MISRA (Meteor Impact Simulator and Risk Assessment): A web tool to compute the environmental consequences of meteor impacts on Earth

Bendezu Pastrana, Tommy
Dept. of Mechatronics Engineering
Universidad Continental
Huancayo, Peru
tommyhanss007@gmail.com

Ildefonso Santos, Steve
Dept. of Computer Science
UTEC
Huancayo, Peru
steve.ildefonso@utec.edu.pe

***Abstract*—The NASA Space Apps Challenge 2025 introduces the *"Meteor Madness"* challenge, which seeks to develop a tool capable of simulating meteor impacts on Earth, assessing their consequences, and helping the relevant authorities implement preventive measures. This project aims to build an interactive web application that enables users to enter specific meteor parameters, such as size, velocity, and entry angle, to compute and visualize environmental consequences. The tool integrates real data from the NASA Near-Earth Object (NEO) API and the USGS, providing a solid scientific foundation for the calculations. The results include estimates of the extent of the affected area, the magnitude of the blast wave, tsunami generation, and other secondary effects. The application also features an educational section that explains the scientific principles behind meteor impacts and mitigation measures. Beyond its direct contribution to planetary safety, the project serves as a starting point for future research on predicting hazardous astronomical phenomena, leveraging real data and scientific models.**

I. Introduction

The NASA Space Apps Challenge 2025, through the *Meteor Madness* prompt, invites the global community to develop innovative solutions that help understand and mitigate the risks posed by near-Earth meteors. Although meteor impacts are rare events, their destructive potential is significant, with effects that include tsunamis, earthquakes, and atmospheric disturbances. Throughout history, impacts of this kind have produced devastating consequences, such as the Chicxulub event, which is believed to have contributed to the extinction of the dinosaurs. This threat underscores the importance of tools that allow us to understand and evaluate these risks [7], [15].

Interactive simulation tools play a crucial role in visualizing these risks, because they allow users to explore different impact scenarios and the associated environmental consequences. These tools not only raise awareness among the public and decision-makers regarding the magnitude of impacts, but also make it possible to test mitigation strategies such as meteor deflection. The ability to simulate how changes in a meteor's parameters (size, velocity, trajectory) affect the outcome can provide valuable information for planetary protection planning [1].

In response to this challenge, this preliminary project proposes the development of an educational web simulator that calculates and visualizes the environmental consequences of an impact. By integrating real data from the NASA Near-Earth Object (NEO) API and the USGS, the simulator will estimate the extent of the affected area, the magnitude of the blast wave, crater formation, and tsunami generation, all grounded in established scientific models. The simulator will also include an educational section that explains the scientific principles behind meteor impacts and mitigation measures [1], [6].

II. Objectives

General objective: Develop an interactive web tool that simulates the impact of near-Earth meteors, allowing users to visualize environmental consequences and explore mitigation strategies using real data from *NASA* and the *USGS*.

Specific objectives:

1) **Simulate the trajectory of near-Earth meteors**:
   - Implement an orbital model that calculates and visualizes an meteor's trajectory based on Keplerian parameters.
2) **Visualize the environmental effects of an impact**:
   - Compute, using scientific models, and display the consequences of a meteor impact, such as crater formation, tsunami generation, and associated seismic activity, relying on existing scientific data and models.
3) **Integrate real data from *NASA* and the *USGS***:
   - Use the NASA Near-Earth Object (NEO) API to retrieve information about nearby meteors and the USGS Earthquake Catalog to model seismic effects, producing a realistic and accurate simulation.
4) **Include an educational section on meteor impacts**:

- Develop an educational section within the tool that explains the scientific foundations behind meteor impacts, the potential environmental consequences, and mitigation strategies.

5) **Provide an interactive and accessible interface for non-expert users**:
   - Create a user-friendly interface that allows users to adjust parameters (such as meteor size and velocity) and see results in real time, ensuring the tool is accessible to both scientists and the general public.
6) **Evaluate and visualize impact mitigation strategies**:
   - Implement the simulation of mitigation strategies, such as meteor deflection, and allow users to visualize how these strategies affect the meteor's trajectory and the resulting impact effects.

III. Methodology

A. Project Approach

The project will follow an **iterative** approach divided into several key phases that will be executed in parallel whenever possible to maximize efficiency. Using **GitHub**, we will evolve each aspect of the app (frontend, backend, etc.) in parallel and then integrate them into a single product.

B. Simulation Development

The simulation will be developed in two main parts: the **meteor trajectory** and the **impact effects**.

- **Trajectory Simulation**: Building on the elliptical orbit design methodology from NASA Mission Visualization, we will define the meteor's orbital elements (semi-major axis, eccentricity, inclination, right ascension of the ascending node, and argument of periapsis) to initialize the simulation. Temporal propagation will be performed by solving Kepler's equation,

$$M(t) = n\,(t - t_0) = E - e\sin E, \tag{1}$$

to obtain the eccentric anomaly E. The true anomaly ν will be computed via

$$\tan\frac{\nu}{2} = \sqrt{\frac{1+e}{1-e}}\,\tan\frac{E}{2}, \tag{2}$$

while the distance to the focus follows

$$r = a\,(1 - e\cos E). \tag{3}$$

With these magnitudes we will evaluate the position vector in an Earth-centered frame by applying Euler rotations over the orbital plane,

$$\mathbf{r}_{\text{ECI}} = R_3(-\Omega)\,R_1(-i)\,R_3(-\omega)\begin{bmatrix} r\cos\nu \\ r\sin\nu \\ 0 \end{bmatrix}, \tag{4}$$

which will allow us to compute position and velocity in an Earth-centered frame. These quantities will generate the three-dimensional trajectory and estimate the impact point by accounting for Earth's rotation, the arrival velocity, and different entry angles [16].

- **Impact Effects Simulation**: We will follow the workflow established in the Earth Impact Effects Program [1], complemented with blast engineering references. The initial projectile properties are defined as

$$m = \tfrac{\pi}{6}\,\rho_i D_i^3, \qquad R_i = \tfrac{D_i}{2}, \tag{5}$$

and the atmospheric evolution will be integrated by solving the deceleration and ablation equations [1],

$$\frac{dv}{dt} = -\frac{C_D A \rho_{\text{atm}}(h)}{2m}\,v^2 - g\sin\gamma, \tag{6}$$

$$\frac{dm}{dt} = -\frac{C_h A \rho_{\text{atm}}(h)}{2Q^*}\,v^3, \tag{7}$$

evaluating breakup when the dynamic pressure exceeds the material strength according to

$$q(h) = \tfrac{1}{2}\,\rho_{\text{atm}}(h)\,v^2 \geq \sigma_c. \tag{8}$$

If the body reaches the ground, we compute the kinetic energy and its equivalent in kilotons of TNT

$$E_k = \tfrac{1}{2} m v_{\text{impact}}^2, \qquad W_{\text{kt}} = \frac{E_k}{4.184 \times 10^{12}\ \text{J}}, \tag{9}$$

distributing the energy among seismic, thermal, and blast fractions following the efficiencies in [1], [14]. For crater sizing we apply Holsapple's dimensionless pi-scaling laws adopted by [1]:

$$\pi_2 = \frac{gR_i}{v_{\text{impact}}^2}, \qquad \pi_3 = \frac{Y_t}{\rho_t v_{\text{impact}}^2}, \tag{10}$$

with Y_t representing target strength. The transient crater radius is obtained by taking the maximum of the gravity and strength regimes [1], [4],

$$R_t^{(g)} = K_g R_i \left(\frac{\rho_i}{\rho_t}\right)^{1/3} \pi_2^{-\mu_g}, \tag{11}$$

$$R_t^{(s)} = K_s R_i \left(\frac{\rho_i}{\rho_t}\right)^{1/3} \pi_3^{-\mu_s}, \tag{12}$$

$$R_t = \max\big(R_t^{(g)}, R_t^{(s)}\big), \tag{13}$$

with coefficients $K_g = 1.161$, $\mu_g = 0.22$, $K_s = 1.03$, and $\mu_s = 0.275$ for rocky targets. The final diameter distinguishes between simple and complex craters [1], [3],

$$D_f = \begin{cases} 1.25\,D_t, & D_t \leq D_{\text{sc}}, \\ 1.17\,D_t^{1.13} D_{\text{sc}}^{-0.13}, & D_t > D_{\text{sc}}, \end{cases} \tag{14}$$

where $D_t = 2R_t$ and D_{sc} is the local simple-to-complex transition threshold. Ballistic ejecta will be modeled with the radial decay laws from the program [1], [2],

$$h_e(r) = 0.14 R_t \left(\frac{R_t}{r}\right)^3, \tag{15}$$

$$v_e(r) = \sqrt{2gR_t}\left(\frac{R_t}{r}\right)^{3/2}, \tag{16}$$

which provide mantle thickness estimates and deposition velocities. For atmospheric effects we adopt the explosive coupling efficiency η_b proposed in [1] and use the Kingery–Bulmash scaling [13] on the equivalent charge $W_{\text{eq}} = \eta_b W_{\text{kt}}$,

$$Z = \frac{R}{W_{\text{eq}}^{1/3}}, \tag{17}$$

$$\Delta P(Z) = \exp\left(\sum_{i=0}^{5} a_i (\ln Z)^i\right), \tag{18}$$

$$I(Z) = \exp\left(\sum_{i=0}^{5} b_i (\ln Z)^i\right) \tag{19}$$

assessing the dynamic structural load through

$$q_{\text{din}} = \Delta P(Z) + \tfrac{1}{2}\rho_{\text{air}} U_s^2(Z), \tag{20}$$

where $U_s(Z)$ comes from the Rankine–Hugoniot relation [13]. The thermal component uses the luminous efficiency η_L tabulated in [1],

$$H(R) = \eta_L \, \frac{E_k}{4\pi R^2 \tau}, \tag{21}$$

integrating the exposure over duration τ to derive stage-specific doses. For ocean impacts we retain the cavity and tsunami modeling by Ward and Asphaug [15],

$$r_c = \left(\frac{3E_k}{2\pi\rho_w g}\right)^{1/3}, \quad \eta(r) = \eta_0 \frac{r_c}{r} \, \exp\Big(-\frac{r - r_c}{L_d}\Big), \tag{22}$$

propagating the coastal *run-up* through Green's law,

$$R_{\text{run-up}} = \eta(r_s)\left(\frac{h_s}{h(r_s)}\right)^{1/4}. \tag{23}$$

This chain of models covers atmospheric burst, land, and ocean impact scenarios and delivers thermal, structural, and inundation damage maps consistent with the referenced sources.

C. Interactivity and Visualization

The interface will be designed to let users input, adjust, and visualize the meteor parameters and simulation results in real time.

- **Parameter Input**: Users will be able to modify parameters such as **size**, **velocity**, and **entry angle** through **sliders** and **text fields**.
- **3D Visualization**: **Three.js** will render the **meteor trajectory** in an interactive 3D environment, allowing users to rotate, zoom in, and zoom out.
- **2D Visualization**: **D3.js** will produce charts and maps that represent the **impact effects**, such as the extent of the affected area, blast-wave magnitude, and tsunami generation.
- **Real-time Updates**: Simulation results will update dynamically as users modify the parameters, providing immediate feedback.
- **Educational Section**: A section with scientific information on meteor impacts will explain the models and data used in the simulation. As both simulations are completed, we will describe the relevant scientific and mathematical concepts employed.

D. Result Validation

To ensure the accuracy and reliability of the simulation, we will conduct a **validation** process on the results obtained.

- **Calibration with Recognized Models**: We will reproduce reference scenarios from the Earth Impact Effects Program [1] and tsunami studies by Ward and Asphaug [6], [15], verifying that crater, overpressure, and run-up profiles agree within 5–10% tolerances.
- **Orbital Validation**: Propagated trajectories will be contrasted with ephemerides from NASA Mission Visualization and JPL Horizons catalogs, measuring the root-mean-square error in position and velocity to guarantee dynamic consistency [16].
- **Historical Cases**: Documented events such as Tunguska [10] and Chelyabinsk [11] will be simulated; the results will be compared with published observations (damage radii, shock-wave intensities, luminosity) calculating relative error metrics.
- **Reproducibility and Auditability**: Parameter sets will be versioned and automated verification notebooks will be published so that third parties can repeat the simulations and review energy and momentum balances at each stage.

E. Optimization and Performance

To maintain interactive response times we will adopt **optimization** strategies across every layer of the application.

- **Adaptive Integration**: The orbital integrator and the effects engine will use variable time steps driven by local error to reduce computation in smooth intervals and focus effort on critical events.
- **Parallel Computing**: Independent calculations (e.g., Monte Carlo sweeps and ejecta propagation) will be delegated to dedicated threads or Web Workers, keeping the main interface responsive.
- **GPU Utilization**: Volumetric rendering and damage maps will be implemented with WebGL, leveraging shaders for real-time interpolation and shading.
- **Numerical Caching**: Recurrent results such as Kingery–Bulmash curves and pi-scaled tables will be stored in memory to avoid recomputation.
- **Continuous Profiling**: The application will be instrumented with monitoring tools (Chrome DevTools, flame graphs) to detect bottlenecks and establish frame-rate and memory budgets.

F. Documentation and Presentation

The delivery will include materials aimed at end users and technical reviewers.

- **User Manual**: An illustrated guide explaining scenario setup, result interpretation, and model limitations.

- **Technical Document**: A specification of physical models, assumptions, validations, and references, linking each equation to its implementation.
- **API and Data**: A description of endpoints, data contracts, and export schemes (GeoJSON, CSV) to integrate the simulation with other tools.
- **Verification Notebooks**: Jupyter/Markdown artifacts that replicate test cases and present comparisons with reference studies.
- **Elevator Pitch**: Slides summarizing key findings, interface screenshots, and recommendations based on the evaluated scenarios.

IV. Architecture

The proposed system follows a **client-server architecture** where the **frontend** interacts with the **backend** through a RESTful API. This modular structure enables the different components of the system to operate efficiently and scale. The following subsections describe the components and data flow in detail.

A. Overview

The architecture consists of the following main modules:

1) **Frontend**: Responsible for user interaction, displaying the simulations, and enabling data input (such as meteor parameters).
2) **Backend**: Handles the simulation computations (trajectory, impact, secondary effects) and integrates external data from *NASA* and the *USGS* APIs.

B. Backend Components

The backend will be built with **Flask**, a lightweight Python framework well suited for handling HTTP requests and running simulation calculations. The main backend components are:

- **Technologies used**:
  - **Flask**: Used to develop the RESTful API that processes frontend requests, executes the simulation calculations, and returns the results.
  - **Python**: The primary language for simulations and data processing. Libraries such as **NumPy** and **SciPy** will support numerical work, while **AstroPy** will cover astronomical computations.
  - **External APIs**:
    - **NASA NEO API**: Provides data about near-Earth meteors, including orbits and sizes.
    - **USGS Earthquake Catalog**: Supplies seismic data associated with impacts (earthquakes, tsunamis, etc.).
- **Data flow**:
  - The **frontend** will send parameters such as size, velocity, and entry angle to the backend through **POST** requests to the API.
  - The **backend** will process these parameters, perform the simulation calculations, and query the external APIs for additional data, returning the results to the frontend in **JSON** format.

C. Frontend Components

The **frontend** will be developed with **React** and **Vite** to deliver an interactive and efficient user interface. The main frontend components are:

- **Technologies used**:
  - **React**: JavaScript framework for building an interactive, state-managed interface. It will serve as the core of the frontend and enable reusable components to display the simulation, results, and input controls.
  - **Vite**: Provides fast development and build workflows. Vite is a next-generation **bundler** that offers rapid reload times and an efficient developer experience.
  - **Three.js**: Library used to render 3D graphics in the browser, enabling visualization of the meteor trajectory and impact effects.
  - **D3.js**: Library for creating 2D data visualizations and maps to portray damage footprints and other secondary effects.
- **Communication with the Backend**:
  - The frontend will send **POST** requests to the backend API containing the meteor parameters entered by the user.

D. Data Flow Between Frontend and Backend

The **data flow** between the frontend and backend will be managed as follows:

- The **frontend** will send the parameters selected by the user (size, velocity, entry angle) to the backend through a **POST** request to the RESTful API.
- The **backend** will receive these parameters, process them using the scientific models to compute the meteor **trajectory** and impact effects, and then return the results to the frontend in **JSON** format.
- The **frontend** will update the visualizations (3D and 2D) in real time as results are received from the backend.

E. Security and Error Handling

To guarantee system security and stability, the following measures will be implemented:

- **Security**:
  - **Input validation**: The backend will validate all parameters received from the frontend to ensure they are correct and within acceptable ranges.
  - **Injection protection**: Measures will be implemented to prevent SQL or XSS injection attacks, ensuring that all user inputs are properly sanitized.
- **Error handling**:
  - **Exception management**: The backend will handle all exceptions and errors that may occur during calculations or when querying external APIs, returning clear error messages to the frontend.

- **User notifications**: The frontend will display user-friendly error messages if problems arise, such as invalid parameters or communication failures with the backend.

V. Open Data (NASA)

The development of this project relies on integrating **open data** provided by **NASA** and the **USGS**, which enables precise and realistic simulations of near-Earth meteor impacts. These data sources offer essential information about **meteors** and the **geological effects** associated with an impact, making it possible to visualize the related risks in greater detail.

A. 1. Use of the NASA NEO API

The **NASA Near-Earth Object (NEO) API** provides access to a vast database of **near-Earth meteors** (NEOs). This API includes critical information such as orbital parameters, size, velocity, and close approach dates to Earth. For our project, these data will be used to calculate and simulate the meteors' **orbital trajectories**, which is fundamental for predicting an impact on Earth.

- **Data used**:
  - **Orbital parameters**: Semi-major axis, eccentricity, inclination, and related elements.
  - **Meteor size**: Needed to estimate the kinetic energy of the impact.
  - **Approach date and distance**: Supports time-dependent trajectory simulations and impact prediction.

The NASA API provides the data required to create accurate simulations and to determine the impact zone and potential secondary effects of the meteor.

B. 2. Use of the USGS Earthquake Catalog API

The **USGS Earthquake Catalog API** provides access to a global catalog of **seismic data**, including information about earthquake location, magnitude, and depth. These data are essential for modeling the seismic effects generated by a meteor impact on Earth. We will also employ **seismic propagation models** to simulate the waves caused by the impact.

- **Data used**:
  - **Earthquake magnitude**: To estimate the strength of the seismic waves.
  - **Location and depth**: To model how seismic waves propagate through the affected terrain.
  - **Impact on geographic areas**: To simulate effects in regions near the impact site.

C. 3. Data Integration and Transparency

Integrating these datasets is key to ensuring the simulations are as accurate as possible. Throughout the project we will guarantee that the data are processed in a **transparent** manner, allowing users to understand where the data come from and how they influence the simulation outcomes. We will also implement mechanisms to verify the **accuracy** and **freshness** of the data in real time, using the **NASA NEO API** for the latest near-Earth meteor information and the **USGS API** for recent seismic records.

D. 4. Application within the Simulation

The data obtained through these APIs will be integrated into the system **backend** to perform the trajectory and impact-effect calculations. The **frontend** will display these results in real time, enabling users to explore different impact scenarios and to see how parameter changes affect the impact areas and environmental consequences.

VI. Scope

This preliminary project defines the functional and technical scope of the **MISRA** platform. Development will cover the following aspects:

- **Physical simulation**: Orbital propagation, atmospheric entry, and impact modeling with damage maps for terrestrial and ocean scenarios using the models described in the methodology.
- **Interactive interface**: 3D and 2D browser visualizations with real-time parameter adjustments, damage indicators, and data export.
- **Educational support**: An explanatory section with physical foundations, references, and a glossary for non-specialist audiences.
- **Infrastructure**: A service backend for computation, scenario storage, and data distribution to web clients.

Out of scope are integration with defense systems, real-time alerting, post-impact climate modeling, and native mobile deployments; these would require additional projects.

VII. Work Plan (48 h) and Deliverables

The project will be executed over a **48-hour** period, splitting the activities into two days with specific tasks to maintain steady, well-organized progress. The following sections describe the work plan and the expected deliverables for each phase.

A. Day 1: Research, Design, and Initial Development

Day 1 Objectives:

- Conduct the necessary research on meteor impacts and the relevant scientific models.
- Define the overall system structure and interface design.
- Begin developing the meteor trajectory simulation and the impact effects module.

Activities:

1) **Research and Review of Scientific Models**
   - Review orbital simulation models and impact effects (such as crater formation and tsunamis).
   - Study the NASA and USGS APIs to integrate meteor and seismic data.
2) **Architecture Definition**
   - Design the client-server architecture.

- Select technologies for the frontend (React, Vite, Three.js, D3.js) and backend (Flask, Python).
- Draft the RESTful API that will connect frontend and backend.

3) **Orbital Trajectory Simulation Development**
   - Implement orbital propagation using Keplerian parameters and equations.
   - Perform an initial validation of the simulation with reference data.
   - Configure the NASA NEO API to retrieve near-Earth meteor data.
4) **Initial Frontend Design**
   - Create a basic interface with React and Vite.
   - Implement the foundational structure for 3D trajectory visualization using Three.js.

Day 1 Deliverables:

- Basic functional prototype with the initial user interface and an early 3D visualization of the meteor trajectory.
- Preliminary technical documentation describing the system design and architecture.
- Initial orbital trajectory simulation results.
- Initial backend code covering the trajectory simulation and integration with the NASA NEO API.

B. Day 2: Complete Development, Validation, and Documentation

Day 2 Objectives:

- Complete development of the impact effects (crater, tsunamis, earthquakes).
- Integrate the NASA and USGS APIs for meteor and seismic data.
- Implement real-time interactivity.
- Execute testing and optimize performance.

Activities:

1) **Impact Effects Development**
   - Implement calculations for crater formation and meteor kinetic energy.
   - Simulate secondary impact effects (tsunamis and seismic activity) using USGS data.
2) **Integration of NASA and USGS Data**
   - Integrate the NASA NEO API to fetch near-Earth meteor data in real time.
   - Integrate the USGS Earthquake Catalog API to model the seismic and geological effects of the impact.
3) **Interactivity and Visualization Development**
   - Build interactive sliders so users can adjust meteor parameters (size, velocity, etc.).
   - Implement 3D visualization of the meteor trajectory and impact effects.
   - Update 2D maps in real time to display tsunami- and earthquake-affected areas.
4) **Testing and Optimization**
   - Conduct usability tests to ensure the tool is easy to use and understand.
   - Optimize performance in both the frontend (3D visualization) and the simulation calculations.

Day 2 Deliverables:

- Complete simulation of the meteor trajectory and impact effects (crater formation, tsunamis, earthquakes).
- Interactive interface with sliders to modify parameters and view real-time results.
- Final prototype of the web tool with 3D visualization and interactive 2D maps.
- Usability testing outcomes and performance optimizations.
- Final project documentation, including usage instructions and technical details.

References

[1] G. S. Collins, H. J. Melosh, and R. A. Marcus, "Earth impact effects program: A web-based computer program for calculating the regional environmental consequences of a meteoroid impact on earth," *Meteoritics & Planetary Science*, vol. 40, no. 6, pp. 817–840, 2005. doi:10.1111/j.1945-5100.2005.tb00157.x

[2] R. R. Herrick, "Updates regarding the resurfacing of venusian impact craters," in *Lunar and Planet. Sci. Conf. XXXVII*, p. Abs. 1588, Lunar and Planetary Institute, Houston, Texas, 2006.

[3] R. R. Herrick, V. L. Sharpton, M. C. Malin, S. N. Lyons, and K. Feely, "Morphology and Morphometry of Impact Craters," in *Venus II: Geology, Geophysics, Atmosphere, and Solar Wind Environment*, S. W. Bougher, D. M. Hunten, and R. J. Phillips, Eds., p. 1015, 1997.

[4] K. A. Holsapple, "The scaling of impact processes in planetary sciences," *Ann. Rev. Earth Planet. Sci.*, vol. 21, pp. 333–373, 1993.

[5] W. B. McKinnon and P. M. Schenk, "Ejecta blanket scaling on the Moon and Mercury - inferences for projectile populations," in *Lunar and Planet. Sci. Conf. Proceedings XVI*, pp. 544–545, Lunar and Planetary Institute, Houston, Texas, 1985.

[6] K. Wünnemann, G. S. Collins, and R. Weiss, "Impact of a cosmic body into earth's ocean and the generation of large tsunami waves: Insight from numerical modeling," *Reviews of Geophysics*, vol. 48, no. 4, 2010. doi:10.1029/2009RG000308

[7] H. J. Melosh, *Impact Cratering: A Geologic Process*. Oxford University Press, 1989.

[8] K. R. Housen and K. A. Holsapple, "Ejecta from impact craters," *Icarus*, vol. 211, pp. 856–875, 2011. doi:10.1016/j.icarus.2010.09.017

[9] T. R. McGetchin, M. Settle, and J. W. Head, "Lunar impact ejection and crater growth," *Journal of Geophysical Research*, vol. 78, no. 11, pp. 10847–10863, 1973. doi:10.1029/JB078i023p10847

[10] C. F. Chyba, P. J. Thomas, and K. J. Zahnle, "The 1908 Tunguska explosion: Atmospheric disruption of a stony meteor," *Nature*, vol. 361, pp. 40–44, 1993. doi:10.1038/361040a0

[11] O. P. Popova et al., "Chelyabinsk Airburst, Damage Assessment, Meteorite Recovery, and Characterization," *Science*, vol. 342, no. 6162, pp. 1069–1073, 2013. doi:10.1126/science.1242642

[12] C. N. Kingery and G. Bulmash, "Airblast Parameters from TNT Spherical Air Burst and Hemispherical Surface Burst," Technical Report ARBRL-TR-02555, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, 1984.

[13] UFC 3-340-02, "Structures to Resist the Effects of Accidental Explosions," U.S. Department of Defense, 2008 (Change 2, 2014).

[14] S. Glasstone and P. J. Dolan, *The Effects of Nuclear Weapons*. U.S. Department of Defense and U.S. Department of Energy, 1977.

[15] S. N. Ward and E. Asphaug, "Meteor impact tsunami: A probabilistic hazard assessment," *Icarus*, vol. 145, pp. 64–78, 2000. doi:10.1006/icar.1999.6336

[16] NASA Goddard Space Flight Center, "Elliptical Orbit Design," Mission Visualization https://nasa.github.io/mission-viz/RMarkdown/Elliptical_Orbit_Design.html, accessed Oct. 2024.