InternPro Weekly Progress Update

Name	Email	Project Name	NDA/ Non-NDA	InternPro Start Date	OPT
Adharsh Prasad Natesan	anatesan@asu.edu	IT-Core Foundation Suriname	Non-NDA	2024-08-05	Yes

Progress

Include an itemized list of the tasks you completed this week.

#	Action Item/ Explanation	Total Time This Week (hours)
1	This simulation creates a 2D visualization of two interacting particles	3
2	This simulation creates a 2D visualization of multiple particles colliding	3
3	3D Particle-Based Terrain Generation with mayavi	3
4	Rover into the mayavi simulation environment	3
5	Literature on wheel soil interaction	3
6	Literature on wheel soil interaction	3
7	Weekly plan	1
8	Report writing	1
	Total hours for the week:	20

Verification Documentation:

Action Item 1: This simulation creates a 2D visualization of two interacting particles – 3 hour(s).

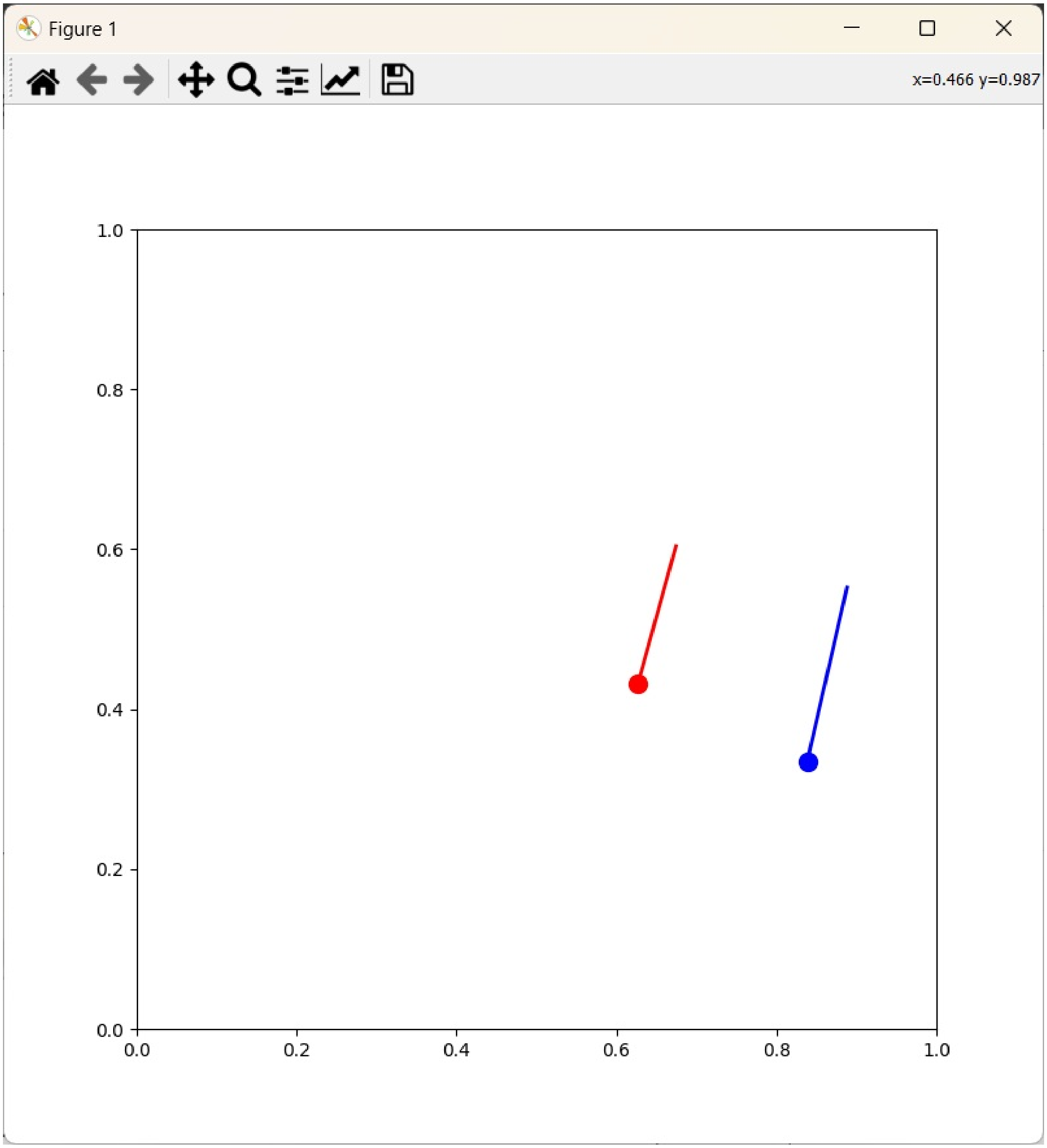
Project Work Summary

- Particle Initialization:
  - Two particles are created with defined initial positions, radii, and masses.
  - Random initial velocities are assigned to each particle to initiate movement.
- Force Calculation:
  - A custom calculate_force function computes the interaction force between the two particles.
  - The force is based on the overlap between particles, simulating a simple spring model.
- Particle Update:
  - The update_particle function updates each particle's position and velocity based on applied forces.

- Gravitational acceleration is included to simulate a downward pull on the particles.
- Collision detection with walls is implemented, causing particles to bounce with a damping factor.

- Visualization:
- The simulation uses Matplotlib to create a 2D animation of the particles.
- A figure is set up with dimensions of 8x8 inches and a 1:1 aspect ratio.
- Each particle is represented as a colored circle (red and blue) with a defined size.
- Direction arrows are added to show the instantaneous velocity of each particle.
- Animation:
- The FuncAnimation function from Matplotlib is used to create a smooth animation.
- In each frame, particle positions and velocities are updated based on their interactions.
- The circles and arrows representing the particles are redrawn to reflect their new positions and velocities.
- Execution:
- The simulation runs for 500 frames with a 20ms interval between each frame.
- The animation shows the particles moving, colliding with each other and the walls, and responding to gravity.

This simulation demonstrates basic principles of the Discrete Element Method (DEM) in a 2D environment, serving as a foundation for more complex particle interaction simulations. It showcases particle collision dynamics, gravitational effects, and basic visualization techniques using Python and Matplotlib.

- 

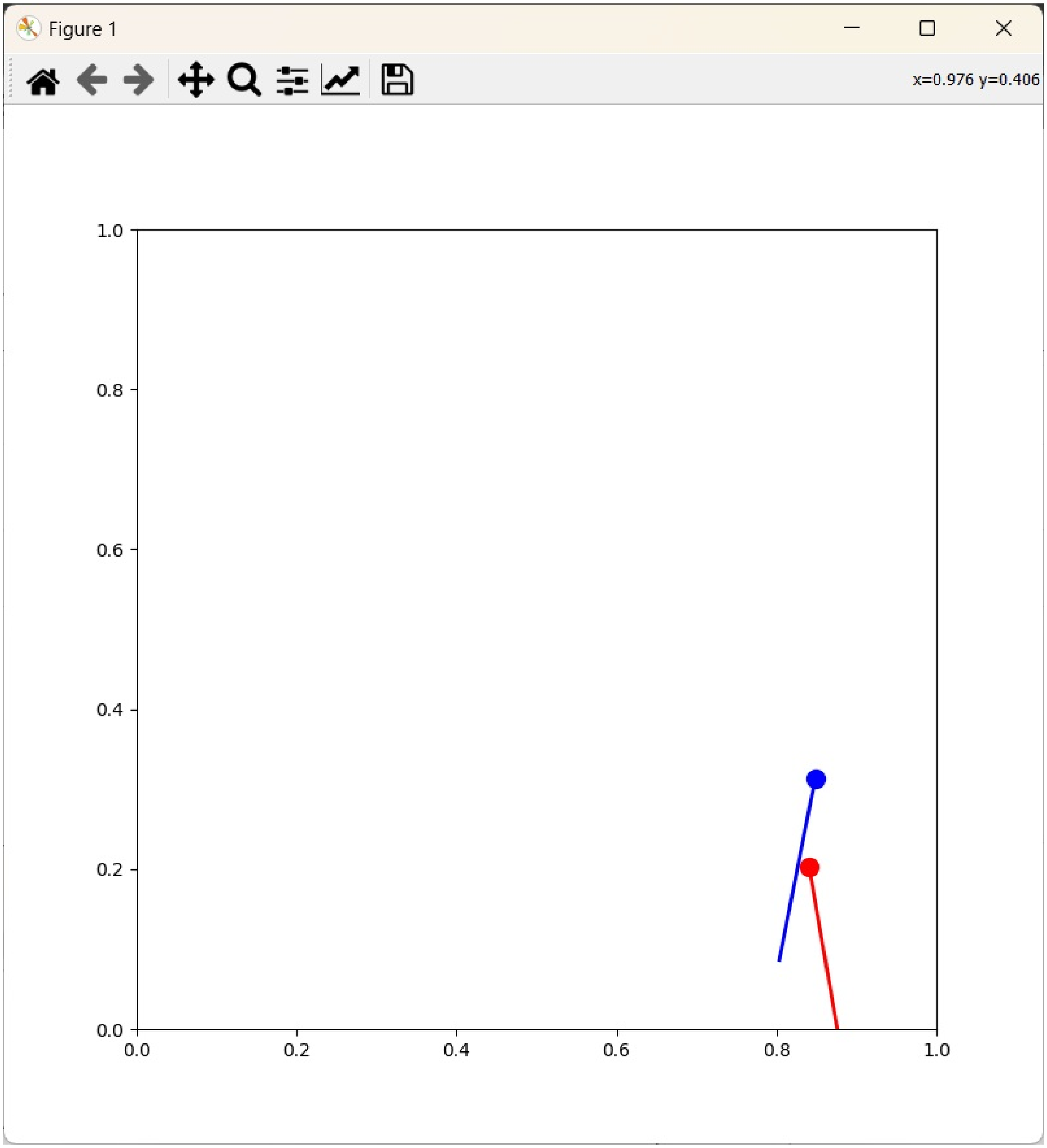
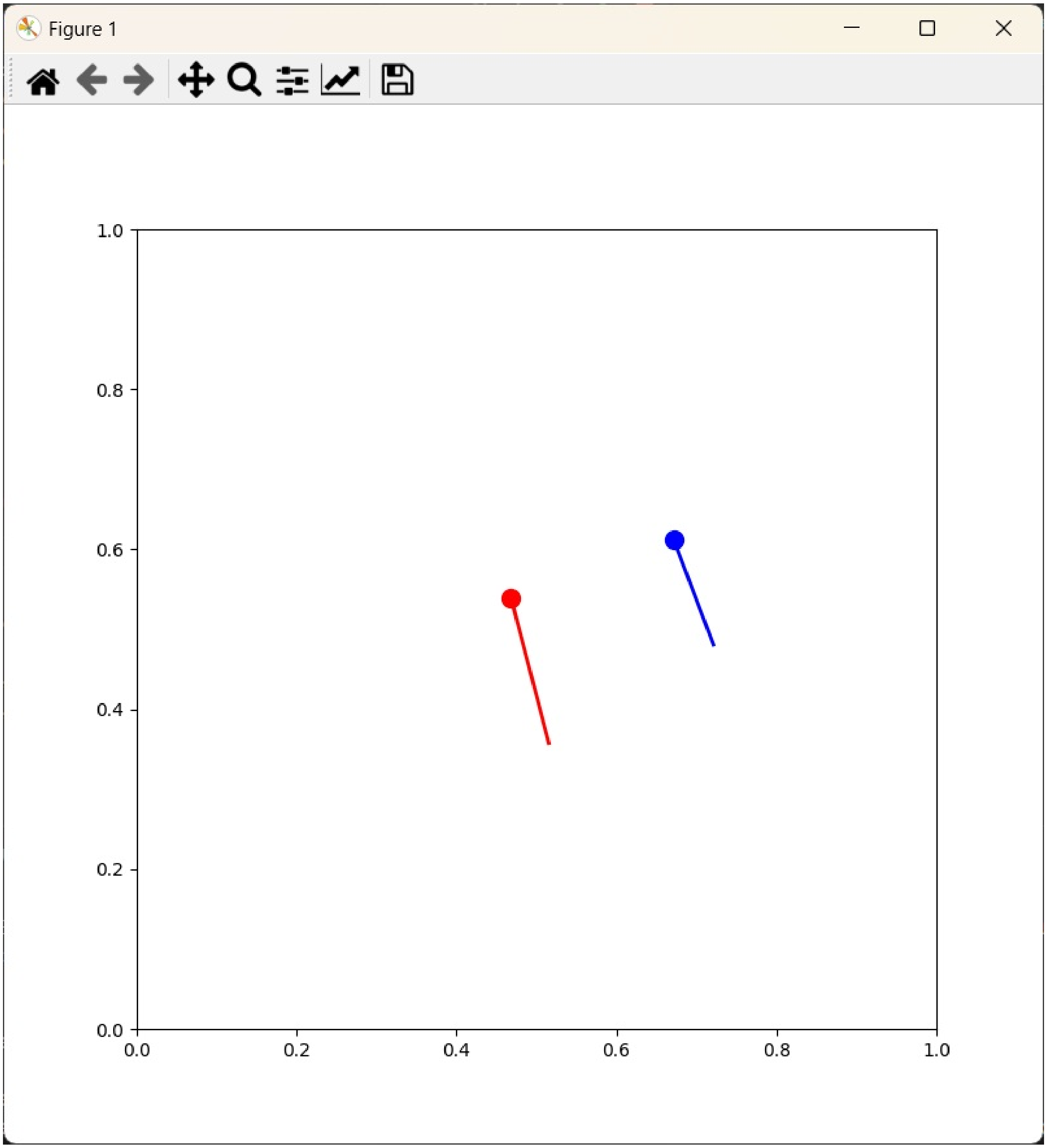


Figure 1
x=0.976 y=0.406
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0



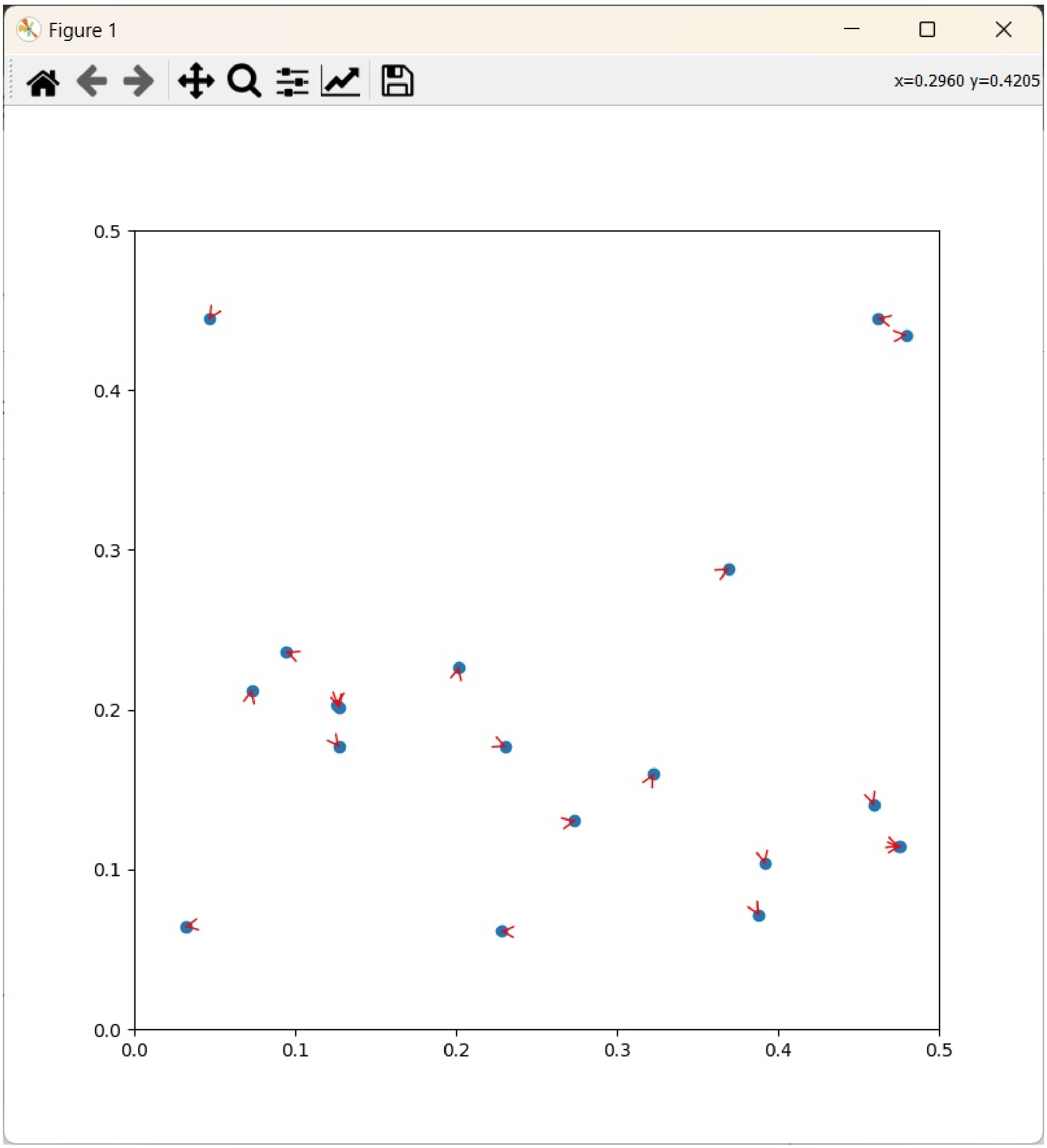
Action Item 2: This simulation creates a 2D visualization of multiple particles colliding – 3 hour(s).

Project Work Summary

- Particle Initialization:
  - Multiple particles (20 in this case) are created with random initial positions and velocities within a 1x1 unit space.
  - Each particle has properties such as position, velocity, radius, and mass.
- Collision Physics:
  - A calculate_collision function implements elastic collision physics between particles.
  - The function uses conservation of momentum and energy to update particle velocities post-collision.
- Boundary Conditions:
  - Particles bounce off the edges of the 1x1 unit space, simulating wall collisions.
  - Velocity is reversed and slightly damped upon wall collision to maintain particles within bounds.
- Visualization:
  - Matplotlib is used to create a 2D animation of the particle system.
  - Particles are represented as colored circles using a scatter plot.

- Direction arrows are added to each particle to indicate their velocity vectors.
- Animation:
  - FuncAnimation from Matplotlib creates a smooth, real-time animation.
  - In each frame, particle positions and velocities are updated based on their interactions.
  - Collision detection and resolution are performed for all particle pairs.
  - Particle positions and direction arrows are redrawn to reflect their new states.
- Key Features:
  - Realistic multi-particle collision simulation in 2D space.
  - Visual representation of particle velocities using direction arrows.
  - Efficient collision detection and resolution for multiple particles.
  - Interactive animation allowing real-time observation of particle behavior.

  This simulation serves as a foundation for more complex particle-based systems, demonstrating basic principles of collision physics and providing a visual tool for studying particle dynamics in a confined space.

- 

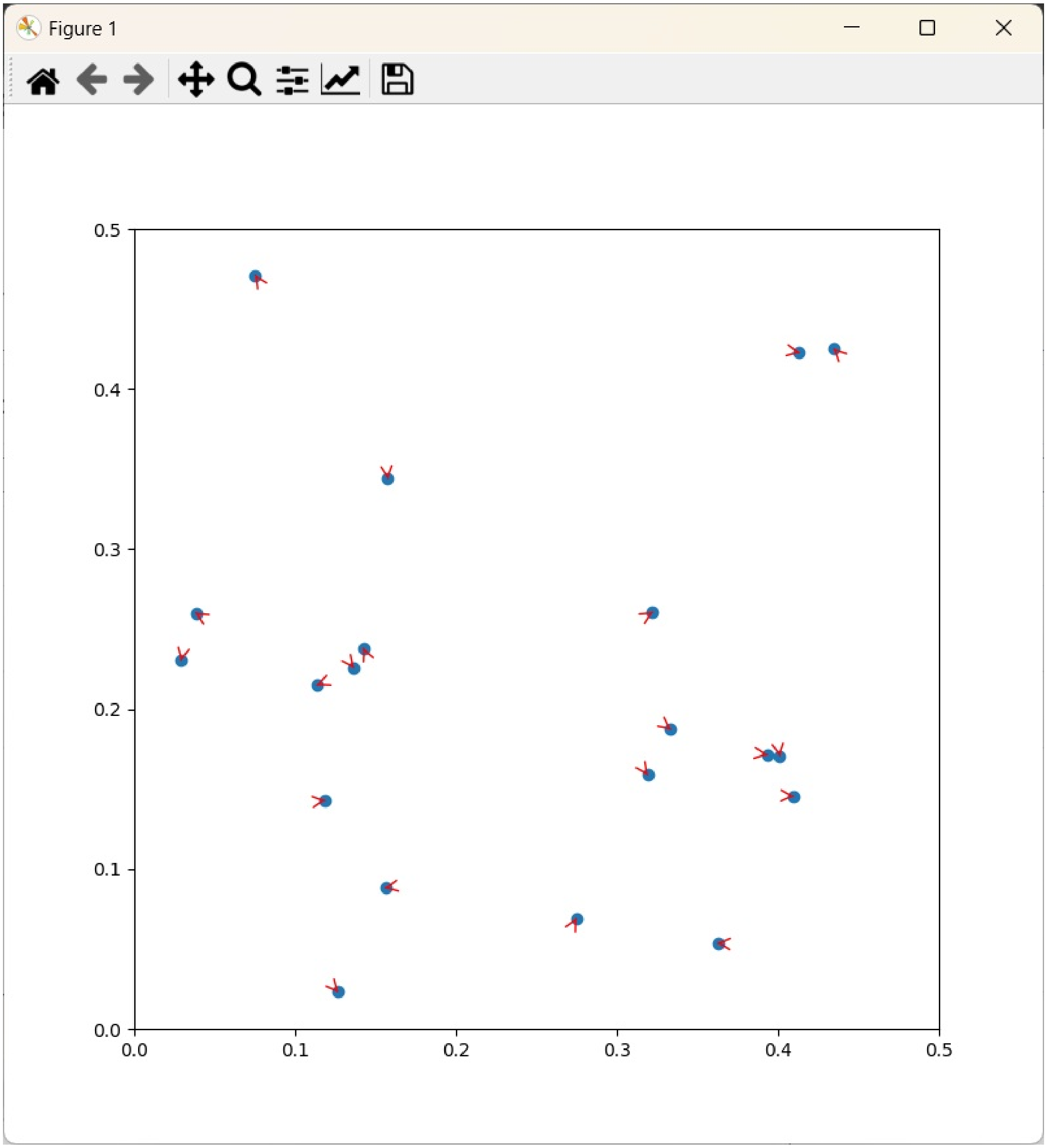
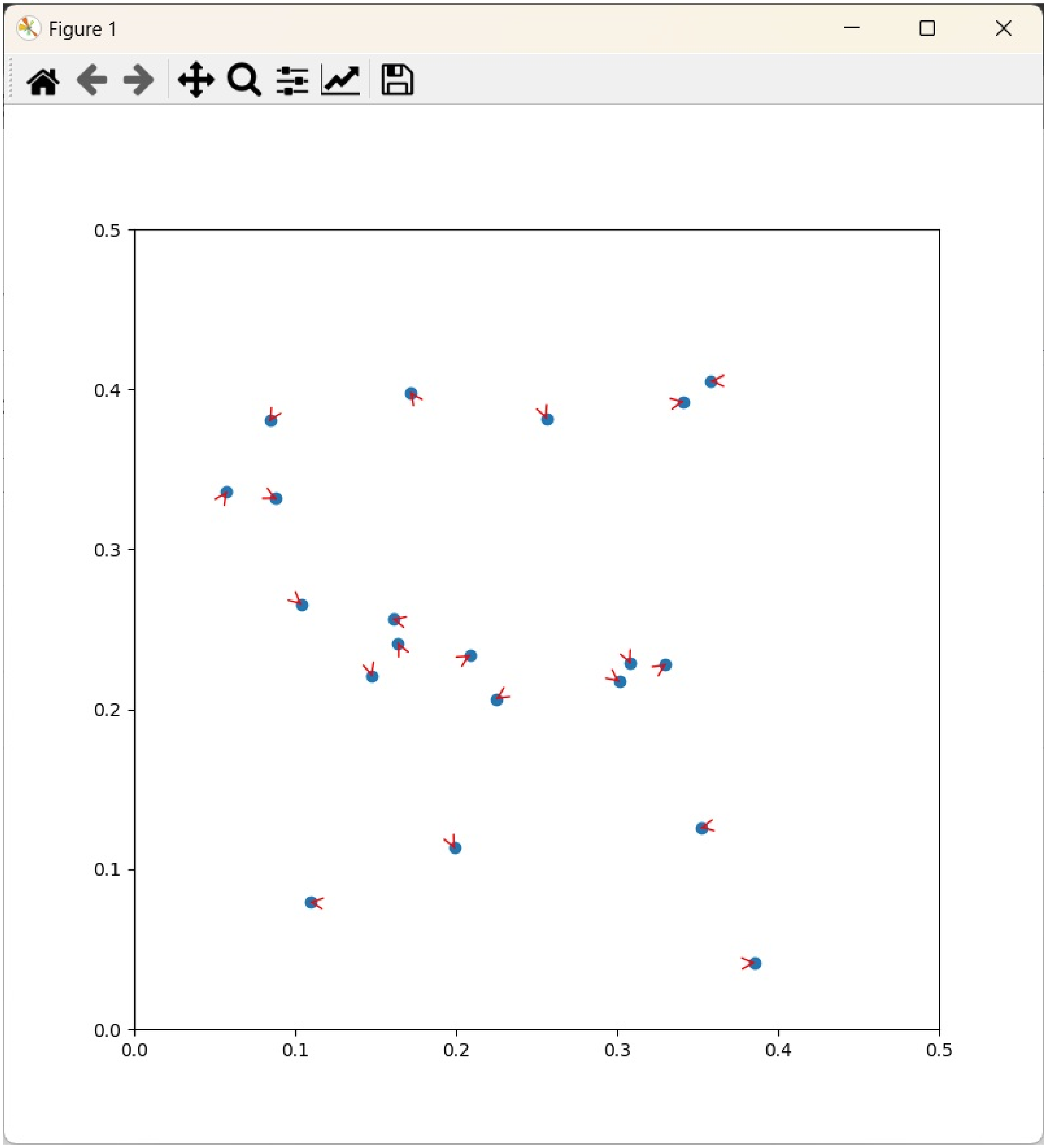


Figure 1
0.5
0.4
0.3
0.2
0.1
0.0
0.0
0.1
0.2
0.3
0.4
0.5

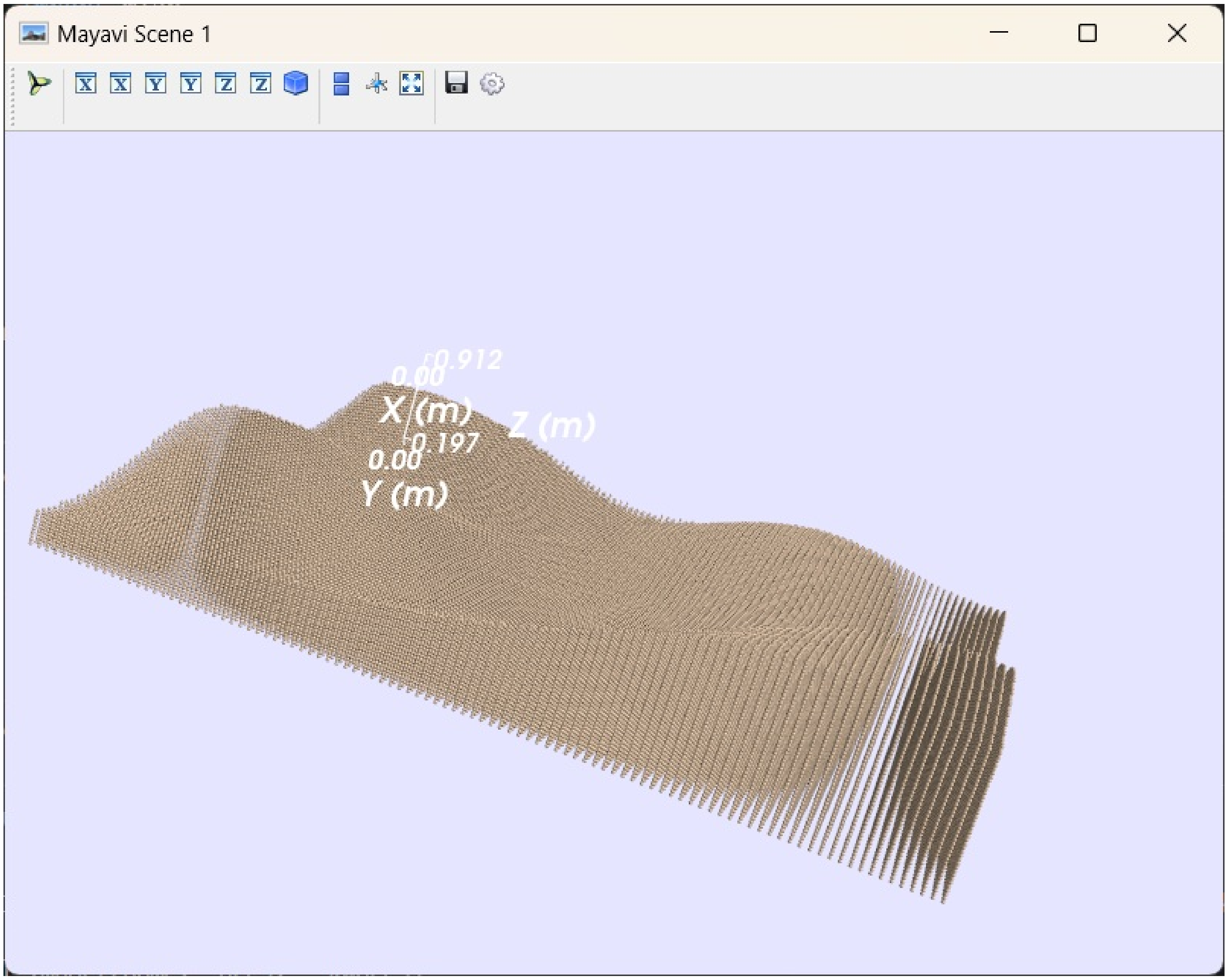
•

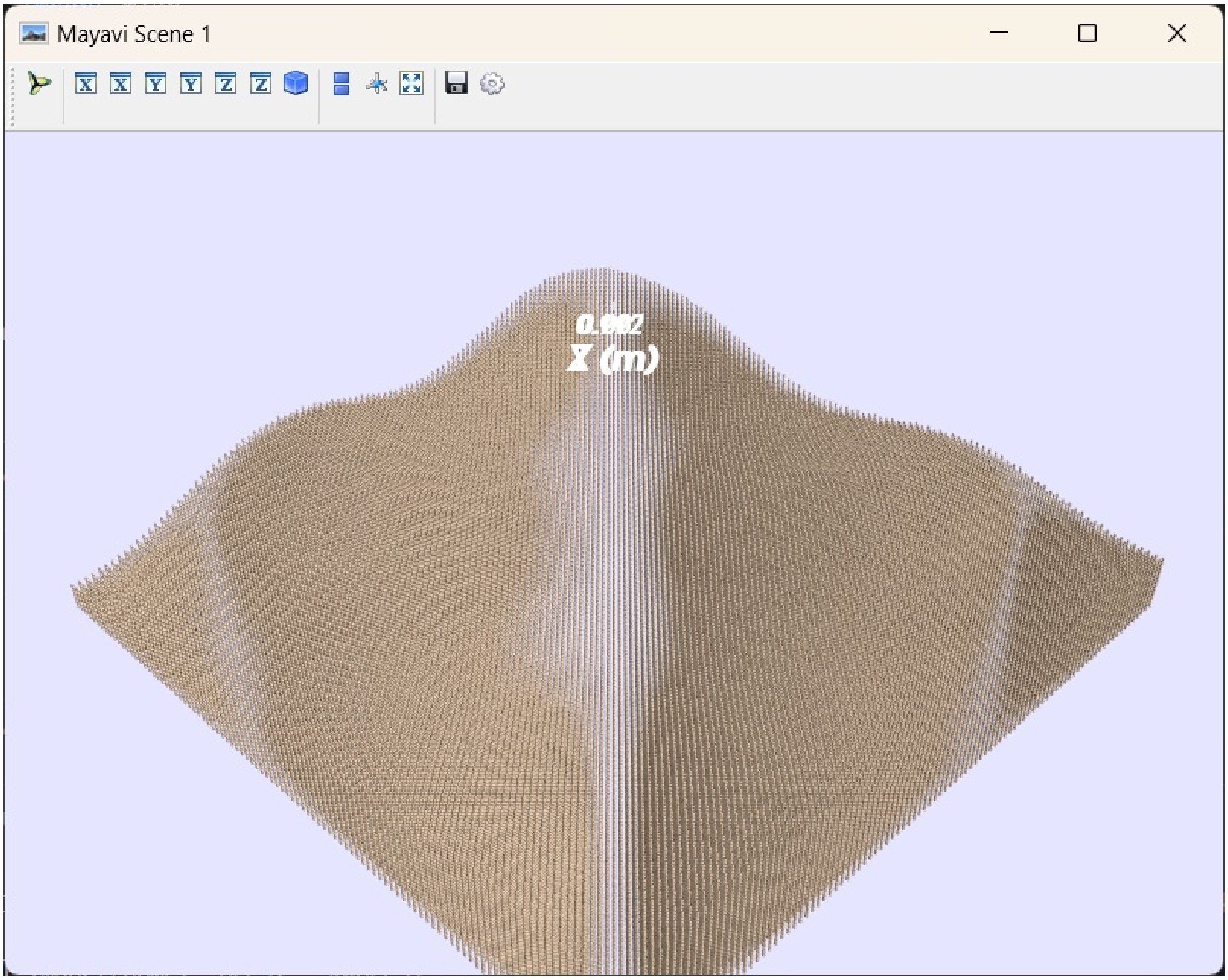


Action Item 3: 3D Particle-Based Terrain Generation with mayavi – 3 hour(s).

Project Work Summary

- This simulation generates a realistic 3D terrain using OpenSimplex noise and represents it with particles. Key components include:
- Terrain Generation:
  - Creates a 5x5 meter terrain grid with 0.05-meter resolution
  - Uses OpenSimplex noise to generate height values for a natural, random terrain
- Particle-Based Representation:
  - Converts the terrain into a volumetric particle representation
  - Fills the space from the ground up to the terrain height with particles
  - Uses small particle size (0.02 meters) for detailed terrain representation
- Visualization with Mayavi:
  - Renders particles as 3D points with a sand-like color
  - Adds a semi-transparent surface overlay for better terrain shape visualization
  - Sets up a 3D view with specific azimuth, elevation, and distance

- Additional Features:
  - Adds coordinate axes for reference
  - Uses a light blue background for sky-like appearance
- 

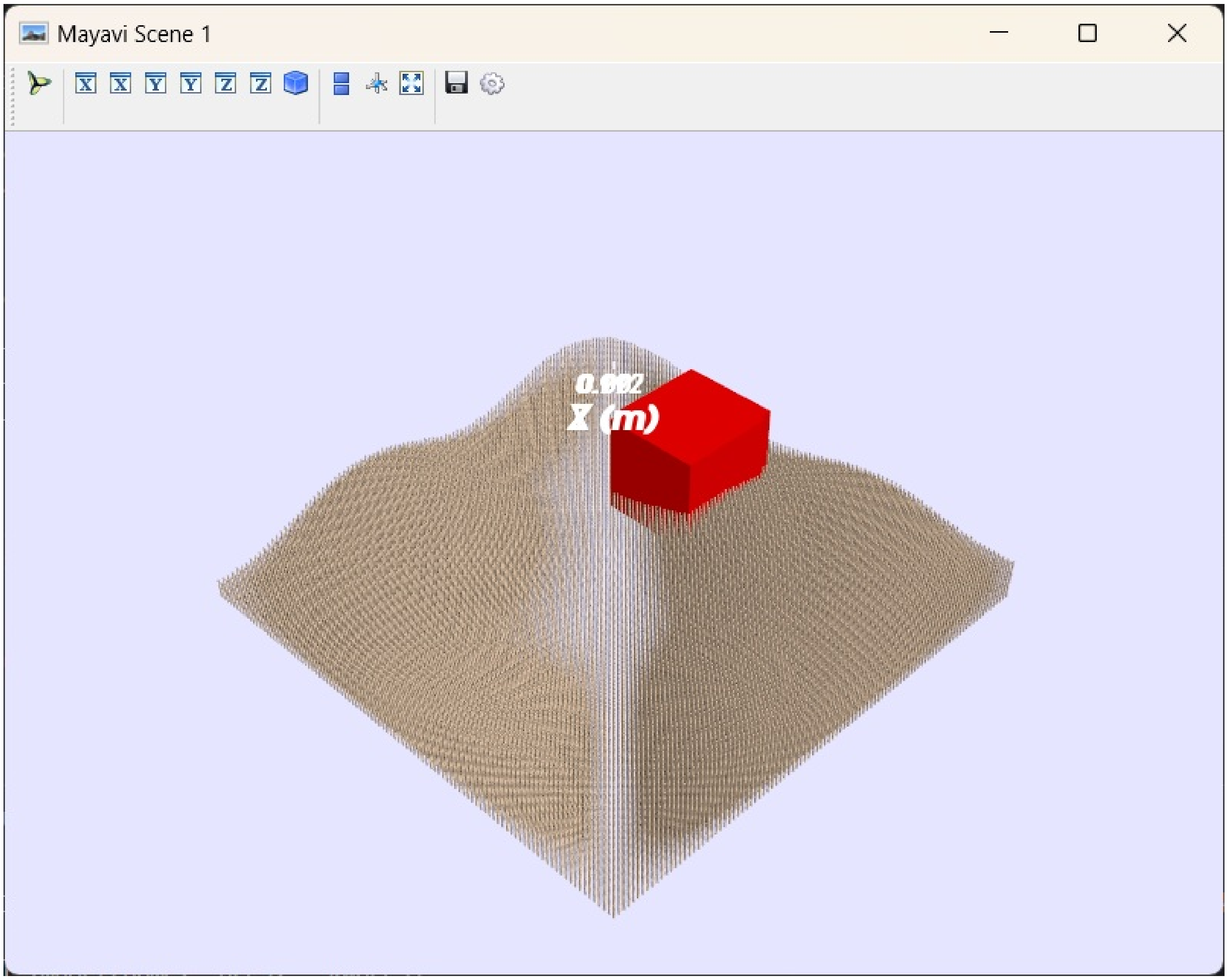
- 

-

Action Item 4: Rover into the mayavi simulation environment – 3 hour(s).

Project Work Summary

- Terrain Generation and Visualization
  - Creates a 5x5 meter terrain using OpenSimplex noise for natural height variations
  - Implements a volumetric particle representation with sand-colored particles (0.02m size)
  - Generates a semi-transparent surface overlay for better terrain visualization
  - Uses high-resolution grid (0.05m spacing) for detailed terrain features
- Particle-Based Representation
  - Fills the terrain volume with particles from ground level to the terrain height
  - Creates a realistic sand-like appearance using beige-colored particles
  - Renders particles using Mayavi's points3d function with appropriate scaling
  - Implements proper depth visualization through semi-transparent surface overlay
- Rover Visualization
  - Places a red cuboid representing the rover on the terrain
  - Positions the rover at coordinates (1.0, 2.0, 0.5) in the 3D space
  - Sets rover dimensions to 0.5m length, 0.25m width, and 0.15m height
  - Creates a semi-transparent representation allowing visibility of underlying terrain
- Scene Configuration
  - Sets up a Mayavi figure with 800x600 pixel dimensions
  - Uses light blue background (0.9, 0.9, 1.0) for sky-like appearance
  - Configures 3D view with 45° azimuth and 50° elevation
  - Includes labeled axes for spatial reference
- This visualization serves as a foundation for simulating rover-terrain interactions and can be extended to include dynamic behaviors and soil deformation effects in future implementations.

- 

-

Action Item 5: Literature on wheel soil interaction – 3 hour(s).

Research

- https://www.janss.kr/archive/view_article?pid=jass-38-4-237
- Title: Development of a New Pressure-Sinkage Model for Rover Wheel
- Summary of Report:
  - Presents a new pressure-sinkage model based on dimensional analysis and bevameter tests
  - Addresses limitations in existing wheel-soil interaction models for planetary rovers
  - Focuses on the interaction between rover wheels and loose soil (regolith)
- Key Points:
  - Introduces a comprehensive model that considers both static and dynamic sinkage
  - Uses custom-built bevameter for experimental validation
  - Demonstrates how wheel sinkage can be divided into static (under vertical load) and dynamic (during rotation) components
- Relation to Project:
  - Provides fundamental understanding of wheel-soil interaction physics
  - Offers validated methodology for simulating rover wheel behavior in loose soil
  - Can be integrated into particle-based terrain simulation systems
- Motivation for Research:
  - Address limitations in traditional Bekker-Wong terramechanics theory
  - Improve accuracy of wheel-soil interaction models for small rovers
  - Develop better prediction models for rover performance in loose soil

Action Item 6: Literature on wheel soil interaction – 3 hour(s).

Project Work Summary

- https://elib.dlr.de/60532/1/elib_Astra2008_KrennHirzinger.pdf
- Title: Simulation of Rover Locomotion on Sandy Terrain - Modeling, Verification and Validation
- Summary of Report:
  - Presents SCM (Soil Contact Model) for simulating physical interaction between rover mobility systems and planetary soil
  - Implements Bekker's empirical terramechanics formulae
  - Uses digital elevation model (DEM) for soil geometry representation
- Key Points:
  - Handles arbitrary contact objects without restrictions
  - Includes bulldozing forces and lateral forces in calculations
  - Implements plastic soil deformation for realistic terrain interaction
- Relation to Project:
  - Directly applicable to our particle-based terrain visualization
  - Provides framework for implementing soil deformation mechanics
  - Offers validated approach for simulating rover-terrain interaction
- Motivation for Research:
  - Create model compatible with standard multi-body simulation engines
  - Enable simulation of various locomotion systems beyond wheeled rovers
  - Improve accuracy of terrain deformation prediction in rover simulations

Action Item 7: Weekly plan – 1 hour(s).

Project Work Summary

- Code Enhancement Tasks
  - Implement soil deformation mechanics when the rover interacts with terrain particles
  - Add realistic friction coefficients between rover and soil particles
  - Develop a more accurate gravity simulation for particle settling
- Rover Model Improvements
  - Create a more detailed rover model with:
  - Wheel geometry and grouser patterns
  - Suspension system representation
  - Mass distribution properties
  - Add wheel-soil contact detection algorithms for better interaction simulation
- Terrain Generation Enhancements
  - Implement multiple soil types with different particle properties
  - Add terrain layers with varying densities and particle sizes
  - Improve the OpenSimplex noise parameters for more realistic terrain generation

Action Item 8: Report writing – 1 hour(s).

Project Work Summary

- Created word document layout to write contents of the weekly progress.
- Created relevant subsections in the epicspro website and documented 20 hours of weekly progress.
- Collected relevant documents research papers, relevant links and company's objective from their portal.