energy, Drag Force, Gravitational Potential Energy, Velocity Update Equation, Newton's Second Law, Work Equation |

### a. Functional Decomposition

*Main function* *Sub-functions* *Design ideas that meet this function*

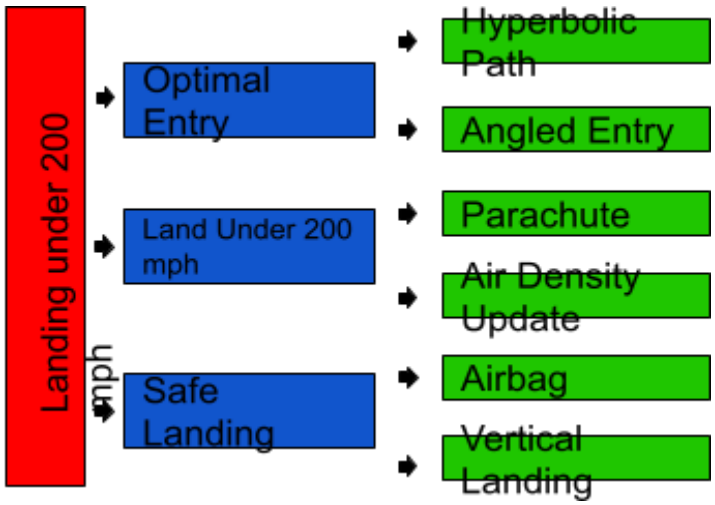
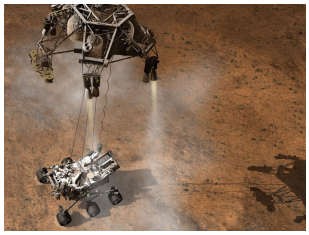



*Figure 2. Functional Decomposition Model*

### b. Exploring Prior Art

*Table 5. Generating ideas by exploring prior art*

| # | Name of the Prior Art | Strengths | Limitations | Adaptations |
|---|---|---|---|---|
| 1 | Perseverance (Khanna 2021)  | Used a combination of a parabolic and a linear decent onto the surface. For its landing sequence, it incorporates a | Landed in the Jezero Crater, which was a hazardous location with boulders. | We analyzed the angle that perseverance took into the atmosphere and adapted that into our code to |

| | | | | |
|---|---|---|---|---|
| | *Figure 3. Perseverance Art* | parachute and a powered descent.<br><br>(Berger 2021) | (Berger 2021) | achieve a parabolic landing shape. |
| 2 | Curiosity<br>(Wall 2013)<br><br>*Figure 4. Curiosity Art* | Incorporated a parachute and a powered landing descent to the surface of mars.<br><br>(Howell 2018) | The parachute could only slow the spacecraft down to 200 mph, which was too fast to land.<br><br>(Howell 2018) | We decided to adapt the parachute from the curiosity rover, however we had to make it bigger to slow the lander to under 200 pmh. |
| 3 | Spirit<br>(Atkinson 2010)<br><br>*Figure 5. Spirit Art* | Utilized both a parachute and airbags to help the lander to safely land on the ground.<br><br>(Howell 2018) | Lander was not able to come in at an angle into the atmosphere.<br><br>(Howell 2018) | We adapted the airbags from the spirit rover to help the lander to safely land on the ground. |

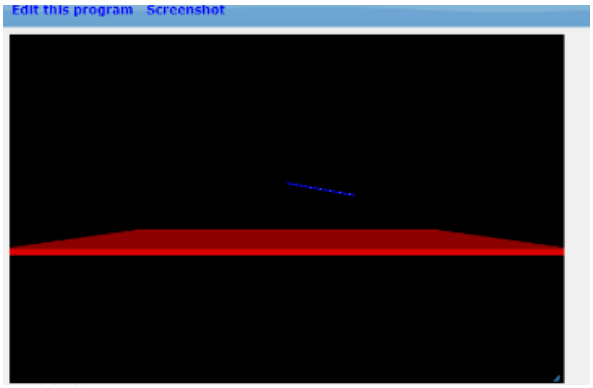
## V. PROTOTYPING

### a. Prototype Description

Attached in the Appendix is The Martian Challenge GlowScript Folder which contains three different versions of the code.

Our initial attempt at creating a system to decelerate from entry velocities to 200 mph at Mars' surface was very difficult and many problems arose, but through trial and error, our team eventually created a somewhat working model. We followed the general coding approach outlined to us from past Python coding experiences and implemented the momentum and position update equation to iteratively change the system's momentum and, thus, the system's position by calculating the net force as time increased.

Prototype 1 - A main problem in coding a solution was the breakdown of the net forces acting on the system. In our first prototype, our team calculated the net force by the force of gravity minus the force of drag. However, as seen by the figures below, the speed increased and the height of the system suddenly decreased by thousands of meters.




```
### Iterative Position Update
t = 0
dt = 0.01
Fg = vec(0, -M_tot * g_Mars, 0)
Speed = InitialSpeed
Height = InitialHeight
while Speed > 200*0.44704:
    ##The next four lines of code are called an "if-else" statement.
    ##The if-else operators allow the program to make decisions based on logical conditions.
    ##If-else operators follow the same indentation rules as the while loops.
    if Height < 10*1609.34 :
        Area = pi*8*8 # the effective cross-sectional area with the parachute open
    else :
        Area = pi*2*2 # the cross-sectional area of the system
    Fdrag = -norm(Lander.mom) * 0.5 * C_d * rho * Area * (Speed ** 2)
    Fnet = Fg - Fdrag
    Lander.mom = Lander.mom + Fnet * dt
    Lander.pos = Lander.pos + (Lander.mom / M_tot)*dt
    t = t + dt

    #The rate(n) command tells the computer to halt computation for 1/n seconds before proceeding.
    #This allows us to slow down the visual refresh rate of the simulation so that we can actually perceive the motion.
    rate(10000)

    Speed = mag(Lander.mom)/M_tot
    Height = Lander.pos.y
    plt1.plot(t, Height)
    plt2.plot(t, Speed)

###-------------------This is the end of the program-----------------------###
```

*Figure 6. GlowScript While Loop for the Initial Prototype 1*

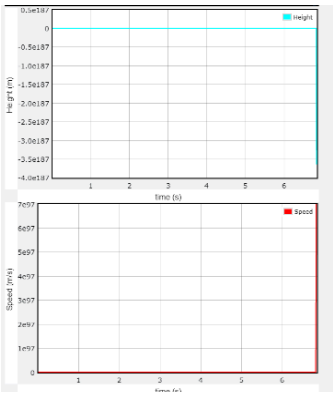



*Figure 7. Glowscript Simulation Output for Prototype 1*

Prototype 2 - For prototype 2, our team changed the calculation of the net force by adding the force of gravity and the drag force. However, the calculation of the drag force became a concern. Initially, the team decided to calculate the drag force through the equation Fdrag = -norm(Lander.mom) * 0.5 * C_d * rho * Area * (Speed ^ 2), but as seen in the below figures, this

equation created numerous problems. The system went below Mars' surface once again although the speed did begin to decrease, but after a certain amount of time the system's speed began to increase also thus creating similar problems to our initial prototype.

```
### Iterative Position Update
t = 0
dt = 0.01
Fg = vec(0, -M_tot * g_Mars, 0)
Speed = InitialSpeed
Height = InitialHeight
while Speed > 200*0.44704:
    ##The next four lines of code are called an "if-else" statement.
    ##The if-else operators allow the program to make decisions based on logical conditions.
    ##If-else operators follow the same indentation rules as the while loops.
    if Height < 10*1609.34 :
        Area = pi*8*8 # the effective cross-sectional area with the parachute open
    else :
        Area = pi*2*2 # the cross-sectional area of the system
    Fdrag = -norm(Lander.mom) * 0.5 * C_d * rho * Area * (Speed ^ 2)
    Fnet = Fg + Fdrag
    Lander.mom = Lander.mom + Fnet * dt
    Lander.pos = Lander.pos + (Lander.mom / M_tot)*dt
    t = t + dt

    #The rate(n) command tells the computer to halt computation for 1/n seconds before proceeding.
    #This allows us to slow down the visual refresh rate of the simulation so that we can actually perceive the motion.
    rate(10000)

    Speed = mag(Lander.mom)/M_tot
    Height = Lander.pos.y
    plt1.plot(t, Height)
    plt2.plot(t, Speed)

###-------------------This is the end of the program-----------------------###
```

*Figure 8. GlowScript While Loop for the Initial Prototype 2 (above)*

*Figure 9. Glowscript Simulation Output for Prototype 2 (below)*

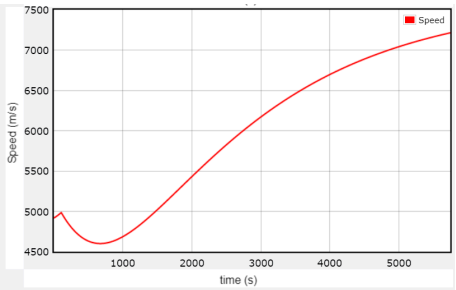
Prototype 3 - For our third prototype, we attempted to fix the drag force problem by using the equation Fdrag = -norm(Lander.mom) * 0.5 * C_d * rho * Area * (Speed ** 2) instead as well as calculating net force by adding force of gravity and drag force together. In this prototype, the system successfully decreases to a speed of below 200 mph, but the lander and rover never actually reach the ground. However, in this prototype, the system is the closest to reaching Mars' surface without actually going below the surface.

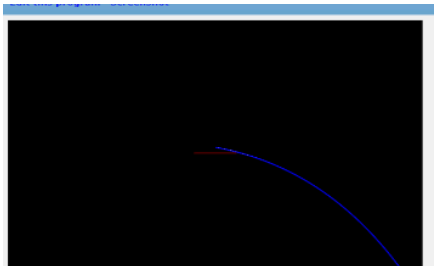
```
### Iterative Position Update
t = 0
dt = 0.01
Fg = vec(0, -M_tot * g_Mars, 0)
Speed = InitialSpeed
Height = InitialHeight
while Speed > 200*0.44704:
    ##The next four lines of code are called an "if-else" statement.
    ##The if-else operators allow the program to make decisions based on logical conditions.
    ##If-else operators follow the same indentation rules as the while loops.
    if Height < 10*1609.34 :
        Area = pi*8*8 # the effective cross-sectional area with the parachute open
    else :
        Area = pi*2*2 # the cross-sectional area of the system
    Fdrag = -norm(Lander.mom) * 0.5 * C_d * rho * Area * (Speed ** 2)
    Fnet = Fg + Fdrag
    Lander.mom = Lander.mom + Fnet * dt
    Lander.pos = Lander.pos + (Lander.mom / M_tot)*dt
    t = t + dt

    #The rate(n) command tells the computer to halt computation for 1/n seconds before proceeding.
    #This allows us to slow down the visual refresh rate of the simulation so that we can actually perceive the motion.
    rate(10000)

    Speed = mag(Lander.mom)/M_tot
    Height = Lander.pos.y
    plt1.plot(t, Height)
    plt2.plot(t, Speed)

###-------------------This is the end of the program-----------------------###
```



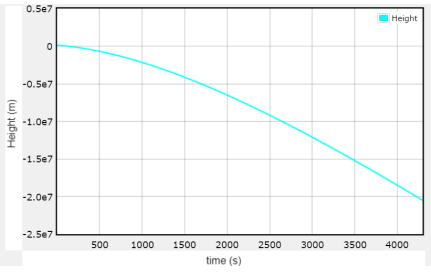
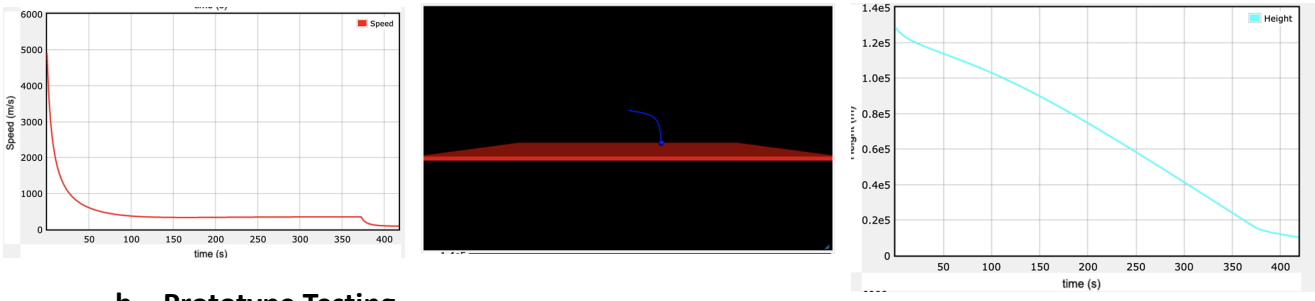


*Figure 10. GlowScript While Loop for the Prototype 3 (above)*

*Figure 11. Glowscript Simulation Output for Prototype 3 (below)*




### b. Prototype Testing

Success will be measured by how closely our computation model's system reached Mars' surface and if the system's speed was less than 200 mph at Mars' surface.

### c. Prototype Assessment

*Table 6. Prototype Assessment*

| Prototype Assessment | Height | Speed |
|---|---|---|
| Prototype 1 | -4.0 x 10 ^ 187 m | 7 x 10 ^ 97 m/s |
| Prototype 2 | -3.5 x 10 ^ 6 m | Lowest speed of 4600 m/s |
| Prototype 3 | 1.05 x 10 ^4 m | 416 m/s |

As seen in Table 6, prototype 3 for both height and speed testing most closely met prototype testing. Prototype 3's height is the closest to Mars' surface without going below the surface and its speed is the closest to 200 mph as well. As such, Prototype 3 most closely aligns with the criteria provided by the client, that being safely reaching the surface of Mars by decelerating to 200 mph prior to landing and displaying a virtual & computational model.
Our three initial prototypes can be seen in the folder below:

GlowScript IDE

## VI. ITERATION #1

### a. Revision

In our first iteration we used the suggestions of our lab instructor to ensure that our team considered all factors needed to execute the code. The main problem our team focused on with our first iteration was changing air density. As the altitude changes, so does the air density and the concept of changing air density was left out of the original code.

### b. Re-testing

We corrected this effectively by making air density a function of altitude and adding this within our while loop. The function we came up with rho =0.699*exp(-0.0009* Height) was inserted within the while loop to solve our issue. The results of the test can be seen below. After further inspection of our code we were confronted with additional issues, however, these issues were evaluated in later weeks. Both the speed and height are lower values than in the initial prototype, so the team's testing is more successful than the initial prototype. However, the system still does not reach Mars' surface.

### c. Assessment

The result of this iteration helped us to come closer to what a real simulation of a mars rover landing sequence would look like. As one can analyze in the graph below, the trajectory of the lander is at an angle as it enters the atmosphere, and continues on that path until it slows down enough to reach a vertical landing phase. This is similar to the landing trajectories of the most recent Mars rover landing, Perseverance. This is a major upgrade from our original prototype, which had a hyperbolic landing trajectory.

```
while Speed > 200*0.44704:
    ##The next four lines of code are called an "if-else" statement.
    ##The if-else operators allow the program to make decisions based on logical conditions.
    ##If-else operators follow the same indentation rules as the while loops.
    if Height < 10*1609.34 :
        Area = pi*8*8 # the effective cross-sectional area with the parachute open
    else :
        Area = pi*2*2 # the cross-sectional area of the system
    rho = 0.699 * exp(-0.00009 * Height)
    Fdrag = -norm(Lander.mom) * 0.5 * C_d * rho * Area * (Speed ** 2)
    Fnet = Fg + Fdrag
    Lander.mom = Lander.mom + Fnet * dt
    Lander.pos = Lander.pos + (Lander.mom / M_tot)*dt
    t = t + dt
```

*Figure 12. GlowScript While Loop (Changing Air Density Update)*

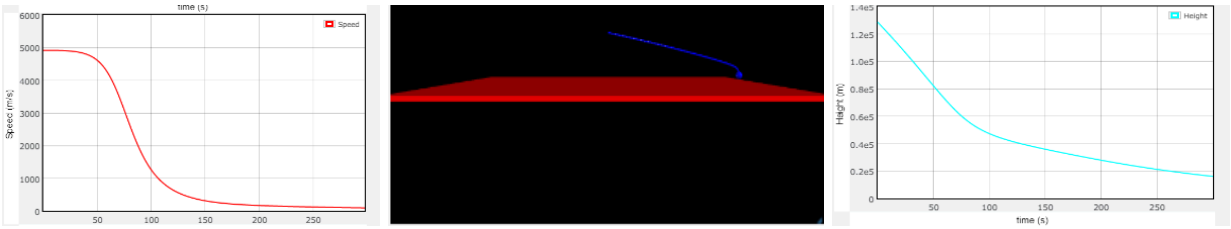



*Figure 13. GlowScript Simulation Output For Iteration 1*

## VII. ITERATION #2

### a. Revision

In this iteration we made significant changes in order to better reflect the necessary requirements of the provided by the client. In iteration two we tackled three primary problems to enhance our prototype and solution. The first problem we tackled was to add a more specific drag coefficient depending on if the parachute has been deployed or not. To address this, we found a study conducted by Georgia Tech which determined the drag coefficient for the lander and the drag coefficient for the parachute. The study stated that in Mars simulations, the parachute drag coefficient typically used is 0.5 and the average drag coefficient for the lander is

1.072. As such, we added the drag coefficient into the while loop in our GlowScript code, where we used just the coefficient for the lander (1.072) before the parachute was deployed and a combination of the two drag coefficients (1.072 + 0.5) for after deployment. In our previous iterations the land rover was not reaching the ground in the simulation, but would rather stop in mid air above the ground. Evidently, the rover not reaching the ground is unrealistic and does not meet the design requirements. In order to address this we changed our while loop to run when the height is greater than 0 rather than when the speed is greater than 200*0.4474. After making this technical change, the rover was able to reach the height of 0 in the simulation. The last problem we tackled within this iteration is the air density update. Updating the density of air throughout the simulation is crucial to conducting a realistic simulation to best reflect the environment/atmosphere on Mars. As such, we made air density a function of altitude within our while loop.

### b. Re-testing

Our final prototype perfectly meets the testing prototypes in the initial prototyping stage. The system reaches Mars' surface without going under the surface and the lander and rover's speed is below 200 mph when the system reaches Mars' surface as seen in the below figures.

### c. Assessment

The result of this iteration helped create a fully functioning prototype. As seen by the graphs below, the utilization of the changing drag coefficient combined with changing the while loop statement to height > 0 allows the system to reach a height of 0 m and a final speed of around 5 m/s.

```
t = 0
dt = 0.01
Fg = vec(0, -M_tot * g_Mars, 0)
Speed = InitialSpeed
Height = InitialHeight
while Height > 0:
    ##The next four lines of code are called an "if-else" statement.
    ##The if-else operators allow the program to make decisions based on logical conditions.
    ##If-else operators follow the same indentation rules as the while loops.
    if Height < 10*1609.34 :
        Area = pi*8*8 # the effective cross-sectional area with the parachute open
        C_d = .5 + 1.072 # drag coefficient with parachute
    else :
        Area = pi*2*2 # the cross-sectional area of the system
        C_d = 1.072 # drag coefficient of the lander
    rho = .699 * exp(-.00009 * Height) #air density update
    Fdrag = -norm(Lander.mom) * 0.5 * C_d * rho * Area * (Speed ** 2)
    Fnet = Fg + Fdrag
    Lander.mom = Lander.mom + Fnet * dt
    Lander.pos = Lander.pos + (Lander.mom / M_tot)*dt
    t = t + dt
```

*Figure 14. Mars Rover GlowScript Simulation While Loop For Iteration 2*

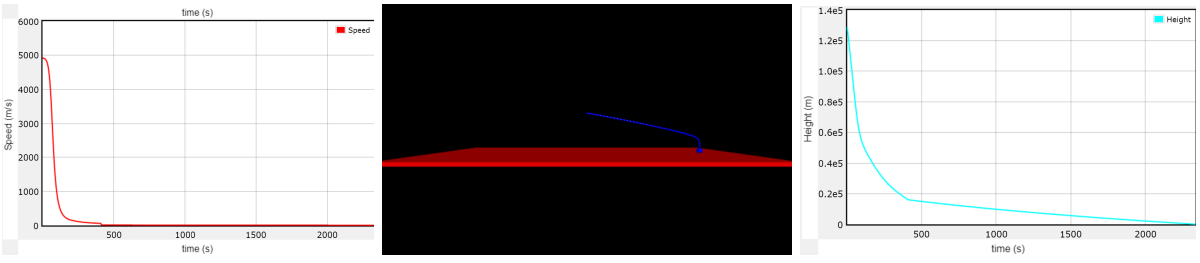



*Figure 15. Mars Rover Simulation Output For Iteration 2*

## VIII. FINAL DESIGN OVERVIEW

### a. Detailed Design

Our team was tasked by NASA with designing a computational model that would decelerate a Mars lander and rover from entry speeds of 10,000 to 12,000 mph to a speed of around 200 mph on impact with Mars' surface. Through idea generation, functional decomposition, and prior art, the team decided on a simple parachute model to slow the system's entry into Mars. Our initial computational model utilized the momentum update principle as well as the position update principle to calculate the position of the system through multiple iterations. While the system did meet the criteria of slowing down to 200 mph, the lander and rover never actually reached Mars' surface for our first prototype. This initial model needed improvement, which was completed by considering the change in air density as the altitude changes. By adding a function that accurately alters the air density within our while loop this was corrected.

Our final design was successful in meeting all of the design requirements. The lander effectively slowed down from an initial entry speed of approximately 10,000-12,000 mph to 200 mph while safely reaching the surface of Mars. We know the lander safely landed on the surface of Mars as the final speed of the rover was 0 mph. We were able to develop a computational and virtual model. This computational model included the changing of air density as a function of height; an accurate reflection of the force of drag using different values for the drag coefficient based off of the state of the parachute deployment; calculated force of gravity and net force; updated the position, momentum, time, speed, and area of the parachute. The virtual model displayed a representation of the lander from the time it entered the atmosphere of Mars to safely landing on Mars's surface (see Figure 11).

In short, the final design combined the fixes from the two aforementioned iterations to create a final design that accomplished and met the desired requirements provided by the client. The link attached below consists of a program "Lab6Part4" which is the code for the final design.

https://www.glowscript.org/#/user/skthomps/folder/MartianChallenge/

The first feature of our final design is the addition of a drag coefficient for both the lander and the parachute. At first only a drag coefficient of 1.072 is needed because the parachute has not been deployed. Once the parachute is deployed, the drag coefficient is changed to 1.072 + .5 to accommodate this. This aspect is important because the air density changes as the altitude changes. This affects the velocity of the land rover greatly. The land rover reaches the ground. We accomplished this by changing the condition of our while loop to run when the height is greater than 0. This changed the final height to 0. The final updated feature of our program is putting the air density in the loop so that it updates. This is crucial because the air density is going to change with height.

### b. Data on Final Solution

The testing plan to collect data on our final solution included adding two print statements to the bottom of the output display. By adding these print statements to the display we were able to receive exact numbers for the final height and speed of the rover, which is crucial to determining if the design requirements were met.

The visual output illustrates that the rover successfully and safely lands on Mars (Figure 16). Figure 16 includes three images: the leftmost image is a graph of the change in speed over time; the middle image illustrates the trajectory of the rover; and the rightmost image is the change in height over time. These visuals evidently demonstrate that the rover landed on Mars with a speed less than 200 mph or 89.41 m/s.

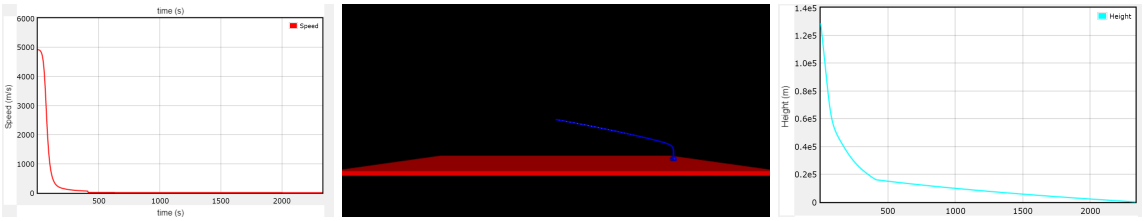



*Figure 16: Final Display for Design Solution*

Final Height: -0.0 m.
Final Speed: 5.9 m/s.

*Figure 17: Final Design Solution Statistics, Computational and Visual Model Display*

We can also view the added print statements to the code to determine the exact end data of the coded simulation. To reiterate, the first design requirement was for the rover to land on the surface of Mars. This first design requirement was evidently met by our solution as the final height of the rover is 0 meters (Figure 17). Another design requirement provided by NASA was to land at a maximum speed of 200 mph, as demonstrated in Figure 17, our design caused the rover to land at a final speed of 5.9 m/s or 13.20 mph. Therefore, based on the data collected through testing, our final solution meets the design requirements provided by NASA.

### c. Trade-Offs Decisions and Limitations of the Solution

Inherently this design had certain limitations, resulting in making important trade-off decisions throughout the process. The biggest constraint on this solution was due to the deadline. The tight turn around caused challenges with innovating a new, successful prototype that met our client's criteria as we did not have a significant amount of time to see through the more interesting, complex ideas we generated. Another reason we were not able to see the more interesting, complex ideas through as we were simultaneously learning GlowScript, generating prototypes, and conducting design evaluation. Had the team been given more time we would have been able to develop a more innovative solution. One concern with our solution is the feasibility of the design to be successful in the real atmosphere of Mars, which could be discovered through the addition of more resources at the disposal of the team. Ultimately, our design was simplistic in idea, but the computational and visual model closely reflected the conditions on Mars. Furthermore, the trade-offs we made led us to a successful prototype which met all of the requirements provided by the client.

### d. Lessons Learned

One of our team's primary strengths during this process was our ability to communicate our ideas effectively to one another and efficiently collaborate. The team's strength in communication and collaboration allowed us to easily bounce ideas off of one another and work efficiently during the problem solving and design process. Each team member had a solid understanding of the project and the principles of physics that came with it which allowed the team to move forward as a collective unit. In specific reference to Problem Solving, a strength of our team was idea generation as we had the ability to develop many different solutions and categorize them appropriately to best breakdown the solutions into components. A strength of our team in reference to the Design Approach was our ability to use code and figure out which aspects would cause a decrease in speed and reach the surface of Mars. Lastly, a strategies strength of ours was our ability to input certain equations into the while loop in order for their values to iteratively update; we also were able to accurately change the conditions of the while loop to best simulate the situation. One area the team could improve on is we fixated on one idea instead of branching out to design prototypes of our more complex ideas that were generated. Through this process we gained a better grasp on Python and learned more about the exploration of Mars. There were many positive take-aways and it was beneficial to the growth of the team to be able to apply concepts from class into a real-world application in the format of an iterative project.

## IX. REFERENCES


Airbags. (n.d.). Retrieved March 17, 2021, from https://mars.nasa.gov/mer/mission/spacecraft/entry-descent-and-landing-configuration/airbags/

Atkinson, N. (2015, December 25). Can an immobile spirit rover survive the martian winter? Retrieved March 31, 2021, from https://www.universetoday.com/52242/can-an-immobile-spirit-rover-survive-the-martian-winter/

Eric Berger - Feb 18, 2. (2021, February 18). What to expect from the Dramatic mars Perseverance Landing [UPDATED WITH LIVESTREAM]. Retrieved March 31, 2021, from https://arstechnica.com/science/2021/02/what-to-expect-from-the-dramatic-mars-perseverance-landing-on-thursday/

How to WATCH NASA Perseverance rover landing on Mars: Live Stream, Indian Time. (2021, February 18). Retrieved March 31, 2021, from https://www.indiatimes.com/technology/science-and-future/nasa-perseverance-rover-landing-on-mars-live-stream-landing-time-534536.html

Howell, E. (2018, July 17). Curiosity rover: Facts and information. Retrieved March 31, 2021, from https://www.space.com/17963-mars-curiosity.html

Howell, E. (2018, September 12). NASA's spirit Rover: Revealing MARS' Early, wet history. Retrieved March 31, 2021, from https://www.space.com/18766-spirit-rover.html

Mars exploration rover mission: The mission. (n.d.). Retrieved March 17, 2021, from https://mars.nasa.gov/mer/mission/spacecraft_edl_airbags.html

Protective fabric. (n.d.). Retrieved March 17, 2021, from http://www.warwickmills.com/Protective-Fabrics-Mars.aspx

Verges, A. M., & Braun, R. D. (n.d.). EVALUATION OF THE MARS PATHFINDER PARACHUTE DRAG COEFFICIENT. Retrieved March 24, 2021, from http://www.ssdl.gatech.edu/sites/default/files/ssdl-files/papers/conferencePapers/IPPW-2007-Verges.pdf#:~:text=Parachute%20drag%20coefficient%20is%20a%20key%20parameter%20in,conditions%20cannot%20be%20matched%20in%20tests%20on%20the

Wall, M. (2013, May 28). Mars rover Curiosity's 7 biggest Discoveries (so far). Retrieved March 31, 2021, from https://www.space.com/20396-mars-rover-curiosity-big-discoveries.html

## X. APPENDICES

Martian Challenge Prototype GlowScript Folder:

GlowScript IDE

Martian Challenge Design Solution GlowScript Folder:

https://www.glowscript.org/#/user/skthomps/folder/MartianChallenge/

## XI. SIGNATURES OF ALL PARTICIPATING STUDENTS

---

**Student # 1**

LAST NAME: Abram

FIRST NAME: Isaac

I hereby affirm that I have contributed substantially to the completion of this report and supporting tasks.

SIGNATURE: *Isaac Abram*

---

**Student # 2**

LAST NAME: Hardin

FIRST NAME: Alexander

I hereby affirm that I have contributed substantially to the completion of this report and supporting tasks.

SIGNATURE: *Alexander Hardin*

---

**Student # 3**

LAST NAME: Hasenour

FIRST NAME: Alexandra

I hereby affirm that I have contributed substantially to the completion of this report and supporting tasks.

SIGNATURE: *Alexandra Hasenour*

---

**Student # 4**

LAST NAME: Lanie

FIRST NAME: Sarah

I hereby affirm that I have contributed substantially to the completion of this report and supporting tasks.

SIGNATURE: *Sarah Lanie*

---

**Student # 5**

LAST NAME: Thompson

FIRST NAME: Samantha

I hereby affirm that I have contributed substantially to the completion of this report and supporting tasks.

SIGNATURE: *Samantha Thompson*

---