WORLD SPACE WEEK 2026

Rocket Simulation Design Challenge

COLLEGE-LEVEL COMPETITION

ORGANISED BY

ASTROVERSE

WANDERERS OF SPACE

ASTRO PATHSHALA

COLLABORATION & AFFILIATIONS

इसरो isro

REGISTERED SPACE TUTOR

NASA INTERNATIONAL SPACE APPS CHALLENGE

सत्यमेव जयते

MCA

सत्यमेव जयते

MSME

MICRO, SMALL & MEDIUM ENTERPRISES

सूक्ष्म, लघु एवं मध्यम उद्यम

OUR STRENGTH • हमारी शक्ति

Ministry of MSME, Govt. of India

सत्यमेव जयते

DPIIT

#startupindia

OFFICIAL COMPETITION GUIDELINES

Technical Specifications, Rules & Regulations, and Judging Criteria

Document Type	Competition Guideline — Official Release
Category	College Level
Event Window	5th October 2026
Venue	Haldwani, Uttarakhand
Organized By	Astroverse Experience Pvt Ltd

1. Preamble & Purpose

Astroverse Experience Pvt. Ltd. presents the **Rocket Simulation Design Challenge 2026** as part of the **World Space Week 2026** programme.

The challenge is designed to introduce college students to the application of **computational modelling, numerical simulation, aerospace engineering and design optimisation** in modern rocketry.

In professional aerospace engineering, simulation is used to predict vehicle behaviour before hardware is built or tested. NASA, for example, uses computational fluid dynamics to characterize rocket aerodynamic performance, aerodynamic loads and launch environments, while trajectory-optimization tools are used to optimize powered and unpowered vehicle trajectories under engineering constraints.

The competition therefore challenges participants not merely to **simulate a rocket**, but to develop a computational engineering solution to a realistic aerospace problem.

2. Objective

To develop a **computational model or simulation system capable of solving a defined rocket-engineering problem**.

3. Eligibility & Participant Conditions

Competition category: **College Level**

Eligible participants: Undergraduate and equivalent college-level students.

Team size: **1–4 students**

Each team may submit **one primary solution** for evaluation.

Participants must comply with all competition, technical and submission requirements.

Each submission must be the team's original work.

Teams may use open-source libraries and publicly available datasets where properly acknowledged.

4. The Challenges

You are not being asked to make a rocket fly.

You are being asked to predict how it will fly—and use that prediction to make it better.

Each team must select **ONE** of the approved engineering challenges listed below.

The final simulation should answer a meaningful engineering question and produce quantitative outputs that can support a design decision.

CHALLENGE 01

ASCENT TRAJECTORY OPTIMISATION

How do you make a rocket reach maximum altitude with limited resources?

Design a computational model that predicts the ascent trajectory of a launch vehicle and determines an optimized flight/design configuration.

Your simulation should model, at minimum:

- Vehicle mass variation
- Propellant consumption
- Thrust
- Drag
- Gravity
- Atmospheric density
- Velocity
- Acceleration
- Altitude
- Time of flight

Engineering Problem

A launch vehicle has a limited propellant mass and a defined vehicle configuration.

Determine the design or operating parameters that maximize a specified performance objective such as:

Maximum altitude / maximum downrange distance / minimum propellant requirement

subject to defined engineering constraints.

Participants may investigate:

- Thrust-to-weight ratio
- Propellant mass fraction
- Burn time
- Vehicle mass
- Drag coefficient
- Launch angle
- Atmospheric conditions

Minimum Expected Output

The simulation should generate:

- Altitude vs. time
- Velocity vs. time
- Acceleration vs. time
- Mass vs. time
- Maximum altitude
- Maximum velocity
- Time to apogee
- Performance comparison between baseline and optimized configurations

Advanced Level

Implement an optimisation algorithm that automatically searches the design space.

NASA's trajectory-optimization tools similarly address targeting and optimization of powered and unpowered trajectories under constraints.

CHALLENGE 02

SIX-DEGREE-OF-FREEDOM ROCKET FLIGHT SIMULATOR

Can you predict not only where a rocket goes—but how it moves?

Develop a **6-DOF flight simulation** of a rocket.

The simulator should model both:

Translational motion

- X, Y, Z position
- Velocity
- Acceleration

Rotational motion

- Pitch
- Yaw
- Roll
- Angular velocity

The model should incorporate appropriate aerodynamic forces and moments.

A 6-DOF simulation should allow the participant to study how disturbances such as wind, misalignment or asymmetric aerodynamic characteristics influence the vehicle.

OpenRocket, for example, provides a full six-degree-of-freedom simulation capability along with stability, wind and aerodynamic analysis features.

Engineering Problem

A rocket experiences a small initial angular disturbance during ascent.

Determine:

- Whether the vehicle remains stable.
- How the disturbance evolves.
- How aerodynamic forces restore or amplify the disturbance.
- How fin configuration affects stability.
- What design changes improve flight robustness.

Minimum Expected Output

- 3D trajectory
- Pitch/yaw/roll history
- Angular rates
- Position and velocity
- Stability response
- Flight-path deviation

Advanced Level

Introduce:

- Wind disturbances
- CG movement
- Mass variation
- Fin misalignment
- Sensor noise
- Control inputs

and perform a **Monte Carlo or sensitivity analysis**.

CHALLENGE 03

REUSABLE ROCKET RE-ENTRY & THERMAL SIMULATION

Can your vehicle survive coming back through the atmosphere?

Develop a simulation of a reusable rocket or spacecraft during atmospheric descent/re-entry.

The objective is to estimate the vehicle's response as it encounters increasing aerodynamic forces and heating.

NASA uses advanced aerodynamic and multiphysics simulations to understand atmospheric flow around launch vehicles and spacecraft, including complex aerodynamic environments.

Engineering Problem

Given a reusable vehicle configuration and initial re-entry conditions, determine a trajectory that balances:

- Aerodynamic deceleration
- Heating
- Dynamic pressure
- Flight stability
- Landing conditions

Your simulation should investigate:

- Atmospheric density
- Velocity
- Mach number
- Drag
- Dynamic pressure
- Convective heating using an appropriate engineering model
- Deceleration
- Altitude

Minimum Expected Output

Generate:

- Altitude vs. time
- Velocity vs. time
- Mach number vs. altitude
- Dynamic pressure vs. altitude
- Estimated heat flux vs. time/altitude
- Maximum deceleration
- Maximum estimated thermal load

Advanced Level

Perform a **trajectory trade study** comparing different:

- Entry angles
- Vehicle masses
- Drag areas
- Lift-to-drag ratios
- Bank/control strategies

The goal is to identify a configuration that meets defined thermal and structural constraints.

CHALLENGE 04

MULTI-STAGE ROCKET PERFORMANCE & STAGING OPTIMISATION

When should a rocket let go of its first stage?

Develop a simulation capable of modelling a **multi-stage launch vehicle**.

The simulator must account for the change in vehicle mass, propulsion and aerodynamic characteristics when a stage burns out and separates.

NASA's SLS development, for example, relies heavily on simulation and aerodynamic databases to understand vehicle behaviour through ascent and booster separation.

Engineering Problem

Design a two- or three-stage launch vehicle model capable of maximizing a defined mission objective.

The simulation should determine how changes in:

- Stage mass
- Propellant mass
- Thrust
- Burn time
- Stage separation timing
- Structural mass fraction
- Aerodynamic drag

affect mission performance.

Minimum Expected Output

The simulator must provide:

- Stage-wise mass profile

- Stage-wise thrust profile
- Velocity vs. time
- Altitude vs. time
- Acceleration vs. time
- Stage separation events
- Final mission performance

Advanced Level

Develop an optimisation routine to determine:

Optimal stage mass ratio + optimal burn duration + optimal staging condition

subject to constraints.

CHALLENGE 05

ROCKET DIGITAL TWIN & FAILURE PREDICTION

Can you build a virtual rocket that predicts when something is going wrong?

Develop a simplified **digital twin** of a rocket flight system.

A digital twin is a virtual representation of a physical system that can be used to simulate and forecast system behaviour. NASA is actively applying digital-twin approaches to spacecraft and aerospace systems to shift testing earlier into the software environment.

Engineering Problem

Create a simulation that represents the nominal behaviour of a rocket and detects deviations caused by failures or abnormal conditions.

The system should simulate selected failure scenarios such as:

- Reduced thrust
- Increased drag
- Sensor error
- CG shift
- Fin misalignment
- Wind disturbance
- Stage-separation delay
- Partial actuator failure

The simulator must answer:

What happened?

Why did it happen?

What will happen next?

Can the system detect the failure early enough to respond?

Minimum Expected Output

- Nominal flight simulation
- Fault-injected simulation
- Sensor/vehicle data
- Fault detection
- Deviation from nominal trajectory
- Failure classification
- Recommended corrective response

Advanced Level

Implement a real-time or near-real-time simulation architecture in which simulated sensor data is continuously fed into the digital twin.

NASA describes digital-twin environments as useful for safely evaluating autonomous systems and testing proposed changes without affecting live operations.

Common SImulation Requirements

Regardless of the selected challenge, every team must demonstrate a technically defensible simulation.

5.1 Mathematical Model

The submission must clearly define the governing equations or computational relationships used.

Examples may include:

- Newton's laws of motion
- Conservation of mass
- Conservation of momentum
- Atmospheric models
- Aerodynamic force models
- Rotational dynamics
- Heat-transfer models
- Propulsion models

The sophistication of the model should be appropriate to the selected challenge.

5.2. INPUT PARAMETERS

The simulation must clearly identify:

Vehicle Parameters

- Mass
- Dimensions
- Reference area
- CG location
- CP location, where applicable
- Moment of inertia, where applicable

Propulsion Parameters

- Thrust
- Burn duration
- Propellant mass
- Mass-flow rate, where applicable

Environmental Parameters

- Atmospheric density
- Temperature
- Pressure
- Gravity
- Wind

Mission Parameters

- Launch conditions
- Initial velocity
- Initial attitude
- Target altitude/range/orbit, where applicable

Teams must clearly distinguish between:

Measured / reference data and **Assumed / estimated data.**

5.3. MODEL VALIDATION

A simulation is only as good as its validation.

Every team must demonstrate that its model produces physically reasonable results.

Validation may be performed using:

- Analytical calculations
- Published/reference data
- Benchmark cases
- Experimental data, where available
- Comparison with an established simulation tool
- Limiting-case analysis

Teams must present:

Input → Model → Simulation → Reference → Error

and calculate appropriate error metrics.

A submission that produces attractive graphs without demonstrating model validity will receive significantly lower technical scores.

5.4. SENSITIVITY & TRADE-OFF ANALYSIS

Teams must identify the parameters that have the greatest influence on the selected mission outcome.

For example:

What happens to altitude if mass increases by 10%?

What happens to maximum velocity if drag increases by 20%?

What happens to landing accuracy if the entry angle changes?

What happens to mission performance if stage separation is delayed?

At least **three major parameters** should be investigated.

The team must explain:

Which parameter matters most—and why?

5.5. OPTIMISATION

Where applicable, teams should implement an optimisation methodology.

Possible approaches include:

- Parameter sweep
- Grid search
- Gradient-based optimisation
- Genetic algorithm
- Particle swarm optimisation
- Bayesian optimisation
- Other justified optimisation techniques

The chosen method must be explained and justified.

The competition does **not** reward the use of a sophisticated algorithm merely for its own sake.

Engineering insight matters more than algorithmic complexity.

5.6. SOFTWARE & TECHNOLOGY

Teams are free to choose their development environment.

Possible platforms include:

- Python
- MATLAB / Simulink
- C / C++
- Julia
- Java
- Fortran
- Other approved computational environments

Open-source libraries may be used.

However:

The submitted source code must be sufficient to reproduce the team's results.

Teams must disclose:

- Programming language
- Major libraries
- External datasets
- External models
- Solver packages
- Version information
- Hardware/software requirements

---

6. What Your Final Submission should Produce

Depending on the selected challenge, the final system should provide meaningful outputs such as:

Flight Performance

Aerodynamics

Propulsion

Re-entry

Systems

Optimisation

---

7. Mandatory Submission Requirements

The competition will follow a **two-stage submission process**.

STAGE 1 — ABSTRACT & METHODOLOGY

To be submitted at the time of registration/submission

Each team must submit:

A. Abstract

The abstract must clearly describe:

- Selected challenge
- Engineering problem
- Proposed simulation
- Expected outcome
- Engineering significance

B. Methodology

The methodology must explain:

- System architecture
- Mathematical/computational model
- Major assumptions
- Input parameters
- Simulation method

- Numerical approach
- Validation strategy
- Optimisation approach, where applicable
- Expected outputs

The Abstract and Methodology will be used to assess whether the proposed project is technically appropriate for the competition.

---

8. Final Code Submission

DEADLINE: 30 SEPTEMBER 2026

Every team must submit the complete **source code of the simulation** before:

30 September 2026

The submission must include:

- Complete source code
- README / installation instructions
- Required libraries/dependencies
- Input files
- Configuration files
- Sample dataset, where applicable
- Example run/output
- Instructions for reproducing the results

The submitted code must be sufficiently complete for the Technical Committee to execute or inspect the simulation.

---

9. Reproducibility Requirement

If the judges cannot reproduce your result, the result cannot be fully evaluated.

The submitted package should allow the judging team to understand:

Input → Code → Model → Solver → Output

Teams must not submit only:

- Screenshots
- Videos
- Graphs
- Executable files without source

- Presentation slides

These may supplement the submission but **cannot replace the required source code**.

---

10. Project Documentation

The final project documentation should contain:

1. Problem Statement
2. Engineering Objective
3. System Architecture
4. Mathematical Model
5. Assumptions
6. Input Parameters
7. Numerical Method
8. Software Architecture
9. Validation
10. Simulation Results
11. Sensitivity Analysis
12. Optimisation
13. Limitations
14. Engineering Conclusions
15. Future Improvements

---

11. How You will be Evaluated

The competition will evaluate both **engineering quality and computational quality**.

1. ENGINEERING UNDERSTANDING — 20%

2. MATHEMATICAL & PHYSICAL MODEL — 20%

3. SIMULATION QUALITY — 20%

4. VALIDATION & VERIFICATION — 15%

5. OPTIMISATION & ENGINEERING INSIGHT — 15%

6. SOFTWARE QUALITY & DOCUMENTATION — 10%

12. Key Rules

1. Original Work

The submitted simulation must represent the team's own engineering work.

2. Open-Source Software

Open-source libraries and frameworks may be used, but teams must clearly disclose their use.

3. External Models

Use of existing models, datasets or software must be declared.

4. No Black-Box Submission

Teams must understand and be able to explain the major computational methods used in their simulation.

5. Source Code is Mandatory

Failure to submit the required source code by **30 September 2026** may make the entry ineligible for final evaluation.

6. Reproducibility

The Technical Committee may execute or inspect the submitted source code.

7. Academic Integrity

Plagiarism, fabricated simulation results, falsified datasets or misrepresentation of another team's work may result in disqualification.

8. Technical Decisions

The decisions of the Technical Committee and Judging Panel regarding technical validity, eligibility and scoring shall be final.