NASA Engineering and Safety Center
Technical Assessment Report

Expansion of Check-Cases for 6DOF Simulation
Volume I

TI-23-01853

NESC Lead Heather Koehler
Co-Technical Lead Matt Hawkins
Co-Technical Lead Jason Neuhaus

September 19, 2024

Report Approval and Revision History

NOTE: This document was approved at the September 19, 2024, NRB.

Approved: Timmy Wilson Digitally signed by Timmy Wilson Date: 2024.09.27 09:12:03 -04'00'

NESC Director

Version	Description of Revision	Office of Primary Responsibility	Effective Date
1.0	Initial Release	Ms. Heather Koehler NASA Technical Fellow for Flight Mechanics	September 19, 2024

Table of Contents

List of Figures

List of Tables

List of Appendices

Technical Assessment Report

1.0 Notification and Authorization

Zack Crues, of the Crewed Human Landing System (HLS) Interfaces for Piloting Working Group (CHIP-WG) Simulation Environments Task (SimTeam), requested that the NESC expand a previous NESC assessment, TI-12-00770 Development of Verification Data for Flight Simulation. That effort provided benchmark check-cases for well-specified, rigid-body, six-degree-of-freedom (6DOF) aerospacecraft models to promote consistent and accurate flight simulations across multiple Agency tools and facilities. The SimTeam regularly refers to that study and requested an expansion to include select lunar-centric simulation check-cases in support of new lunar exploration initiatives.

Request Submitted	February 14, 2023
Initial Evaluation Approved	March 23, 2023
Assessment Plan Approved	May 9, 2023
Team Kickoff Meeting	June 8, 2023
Stakeholder Briefing (Interim Status Report)	September, 2023
Stakeholder Briefing (Draft of Final Results)	August, 2024
Final Report Delivery and Stakeholder Update	September 19, 2024

The initial evaluation for this request was approved on March 23, 2023. Simulation teams across Centers were surveyed for interest in participating, starting with simulations that had participated previously in *Check-Cases for Verification of 6-Degree-of-Freedom Flight Vehicle Simulations*. The assessment plan was approved at a NESC Review Board on May 9, 2023. Heather Koehler, NASA Technical Fellow for Flight Mechanics, was assigned as the NESC Lead; Matt Hawkins of the Guidance, Navigation and Mission Analysis Branch (MSFC), and Jason Neuhaus of the Simulation Development and Analysis Branch (LaRC) were assigned as technical co-leads.

The HLS customer/stakeholder was provided an interim status briefing in September 2023. Additional stakeholders for this assessment are the simulation development teams at each Center.

2.0 Signatures

Submitted by: NESC Lead

Heather Koehler Digitally signed by Heather Koehler Date: 2024.09.24 15:13:33 -05'00'

Ms. Heather Koehler

Significant Contributors:

Matthew Hawkins Digitally signed by Matthew Hawkins Date: 2024.09.24 13:26:55 -05'00'

Dr. Matt Hawkins

Jason Neuhaus Digitally signed by Jason Neuhaus Date: 2024.09.24 15:01:44 -04'00'

Mr. Jason Neuhaus

Signatories declare the findings, observations, and NESC recommendations compiled in the report are factually based from data extracted from program/project documents, contractor reports, and open literature, and/or generated from independently conducted tests, analyses, and inspections.

3.0 Team Members

Name	Discipline	Organization (include host Center
Core Team		
Heather Koehler	NESC Lead	MSFC
Matt Hawkins	Technical Co-Lead	MSFC
Jason Neuhaus	Technical Co-Lead	LaRC
Tannen VanZwieten	Guidance, Navigation, and Control (GNC) Technical Discipline Team (TDT) Rep	KSC
Jesse Couch	Data Analytics	LaRC/AAG
Robert Bossinger	Data Analytics	LaRC/AAG
Kyler Brazukas	Data Analytics Intern	LaRC/AAG
Trey Fiebelkorn	Data Analytics Intern	LaRC/AAG
Edwin (Zack) Crues	Automation and Robotic Systems	JSC
James Gentile	Automation and Robotic Systems	JSC
Benjamin W. L. Margolis	Condor Analyst	ARC
Tristan Hasseler	DSENDS-DARTS Analyst	JPL
Abhinandan Jain	DSENDS-DARTS Senior Research Scientist	JPL
Marcus Lobbia	DSENDS-DARTS Analyst	JPL
Justin Ganiban	GLASS Analyst	MSFC
Nicholas Olson	GLASS Analyst	MSFC
Zu Qun Li	JEOD Analyst	JSC
Nathan Perreau	LaSRS++ Analyst	LaRC
Timothy Curry	MAVERIC Analyst	MSFC
Dane Erickson	MAVERIC Analyst	MSFC
Matthew Andreini	POST2 Analyst	LaRC
Alejandro Pensado	POST2 Analyst	LaRC
Zhiqiang (Joe) Zhou	STARS Analyst	LaRC
Wei Lin	Ast. Experimental Facility Development	ARC
Laura Maynard-Nelson	Technical Discipline Team Deputy, Software	GRC
John Penn	Trick Simulation Analyst	JSC
Michael Squire	NESC Principal Engineer / Sr. Technical Leader	LaRC
Curtis Zimmerman	Aerospace Vehicle Design and MSN Analysis	MSFC
Consultant		
Gregory Dukeman	Flight Mechanics Lead	MSFC
Bruce Jackson	Senior Research and Development Engineer	LaRC/AAG
Rafael Lugo	Aerospace Flight Systems	LaRC
Juan Orphee	GNC Lead	MSFC
Robert Shelton	Automation and Robotics Systems	JSC
Jeremy Shidner	Flight Mechanics	LaRC
Everett Bolduc	Flight Systems Training and Operations	JSC
Business Management		
Linda Moore	Program Analyst	LaRC/MTSO
Administrative Support		
Melissa Strickland	Project Coordinator	LaRC/CPSS
Linda Burgess	Planning and Control Analyst	LaRC/AMA
Christa Hahn	Technical Editor	LaRC/AMA

3.1 Acknowledgements

The team acknowledges and thanks the strong support and advocacy from HLS, specifically the Systems Engineering and Integration Lead Tom Percy, Spacecraft and Vehicle Systems Lead Leo Barreda, CrewCo Lead Nikki Williams, Flight Mechanics Lead Greg Dukeman, GNC Lead Juan Orphee, HLS Chief Engineer Rene Ortega, and Assistant Program Manager, Don Krupp. Without their support, many of the simulation teams would not have been available to contribute to this effort. We wish to thank Melissa Strickland, Christa Hahn and Linda Burgess and Linda Moore for excellent coordination, planning and editing that greatly improved the quality of the final product. Our technical consultants, Bob Shelton and Bruce Jackson, who brought tremendous experience from the previous related Flight Simulations Check-cases, guided us in the right direction and reminded us of past lessons learned to keep us on track. We also acknowledge the participation and insight of our peer reviewers: Ben Burger, Simulation Development Expert, MSFC; Soumyo Dutta, POST2 Development Expert, LaRC; Steven Gentz, NESC Chief Engineer, MSFC, Tim Barth NESC Integration Office; Dan Murri, Lead of previous check-cases assessment, LaRC; and Robert Beil, Technical Management, NESC Integration Office. Lastly, we acknowledge the larger simulation teams, their analysts and experts that participated in the discussions and simulation activities improving the overall product.

4.0 Executive Summary

The previous assessment, TI-12-00770 Development of Verification Data for Flight Simulation assessment, initiated in 2012, and completed in 2015, developed Earth-based benchmark check-cases for simulation verification [ref. 1]. It was noted in that assessment that many NASA Centers have independently developed preferred frameworks for flight simulation software, and that differences in model implementation and numerical approaches resulted in variations between simulations and resulting analyses. As more commercial companies provide services for NASA in the cislunar region, providing transportation elements, and/ or deploying orbital and lunar surface assets, using validated simulations is crucial to understanding the analyses referenced to meet Program requirements.

Utilizing benchmarking check-cases improves the simulations being assessed, reduces errors, builds confidence in solutions, and serves to build credibility of simulation results per NASA Standard 7009A Standard for Models and Simulations(M&S). This Standard specifies that comparing simulations is a way to validate system performance; this standard states: "Once the computational model is available, the next step is empirical validation, which is the comparison of M&S results with a referent (generally, data from either an operational simulation, or a 'representative system'). In some instances, e.g., for the development of so-called 'surrogate models', the referent can be the results obtained from a higher-fidelity (and typically computationally expensive) model." Often, validating a simulation against flight or test data can be challenging given the limited opportunities and resources to conduct flight tests, and demonstrating those tests in the desired flight environments and conditions. Cross-comparisons with different models and simulations with different implementations given the same inputs is a valid domain for simulation validation.

This effort seeks to compare multiple simulation tools while utilizing a common defined set of inputs. Comparing results to previously published simulation outputs to evaluate reasonable-ness and accuracy of model implementation can be helpful in validating simulations and can improve overall credibility of the simulation and system being modeled. Simulation comparisons can benefit from utilizing common standards for defining parameters and sharing models and reduces the occurrence for implementation errors.

5.0 Assessment Plan

The original plan included development of benchmark scenarios to obtain simulation comparisons of trajectories covering multiple phases of flight for lunar exploration class missions: Near-rectilinear Halo Orbit (NRHO); Low Lunar Orbit (LLO); Deorbit, Descent, and Landing (DDL); lunar ascent; and Rendezvous, Proximity Operations, and Docking (RPOD).

This assessment builds on a prior assessment that included 3 degrees-of-freedom (3DOF) and 6DOF comparisons of a variety of vehicles in orbit or in maneuvering flight. The prior assessment used Earth-centered cases, while the current study is focused on the cislunar environment. The prior assessment included several complex cases using propulsive forces, active maneuvering, and a full GNC model. For the current assessment, it was desired to avoid repeating complex cases that were already thoroughly examined, and to devise a smaller, focused set of cases that exercise new and unique features of the lunar environment. To maximize the usefulness of the assessment, it is recommended that users implement cases of interest from NASA/TM-2015-218675, the previous assessment. The cases were selected to be representative of the variety of orbital conditions typically encountered during lunar mission simulation. As the Moon lacks an appreciable atmosphere, only different orbits need to be considered, and no atmospheric or aerodynamic modeling is required.

This study targets a lesser scope; it is limited to space flight for two reference vehicle models. Some scenarios involve primarily long-duration orbital propagation without maneuvering or use of thrusters. Originally, other scenarios were to focus primarily on shorter-duration maneuvers involving orbital changes, descents to landing, ascents from the lunar surface, rendezvous.

The plan was modified to remove closed-loop guidance, navigation, and control (GNC) cases associated with DDL, ascent, and RPOD phases given the resource limitations and complexity associated with developing and sharing a complex closed-loop GNC model.

As in the previous effort, which involved six NASA simulation tools and one open-source flight simulation tool, this effort involved eight NASA simulation tools. An additional similarity for both efforts is that each participating Center found opportunities to make changes which improved the quality of their simulation, “The study paid for itself.”

6.0 Problem Description and Background

The Crewed HLS Interfaces CHIP-WG SimTeam meets regularly to discuss all topics related to HLS M&S. A recurring area of interest and need involves the comparison of modeling and simulation tool sets used across the Programs for the analysis of manual control. The principal tools are located at JSC, LaRC, and MSFC, but others are available across the agency. The CHIP-WG SimTeam often refers to the previous 2015 NESC study (NESC-RP-12-00770) associated with 6DOF check-cases for flight vehicle simulations but, there are not any cases for lunar-centric simulations. Lunar-based check-cases would support the future Artemis mission Program and commercial lunar landers as they seek to simulate complex integrated systems and demonstrate requirements and objectives are met through analysis.

Using benchmarking check-cases improves the simulations being assessed, reduces errors, builds confidence in solutions, and serves to build credibility of simulation results per NPR 7009 Models and Simulation Standard. As in industry, multiple simulation tools are developed within each NASA Center that are somewhat incompatible in terms of moving a spacecraft model

between them. Various simulations provide for cross-comparisons, but incompatibility limits and slows cooperative studies of any particular vehicle model. This effort seeks to compare multiple simulation tools while utilizing a common defined set of inputs, sharing models across simulation platforms was not in scope due to the limited time and availability of the team members to implement but some concluding thoughts are offered to improve this more complex problem.

7.0 Analysis

7.1 Participating Simulations and Descriptions

Eight NASA simulations participated in this assessment. The simulations are described below and are developed and maintained from Ames Research Center, JPL, JSC, LaRC, and MSFC. While some simulations are available to the public, some require licensing and authorization to use in direct support of NASA missions.

7.1.1 Condor Flight Vehicle Toolkit

Condor Flight Vehicle Toolkit is a flight vehicle simulation tool built using Condor [ref. 1], an open-source software library for the rapid development of mathematical models of complex systems in Python. Condor was developed in the Ames Research Center's Systems Analysis Office to address rapidly evolving analysis tasks across a wide range of applications, including conceptual aircraft design performance analysis and robust orbital trajectory design. Model templates are used to define models from different categories, e.g., systems of algebraic equations, table lookups, optimization problems, and trajectory analyses of systems of ordinary differential equations (ODEs) with events. These models can be used as building blocks to assemble more complex system models.

Condor Flight Vehicle Toolkit is a translation of SimuPy's [ref. 2] Flight Vehicle Toolkit [ref. 3] into custom model templates for planetary bodies and flight vehicles. The planet model templates enable users to specify the planet characteristics including planetodesy, gravitation, and atmosphere. The vehicle model template allows the construction of any number of vehicles parameterized by inertial properties, aerodynamics, user-defined force models e.g., propulsion systems or other actuators, and flight software. The 6DOF equations of motion are then automatically compiled into efficient numerical code using the symbolic backend casadi [ref. 4] and then passed to CVODE [ref. 5] or SciPy's dopri5 or dop853 ODE solvers [ref. 6] with an event-handling wrapper. Trajectories may be optimized using an implementation of a sweeping gradient method for ODEs with events [ref. 7] and one of several optimization problem solvers. All ODE solvers currently supported in Condor use adaptive step-sizes and are sampled for solution comparison.

Condor was written to be modular, allowing alternate symbolic backends or solvers to be used with minimal additional effort. An emphasis on employing off-the-shelf, third -party, open-source software packages (widely used in Python's scientific computing and machine learning communities) reduces the development burden of the Condor team while leveraging the open-source software community's inherent advantages of efficient computing performance and verification and validation histories.

7.1.2 Dynamics Simulator for Entry, Descent, and Surface Landing (DSENDS-DARTS)

DSENDS – DARTS (Dynamics Algorithms for Real Time Simulation) is a multi-mission flight dynamics and simulation tool for closed-loop flight dynamics and atmospheric Entry, Descent, and Landing simulations. DSENDS-DARTS belongs to the family of spacecraft and robotics system simulation tools developed by the DARTS Lab [ref. 8] at NASA's Jet Propulsion Laboratory (JPL).

DSENDS-DARTS has been in active development and use [ref. 9] since its early beginnings with the Cassini space probe in the early 1990s. In recent years. DSENDS-DARTS has been used by several flight missions and technology development efforts including the Mars Phoenix, Mars Science Lab, InSight and Mars 2020 landers, the Low-Density Supersonic Decelerator technology demonstrator, and the precision landing technology development and asteroid retrieval mission concept development (ARRM). A collaboration with Johnson Space Center (JSC) Flight Operations Directorate team is using DSENDS-DARTS for ascent, descent simulations for Space Launch System (SLS) and Orion Multi-Purpose Crew Vehicle (MPCV), including its use for day of launch simulations for the Artemis I mission.

DSENDS-DARTS is part of the larger DARTS simulator family which provides a wide domain of applications including autonomous robots, ground vehicles (ROAMS-DARTS), and rotorcraft (HeliCAT-DARTS). This family of tools builds upon middleware like the DARTS rigid/flexible multibody dynamics engine and the DARTS Shell (Dshell) simulation framework. All DARTS simulators use C++ for speed but feature a rich and user-friendly Python interface.

DSENDS-DARTS was used to exercise all the check-cases in the present study. It used the Dormand-Prince adaptive step size Runge-Kutta 45 integrator provided in the freely available ARKODE SUNDIALS package developed by Lawrence Livermore National Laboratory. Because Dormand-Prince is an adaptive integrator, absolute and relative error tolerances were used to control integration error instead of simulation frame rate. Tolerances were set to 1e-10 or tighter for each case, which was a tolerance observed to provide close agreement with the analytical solution for the first check-case.

7.1.3 GeneraLized Aerospace Simulation in Simulink (GLASS)

The GeneraLized Aerospace Simulation in Simulink (GLASS) is a 6DOF vehicle simulation developed at MSFC. GLASS is designed to be a modern, modular, user-friendly, and efficient 6DOF simulation tool for aerospace vehicles and is built within the MathWorks MATLAB® and Simulink environment. This enables engineers to use the pre-built and tested toolboxes to speed up their development effort. To further simplify the development work for engineers, the Simscape Multibody toolbox forms the basis for the 6DOF equations of motion. Simscape Multibody generates the equations of motion internally and provides a multibody simulation environment using joints, bodies, force elements, and sensors.

GLASS provides a layer of abstraction over the base Simscape Multibody blocks that simplifies the building of a vehicle by combining multiple Simscape Multibody blocks into single-masked GLASS blocks. These parameterized blocks ensure models are constructed uniformly regardless of the vehicle design and have been compiled in a Simulink library. This Simulink library has taken the title of 'GLASS_Core' to note that these blocks form the core of any simulation built using the GLASS framework.

The HLS GNC Insight team uses GLASS to test and validate GNC algorithms for the contractors working the HLS program. GLASS has modeled multiple vehicles and multiple GNC algorithms within the framework, showing its flexibility. GLASS has also been validated against the Earth-based NESC 6DOF Orbital check-cases prior to this new study.

All the lunar check-cases were exercised in GLASS utilizing an Runge-Kutta fourth-order integration scheme (ode4) with 0.1-sec time steps.

7.1.4 JSC Engineering Orbital Dynamics (JEOD) Software Package

The Johnson Space Center (JSC) Engineering Orbital Dynamics (JEOD) Software Package [ref. 19] is an open-source simulation tool designed to work with the NASA Trick Simulation Environment that provides vehicle trajectory generation by the solution of a set of numerical dynamical models. These models are subdivided into four categories: environment models representing the conditions surrounding the vehicle; dynamics models for integrating the equations of motion; interaction models representing vehicle interactions with the environment; and a set of mathematical and orbital dynamics utility models.

JEOD is designed to simulate spacecraft trajectories in flight regimes ranging from low Earth orbit to lunar operations, interplanetary trajectories, and other deep space missions. JEOD can be used to simulate a stand-alone spacecraft trajectory and attitude state, or it can be interfaced with a larger simulation space, e.g., coupling with spacecraft effectors and guidance, navigation, and control GNC systems. More than one spacecraft can be simulated about one central body or separate spacecraft about separate central bodies.

JEOD has been in development since the early 1990s, and has been managed by various engineering groups at JSC throughout its lifecycle. The Space Shuttle Program and the International Space Station have used JEOD for years, and now, Orion MPCV, and HLS, and other programs, plan to use it.

7.1.5 Langley Standard Real-Time Simulation (LaSRS++)

The Langley Standard Real-Time Simulation in C++ (LaSRS++) is an object-oriented framework for construction of aerospace vehicle simulations. The Simulation Development and Analysis Branch [ref. 20] uses LaSRS++ simulations to support desktop analysis, hardware in-the-loop simulations, and high-fidelity, human-in-the-loop simulators. Projects using LaSRS++ have modeled planetary landers, crewed spacecraft, launch vehicles, RPOD, planetary aircraft, advanced concept aircraft, commercial transport aircraft, military fighters, and unmanned aerial vehicles.

LaSRS++ was employed for the orbital check-cases. In all cases, the integration methods used were the same, but customized depending upon which state was being propagated, as shown in Table A below. The selected LaSRS++ frame rate of 1000 Hz minimized the integration error differences between the LaSRS++ solution-data and the truth dataset for the first check-case. That frame rate was used for all ensuing check-cases. (LaSRS++ simulations used with human-in-the-loop real-time simulations are normally operated with frame rates of 100 Hz or less.)

Table A. LaSRS++ Integration Methods for Orbital Check-cases

States	Method
Translational velocity	Second-order Adams-Bashforth
Position	Second-order Taylor series
Angular velocity	Second-order Adams-Bashforth
Quaternion Attitude	Local linearization algorithm [22]

7.1.6 Marshall Aerospace Vehicle Representation in C (MAVERIC)

Marshall Aerospace Vehicle Representation in C (MAVERIC) is a low-to-high fidelity 3DOF/6DOF vehicle flight simulation program developed at MSFC, written primarily in the C and C++ programming languages. MAVERIC was designed to be generic, and data driven and can provide for the rapid development of end-to-end vehicle flight simulations for a variety of launch or on-orbit scenarios. MAVERIC vehicle simulation models are layered upon a set of foundational software called TFrames.

TFrames is a time-based differential equation solver environment. TFrames provides an environment for developing a dynamic simulation that insulates the simulation developer from the programming details associated with numerical integration, discrete data sampling, table look-ups, etc. High-level routines provide convenient interfaces between the simulation code and the numerical integration engine.

MAVERIC was developed with data structures, code, and interfaces for standard vehicle components and is designed to easily incorporate and interface with customized models. MAVERIC has been in active development at MSFC since the mid-1990s and has been used to model over a dozen different aerospace vehicles. Currently, MAVERIC is the primary 6DOF simulation for the SLS ascent phase and GNC design.

7.1.7 Program to Optimize Simulated Trajectories II (POST2)

POST2 is a generalized 3/6/multi-DOF event-based trajectory simulation software that is descended from the original POST developed by Martin Marietta in 1970. POST2 provides the capability to simulate, target, and optimize point mass trajectories for multiple powered or un-powered vehicles near an arbitrarily rotating, oblate planet. POST2 has been used successfully to solve a wide variety of atmospheric ascent and re-entry problems, as well as exo-atmospheric orbital transfer problems. This flexible simulation capability is augmented by an efficient, discrete parameter-optimization capability that includes equality and inequality constraints.

POST2 include many generalized models for atmosphere, gravity, and propulsion that are used to simulate a wide variety of launch, orbital, and entry missions. Users may also provide custom models of varying fidelity, including flight software. POST2 can simulate multiple vehicles in a single simulation, each with independently defined environments, attracting body characteristics, and flight software. Thus, each vehicle can have its own GNC system for completely independent, on-board autonomy. Additionally, effects of multi-body and interaction forces that depend on the relationship of one vehicle to another can be included. POST2 also supports 3DOF and 6DOF trajectories within the same simulation.

POST2 has extensive flight heritage, and its core codebase is considered flight-validated via pre- and post-flight analysis of dozens of Earth and Mars Entry, Descent, Landing (EDL) missions and ground/flight tests. This includes all NASA Mars entry systems since Pathfinder, and all

elements of the Mars Sample Return Campaign (Sample Retrieval Lander, Mars Ascent Vehicle, and Earth Entry System). POST2 is the primary simulation tool to model various NASA flight tests such as Low-Earth Orbit Flight Test of an Inflatable Decelerator and Advanced Supersonic Parachute Inflation Research Experiment, EDL for ongoing missions such as Deep Atmosphere Venus Investigation of Noble gases, Chemistry, and Imaging and Dragonfly, and is used as human spaceflight Insight and Independent Verification and Validation (IV&V) for Human Landing System (HLS) Lunar descent and ascent, Commercial Crew Program (CCP) deorbit and entry, and Space Launch System (SLS) ascent. In these capacities, POST2 has been used across all phases of flight, from Pre-Phase A through F, and been both Prime and IV&V tool for flight vehicle design and integrated performance assessments.

For the lunar check-case effort, POST2 used a fourth-order Runge-Kutta integration method at 100 Hz.

7.1.8 Space Transportation and Aeronautics Research Simulation (STARS)

STARS is MATLAB/Simulink-based air, launch, and space vehicle dynamics simulation and GNC design software. STARS takes advantage of MATLAB/Simulink capability and flexibility to make the creation of the simulation and GNC design much faster. STARS enables the time domain simulations of the vehicle dynamics with the GNC system to be conducted. In addition, the Bode and Nichols plots of the frequency domain models can be obtained in STARS to conduct GNC design.

STARS has been used for the SLS vehicle dynamics simulation and GNC design. SLS STARS is aimed to predict the overall vehicle dynamics of the SLS launch vehicle stack, including rigid-body and flexible modes, as well as propellant slosh dynamics, dynamics associated with the engine actuation and gimballing, and sensor dynamics. Additionally, interactions between many of these effects are also modeled. All of these effects are modeled in SLS STARS, and many of them can be enabled or disabled for a given run.

7.2 Overview of Check-Cases

In general, the basic cases were selected to build in an up-and-out fashion, starting from a simple Keplerian low lunar orbit, then adding effects e.g., a detailed gravitational field and third-body perturbations. Typically, a minimal implementation of additional effects is used to best permit future simulation comparisons. This section gives an overview of the check-cases, including the reasoning for including each check-case. Implementation details follow in further sections.

The first cases (1, 2, and 3) use a near-circular low lunar orbit with a high inclination. The specific orbit was chosen to be like orbits that the participating simulation teams are regularly using in support of the Artemis campaign Program to land near the lunar south pole. Case 1 is a simple Keplerian propagation, which allows comparison with a known analytical solution (see Section 7.7.1). Case 2 uses the Gravity Recovery and Interior Laboratory (GRAIL) gravity model to order and degree 8 (8x8 GRAIL), allowing testing of the actual gravitational model most lunar simulations will use. Case 3 uses a high-fidelity GRAIL gravity model to order and degree 320 (320x320 GRAIL), allowing testing of ingesting and using a complex gravitational model. Case 1 was created explicitly from the desire to exactly match the known analytical solution data at the start time. Case 1 is the only cases where simulations can be compared directly with the closed form solution.

The 8x8 GRAIL model is used for all cases after case 3. For actual lunar simulations, a higher-order GRAIL model is generally recommended, with considerations made for the length of the simulation time and proximity to the Moon. However, for the purposes of this assessment, the 8x8 model is used to best allow simulation-to-simulation comparisons. As will be seen, simulation-to-simulation differences naturally grow as higher order models are implemented, so an 8x8 model ensures that a simulation is successfully implementing the GRAIL model while minimizing differences due to the GRAIL model itself.

Only the spherical harmonic gravity model is used in this assessment. Detailed simulation of local terrain may require a different type of gravity model, for example one based on mass distribution. This assessment is limited to cases where a spherical harmonic model is appropriate.

Cases 4 and 5 use a near-circular high lunar orbit with an altitude of approximately 500 km. This is the altitude where perturbations from the Earth start to become important. Case 4 uses no third-body perturbations, establishing a baseline before third-body effects are introduced. Case 5 uses perturbations from the Earth and the Sun, allowing testing of these perturbation effects.

Only Earth and Sun perturbations are used in this assessment, as they are the two dominant perturbation effects. Although most simulations will have several third-body perturbations available, limiting the study to Earth and Sun is sufficient to test the implementation and minimizes the risk that a given simulation does not have a particular third-body perturbation available.

Cases 6 and 7 introduce a highly elliptical orbit, with initial perilune of approximately 250 km and initial apolune of approximately 9,385 km. These cases include a range of dynamics, including apolune with higher orbital velocity where the GRAIL model is most important, to perilune with lower orbital velocity where the third-body perturbations are at their most significant.

Case 8 utilizes a NRHO that is of interest for the Artemis program and presents interesting astrodynamics given the nature of this orbit as a periodic quasi-stable family of orbits around the Earth-Moon Lagrange points. The NRHO gets to within an altitude of 2000 km at closest approach, to nearly 70,000 km at its furthest point. The NRHO is well beyond the regime of two-body astrodynamics and approximations, and alternate orbital elements may need to be utilized.

Finally, Case 9 is a nearly polar orbit of similar size to the low lunar orbit. The two most common coordinate systems, Mean Earth (ME) and Principal Axis (PA), have slightly different equatorial planes, therefore the inclination of an orbiting vehicle is different in each system. The purpose of the polar orbit is to be near the origin of the GRAIL model, which is expressed in the PA system. This case is nearly polar in the PA frame.

Two different vehicle models were defined for the check-cases. The first vehicle is a simple cylinder, which matches the cylinder model in the original check-cases. The cylinder is relatively easy to implement and tests basic rotational dynamics and modeling. The second is the Apollo spacecraft, drawn from documented Apollo mass properties. The Apollo spacecraft is asymmetric and has off-diagonal moment of inertia terms. The Apollo spacecraft tests more complex dynamics and provides a means to check sign conventions and coordinate systems.

Typically, the vehicle has an initial pitch rate that is matched to the orbital period. For circular orbits, the flight-path angle remains approximately constant, while for elliptical orbits the flight-

path angle will deviate before returning after one orbit. Certain cases use other initial rates, as described in the following paragraphs.

The initial orientation of the vehicle body frame is specified relative to a Vehicle-carried Orbit-defined (VO) frame for the first 7 cases. Cases 8 and 9 explicitly give the vehicle orientation in the Moon-centered Inertial (MI) system, an inertial coordinate system. It is typical to express the orientation with respect to a vehicle-carried system, however propagations are typically done in an inertial system. Cases 1 through 7, then, include an initial transformation from the VO frame to the internal inertial representation, while Cases 8 and 9 allow direct comparison of propagation through specifying the underlying inertial states directly.

After defining the basic check-cases, several interesting variations were discussed during working meetings and selected for inclusion. It was determined that there was little value in defining a new orbit for each variation, so instead the new cases were based on existing cases with a letter appended to the case number.

Case 5 tests a high circular with third-body perturbations, and includes the typical pitch rate for approximately constant flight-path angle. A case that tumbles about all three axes was desired, so Case 5A repeats Case 5 with initial rotation rates along all three axes. The additional rotations were added to expand the tests to include testing of signs on cross products of inertia.

Case 6 tests a highly elliptical orbit. The initial pitch rate is selected to match the orbital period, however since the orbit is elliptical, the flight-path angle nods up and down during one orbit with respect to the VO frame. A case that includes no inertial rotations was desired, so Case 6A repeats Case 6 with zero inertial rotation. Most simulations will likely propagate in an inertial frame, so Case 6A should, in effect, test integrating zero. In practice, simulation details vary for high-fidelity simulations, so very small differences are not unexpected.

Case 8 tests the NRHO, which is of great interest for the Artemis program. The NRHO is nearly stable, but spacecraft will eventually escape without station-keeping maneuvers. To explore the sensitivities of the NRHO, four additional cases were defined. Case 8A re-initializes at a true anomaly of 180 degrees, after approximately half of the orbital period. Case 8B re-initializes at a true anomaly of 0 degrees, after nearly one orbital period. Case 8C perturbs the initial conditions of Case 8 by adding 10 meters to the initial position vector. Case 8D perturbs the initial conditions of Case 8 by adding 0.1 m/s to the initial velocity vector.

Case 9 tests the polar orbit. Cases with a sensor station offset from the center-of-mass were desired, so two sensor stations were defined. Case 9 uses a sensor station that is offset only in the body X-direction. Case 9A and 9B use a sensor station that is offset from the center of mass (CoM) in all three body directions. Additionally, Case 9B introduces an active moment profile. A moment profile was defined to apply to various combinations of body axes and is specified as both a table and with pseudo-code.

In addition to the sensor stations, cases that ingest digital elevation models (DEMs) were desired. Case 9 includes two test points, one fixed and one that moves as a function of time. For the specified DEM, altitude, latitude, and longitude are output, testing the ingestion of the DEM.

Cases 8 and 9 specify the vehicle orientation differently than the others. Cases 1 to 7 all specify the initial orientation of the body frame with respect to the VO frame. Cases 8 to 9 were modified to allow more of a focus on pure propagation differences by explicitly stating the initial quaternion with respect to MI to use for the body frame orientation. It is assumed that the

underlying equations of motion (EOM) state is the attitude with respect to MI for all simulations. Cases 1 to 7 may include effects from differences in converting from VO to MI relative attitude at t=0 and propagate this difference throughout the run. Cases 8 to 9 should represent a purer comparison of propagation techniques.

Table B below summarizes the cases, including the orbit and additional information on the effects under study. Complete definitions of each case can be found in Section 7.7.

Table B. Summary of Effects Tested by Each Case'

Case	Orbit	Vehicle	Sun/Earth Perturbations	Additional Notes
1	LLO (Low Lunar Orbit) ~120x120km	Cylinder	No	Keplerian gravity, permits analytical solution
2	LLO	Cylinder	No	Introduces 8x8 GRAIL
3	LLO	Cylinder	No	320x320 high-fidelity GRAIL
4	HLO (High Lunar Orbit) ~500x500km	Apollo	No	Introduces Apollo vehicle model
5	HLO	Apollo	Yes	Introduces third-body perturbations
5A	HLO	Apollo	Yes	Body tumbles about all three axes
6	HEO (Highly Elliptical Orbit) ~250x9385km	Cylinder	Yes	Re-visits cylinder model
6A	HEO	Cylinder	Yes	Zero inertial angular rotation
7	HEO	Apollo	Yes	Returns to Apollo vehicle model
8	NRHO (Near Rectilinear Halo Orbit)	Apollo	Yes	Introduces NRHO orbit (radius ranges from ~2000km to ~70,000km)
8A	NRHO	Apollo	Yes	Re-initializes at a true anomaly of 180 degrees, after approximately one half of the orbital period
8B	NRHO	Apollo	Yes	Re-initializes at a true anomaly of 0 degrees, after approximately one complete orbital period
8C	NRHO	Apollo	Yes	Initial radius perturbed by +10m relative to case 8
8D	NRHO	Apollo	Yes	Initial velocity perturbed by +0.1 m/s relative to case 8
9	LPO (Lunar Polar Orbit) ~120x120km	Apollo	Yes	Polar orbit, includes a sensor station offset from the CoM along one direction; tests DEM ingestion
9A	LPO	Apollo	Yes	Includes sensor station offset from CoM in all three directions
9B	LPO	Apollo	Yes	Includes open-loop moment profile

Other cases considered, but ultimately not done due to insufficient resources and schedule, are listed here for consideration as forward work:

1. Landing (closed loop)
2. Rendezvous (closed loop)
3. Orbit raising (closed loop)
4. Ascent (open loop)
5. Apogee raising (open loop)

All these cases included some number of prescribed forces and moments and/or force and moment inputs as a function of state.

The original check-cases [ref. 12] included a variety of cases with prescribed forces and moments. It is recommended that the analyst implement the original check-cases. One case from the current assessment, Case 9B, included an open-loop moment profile, as this case tested new effects that were not covered in the original assessment.

Several of the proposed cases would utilize closed-loop forces and moments as a function of measured states. In addition to the complexities of defining even a simple closed-loop function, the check-cases by their nature will have slightly different measured states. Using these measured states to drive forces and moments introduces additional differences in output states with no clear way to determine the contribution of the closed-loop inputs.

7.3 Ground Rules and Assumptions

7.3.1 Vehicle Models

A set of reference vehicles were used for the check-cases, based on non-proprietary sources.

All vehicles defined here are defined by their equivalent body coordinate systems (see Section 7.3.3.3 for the body coordinate system definition). X, Y, and Z are in reference to the equivalent axis of the body coordinate system.

All cross products of inertia are specified using the positive integral definition, as in the following x-z cross product term:

$$I_{xz} = +\int x \cdot z \, dm, \text{ etc.} \qquad \text{(Equation 1)}$$

7.3.1.1 Cylinder

The cylinder is of uniform density and size 12m x 1m x 1m as shown in Figure 1. Mass properties and moments of inertia about the center of gravity are given in table C.

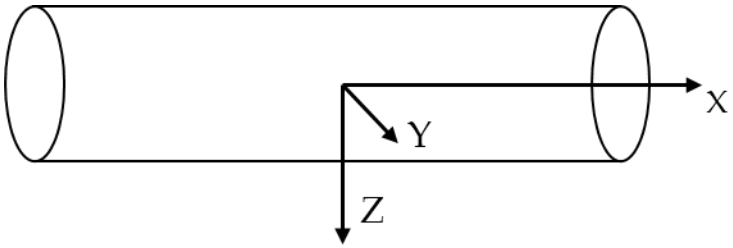

Figure 1. Cylinder Body Frame (not to scale)

Table C. Cylinder Mass Properties

Cylinder (Uniform Density)	
I_{xx}	500 kg m^2
I_{yy}	12,250 kg m^2
I_{zz}	12,250 kg m^2
I_{xy}	0.0 kg m^2
I_{yz}	0.0 kg m^2
I_{zx}	0.0 kg m^2
m	1000 kg

7.3.1.2 Apollo Lunar Module

The Apollo Lunar Module was used as model as shown in Figure 2. The mass properties and moments of inertia about the center of gravity are given in Table D.

It should be noted that the source document used English Engineering Units; for this study, the values were converted to *approximate* metric equivalents. Implementing the original source model verbatim will lead to differences [ref. 10].

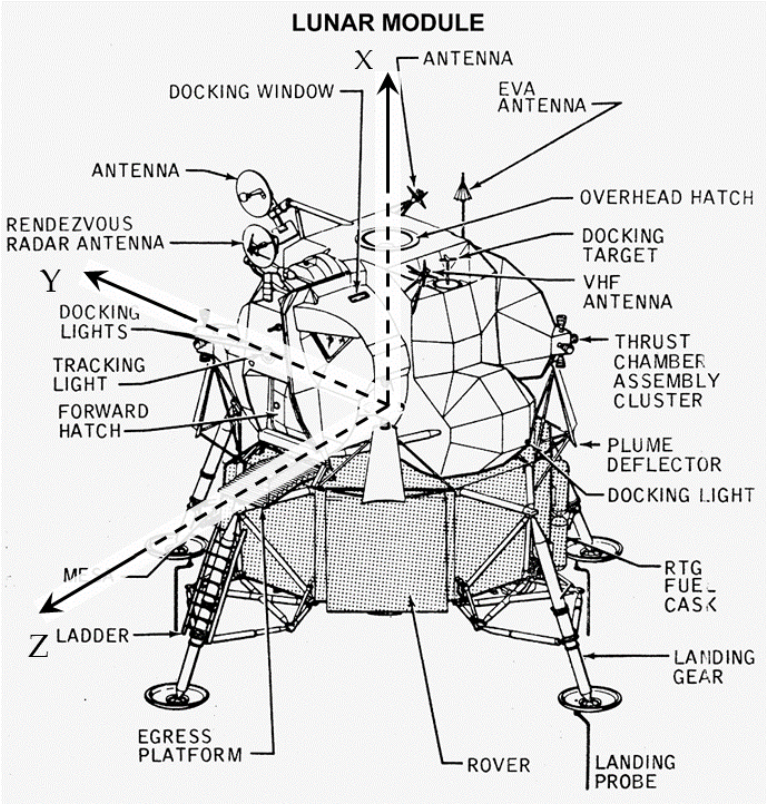

Figure 2. Apollo Lunar Module Body Frame (not to scale)

Table D. Apollo LM Mass Properties

Apollo Model	
Ixx	36,502.7 kg m2
Iyy	38,372.4 kg m2
Izz	36,514.9 kg m2
Ixy	-27.1 kg m2
Iyz	1,152.4 kg m2
Izx	233.2 kg m2
m	16,642.0 kg

7.3.2 Lunar Geodesy Models

This section describes the way in which the flight simulation tools model the surface of the Moon.

Unique to the lunar environment is also the existence of two planet-fixed frames. Care should be taken with these frames as the Moon is unique in this aspect and use of one of the frames is

sometimes assumed without explicitly stating the frame used, possibly resulting in errors that can be easily overlooked due to the small differences in the orientation of the two frames. This small different in orientation can result in differences on the order of hundreds of meters on the lunar surface. These two planet-fixed frames are the Mean Earth System (ME) frame and the Principal Axis (PA) System frame and are described in the following sections.

7.3.2.1 Spheroidal Moon/International Astronomical Union (IAU)

In this study, the Moon was modeled as a perfect sphere with the radius defined by the IAU as shown in Figure 3. [ref. 13] The radius of the spheroidal Moon used for this study was 1737.4 km (r_M). The notation of 'IAU' is used in this assessment to refer to data that is referenced to this spheroid.

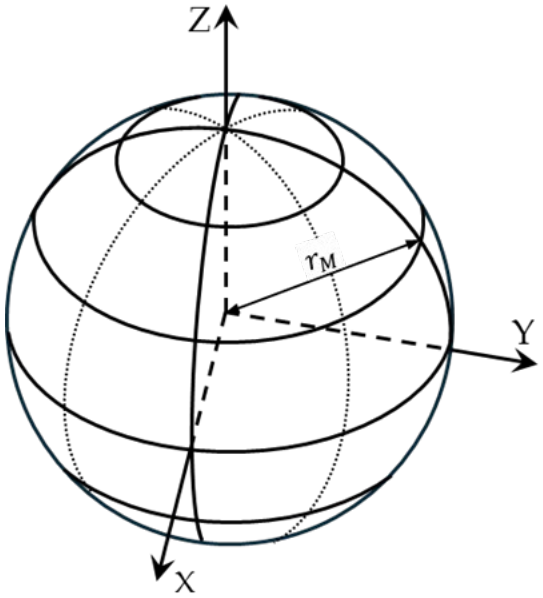

Figure 3. Reference Spheroid for IAU 2015

7.3.2.2 Mean Earth/MEM

One of the planet fixed coordinate systems used is ME. The orientation of the ME frame with respect to MI is defined by the Development Ephemerides (DE) products. See section 7.3.5 below for the Development Ephemerides products used in this assessment. The ME frame aligns the X axis at the average point on the surface pointing towards Earth as defined by the DE products. The notation of 'MEM', ME Moon, is used in this study to refer to this Moon-centered, planet-fixed coordinate system.

The following parameters define the ME planet fixed frame as shown in Figure 4.

- Centered at the Moon's center
- The X axis points from the Moon's center to the intersection of the ME Equator and Prime Meridian.
- The Z axis points from the Moon's center to the ME north pole

- The Y axis is defined Y = Z × X

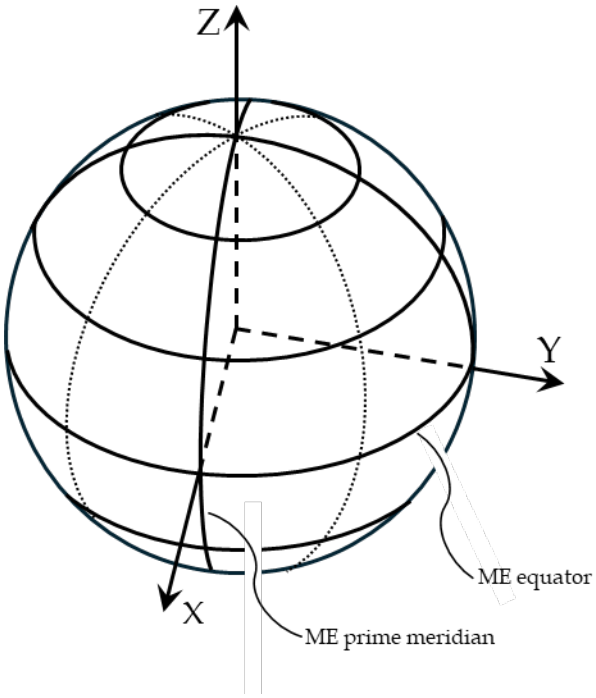

Figure 4. Mean Earth Planet-Fixed Frame

7.3.2.3 Principal Axis/PAM

The second planet fixed coordinate systems used is PA. The orientation of the PA frame with respect to MI is defined by the Development Ephemerides products. See Section 7.3.5 for the DE products used in this assessment. The PA frame aligns the axes with the principal axes of rotation of the Moon as defined by the DE products. The notation of 'PAM', PA Moon, is used in this study to refer to this Moon-centered, planet-fixed coordinate system.

The following parameters define the PA planet fixed frame as shown in Figure 5.

- Centered at the center of the Moon
- The X axis points from the center of the Moon to the intersection of the PA Equator and Prime Meridian.
- The Z axis points from the center of the Moon to the PA north pole
- The Y axis is defined Y = Z × X

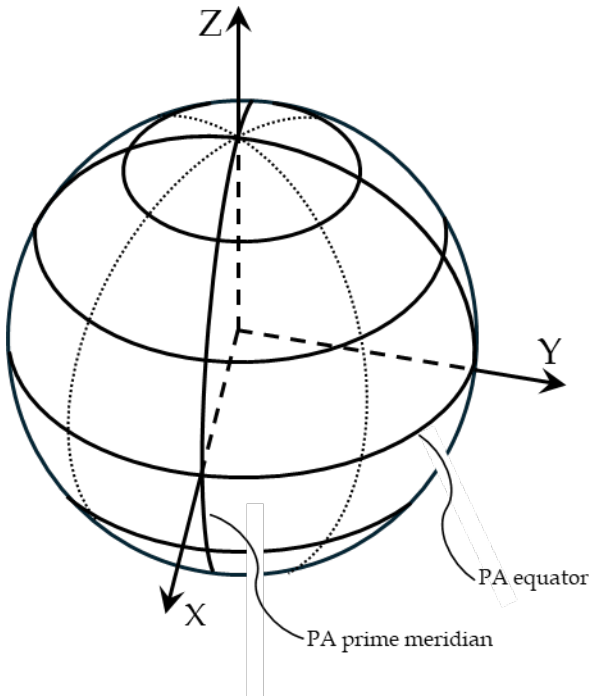

Figure 5. Principal Axis Planet-Fixed Frame

7.3.2.4 Digital Elevation Model (DEM)

For one of the cases, a Digital Elevation Model (DEM) lookup was added to test a specific set of data points. The output 'altitudeIauOfTp2_m' (see Section 7.4.4) is a lookup in the DEM data of the test point 2 location (see Section 7.7.15).

DEM data for south pole came from the following website for 80 m/pixel DEM data for -80 deg to the south pole, and is shown in Figure 6:

LOLA South Pole GDR information
https://imbrium.mit.edu/BROWSE/LOLA_GDR/POLAR/SOUTH_POLE/

JP2 Files:

LDEM_80S_80M.JP2
LDEM_80S_80M_JP2.LBL
LDEM_80S_80M_AUX.XML

IMG Files:

LDEM_80S_80M.IMG
LDEM_80S_80M.LBL

Data Set ID: LRO-L-LOLA-4-GDR-V1.0
Product ID: LDEM_80S_80M
Instrument Host Name: LUNAR RECONNAISSANCE ORBITER
Instrument Name: LUNAR ORBITER LASER ALTIMETER
Instrument ID: LOLA

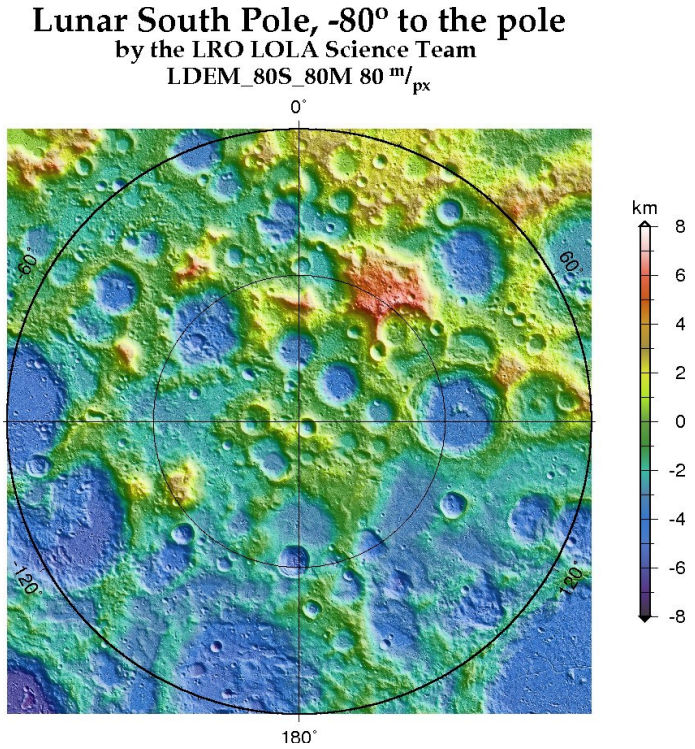

Figure 6. LOLA South Pole GDR, DEM Height Map

7.3.3 Coordinate Systems

An important part of developing dynamic simulations is dealing with coordinate systems. As in the previous study, there are several such systems involved in this assessment.

Of prime interest is the selection of the system that is inertial, or non-moving. While in real-life no such system exists, lunar orbit simulation tools typically use a pseudo-inertial system whose origin translates with the planet center and whose axes are fixed with respect to the stars, such as the Earth Mean Equator Mean Equinox J2000 (EMEJ2K) coordinate system described in the following section. The Moon is a rotating object in the EMEJ2K system, and its orientation with respect to the EMEJ2K system is determined through a time-dependent transformation.

Selection of the inertial coordinate system is important to avoid introducing errors in the calculation of derivatives (linear velocities and accelerations), or equivalently, in performing numerical integration of rotational states, due to rotational effects.

7.3.3.1 EMEJ2K/MI

The inertial coordinate system used for this study was the EMEJ2K coordinate system. The EMEJ2K is not a true inertial system because its origin is accelerating with the Moon. The key ‘inertial’ feature of the EMEJ2K frame is that the axes do not rotate. The EMEJ2K coordinate system is aligned with the J2000 coordinate frame and centered at the center of the Moon spheroid. The notation of ‘MI’, Moon-centered Inertial, is used in this study to refer to this Moon-centered Inertial coordinate system.

7.3.3.2 Vehicle-Carried Orbit-Defined/VO

An additional frame was added for the orientation with respect to a Local Vertical Local Horizontal (LVLH) frame. The Vehicle-Carried Orbit-Defined frame (VO) is an LVLH frame defined as defined in [ref.16]:

The following parameters define the VO frame as shown in Figure 7.

- Centered at the CoM of the vehicle
- The Z axis points from the vehicle CoM toward the nadir
- The Y axis is normal to the orbital plane
    - Positive to the right when looking in the direction of the spacecraft velocity
- The X axis is defined X = Y × Z
    - The X axis points in the ***direction*** of the velocity vector but is not necessarily coincident with it.

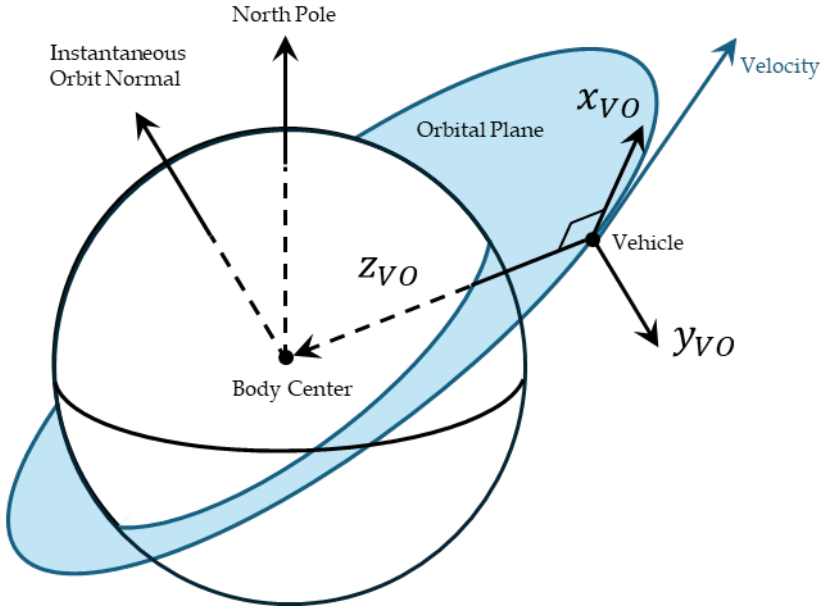

Figure 7. Vehicle-Carried Orbit-Defined Frame

7.3.3.3 Body

This assessment makes use of a singular vehicle body-axis system with axes and origin fixed with respect to a rigid vehicle. The vehicle's mass properties are defined in terms of this system. The body coordinate system is defined as follows:

- Centered at the instantaneous CoM of the vehicle
- The X, Y, and Z axis form a right-handed coordinate system defined by the vehicle's mass properties (see Section 7.3.1).

7.3.3.4 Lunar Polar Stereographic (LPS)

A polar stereographic coordinate system was chosen for this assessment to compare simulation results in calculating in a coordinate system defined by the Artemis mission Program [ref. 17].

The LPS coordinate system is used in this assessment to calculate the vehicle's position over the north and south poles in a standardized polar stereographic projection. Figure 8 shows an example conceptual polar stereographic projection.

The properties of the LPS coordinate system, as defined by the reference above, are:

- Spherical model with lunar radius of 1737.4 km, as in the IAU definition (see Section 7.3.2.1)
- LPS is referenced/aligned to the Mean Earth frame
- False Northing of 500 km
- False Easting of 500 km
- Center latitude for north/south projections respectively $\varphi 0 = \pm 90°$
- Central scale factor $k0 = .994$ at $\varphi 0$
- Outputs are only valid for latitudes $\varphi >= 80°$ or $<= -80°$ for north/south projections respectively
    - Sim products output 0,0 for coordinates when outside of the valid range

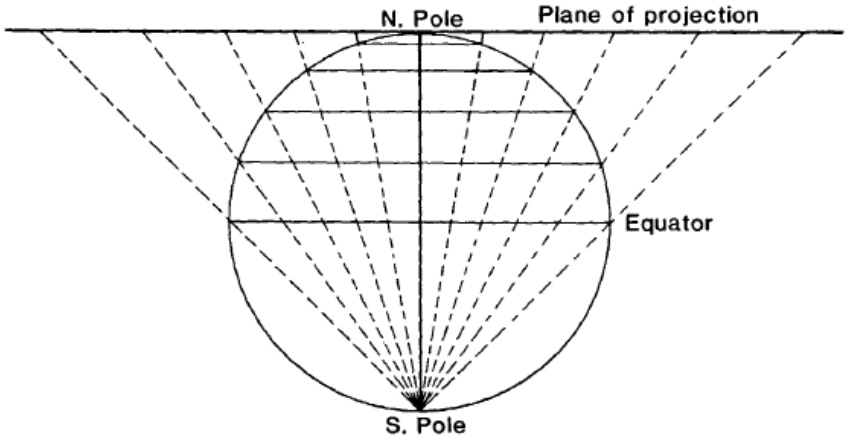

Figure 8. A Lunar Map of Conceptual Polar Stereographic Projections [ref. 14]

7.3.4 Gravitation Models

Two gravitational acceleration models were implemented for the check-cases. Case 1 uses inverse radius square gravitation for the Moon. All other cases use the GRAIL model for modeling lunar gravitational acceleration on the vehicle (see Section 7.3.4.2 for definition).

Third-body contributions are implemented using inverse square gravitation.

7.3.4.1 Inverse Square Gravitation

As in the original assessment, a simple inverse square law gravitation model was used for the primary body/Moon and for third-body contributions where applicable.

Third-body gravitational acceleration terms should not include any aberration nor relativistic corrections. Gravitational constants for each of the bodies are listed in Table E as specified by DE440 [ref. 18].

Table E. Planetary Gravitational Constants

Body	Gravitational Constant (km3/s2)
Moon	4902.8001184575496
Earth	3.9860043550702266E+05
Sun	1.3271244004127942E+11

7.3.4.2 Gravity Recovery and Interior Laboratory (GRAIL)

The GRAIL gravitational model is used for most of the cases. GRAIL is a spherical harmonic gravitational model use for high-fidelity modeling of the gravitational environment near the Moon. The GRAIL model is kept at the Planetary Data Sciences (PDS) Geosciences Node at https://pds-geosciences.wustl.edu/. For most of the check-cases, the low-fidelity GRAIL model is truncated to degree and order 8, while the high-fidelity GRAIL model is truncated to degree and order 320. The spherical harmonic coefficients are available in a variety of formats from:

pds-geosciences.wustl.edu - /grail/grail-l-lgrs-5-rdr-v1/grail_1001/shadr/

GRAIL data files have values for reference radius and gravitational constant. These values are part of the model and should only be used for the GRAIL model itself. For all other non-GRAIL uses, the IAU 2015 radius and gravitational constant values specified for each case are used.

7.3.5 Development Ephemerides

For this assessment, the JPL Planetary and Lunar Ephemerides DE440 was chosen as the standard basis of calculation for all ephemerides. All participating simulation tools in this assessment used Spacecraft, Planet, Instrument, C-Matrix, Events (SPICE) to calculate the ephemerides. The following is a list of the SPICE Kernels for DE440 recommended by the assessment team in the order specified to ensure a common data set for all participating simulations and can be found at the following website https://naif.jpl.nasa.gov/naif/data.html:

moon_pa_de440_200625.bpc
moon_de440_220930.tf
pck00011.tpc
naif0012.tls
de440.bsp

7.3.6 Time Frames

Coordinated Universal Time (UTC) was used as the specification for the starting time for each of the cases and is specified in each of the case descriptions.

A standard number of 37 leap seconds was used for all cases, corresponding to the number of leap seconds at the time the assessment was created.

The standard definition for the offset between Terrestrial Time (TT) and International Atomic Time (TAI) of 32.184 second was specified.

Barycentric Dynamical Time (TDB) is used in the outputs for comparison of the base time frame for calculation of ephemerides.

7.4 Output Specifications

The simulation output products are listed in this section along with any additional clarifications for the outputs. The output variable names follow the AIAA S-119 [ref. 16] naming convention.

Simulation outputs are recorded every 60 seconds of elapsed time (elapsedTime_s).

Simulation data was requested to be recorded with 16 significant digits of precision into a Comma Separated Values (CSV) output format.

The CSV format is broadly used in engineering and maintains all significant digits of precision. However, individual programs may alter data slightly depending on the implementation of a data read-in routine, or when using the copy-paste function. The CSV formatted file produced by the simulation tool is considered the authoritative output file.

One common method to check data integrity is a checksum. A checksum is generated by an algorithm that takes a large data file and returns a small digital "fingerprint", typically a short alphanumeric string. A checksum algorithm virtually guarantees that any two files with even a single difference will return different checksums.

A checksum was not requested as an output for this assessment. Late in the course of the assessment, even very small differences introduced by software programs could be seen when comparing data. Given the ease of computing and sharing checksums, any effort that involves sharing data should consider using checksums.

For outputs of the transformations between frames, quaternions were ultimately chosen as the primary output product, along with optional Euler angle outputs (see Section 7.4.2), and post-processed Euler angles (see Section 7.4.6). Approximately half the participating simulations use left transformative and half use right transformative quaternions [ref. 15]. Texts and the general literature most commonly use right transformative quaternions. Ultimately, the right transformative quaternions were chosen for the simulation output products.

All IAU referenced outputs as well as orbital element outputs use the IAU 2015 spheroid radius specified in section 7.3.2.1.

7.4.1 Common Output Variables

The following outputs are recorded for all scenarios.

elapsedTime_s

Current elapsed time in seconds from the start of the scenario.

j2000UtcTime_s

Current UTC referenced to the J2000 epoch in seconds.

(UTC Julian date - 2451545.0 days, January 1.5, 2000)

j2000TtTime_s

Current Terrestrial Time (TT) referenced to the J2000 epoch in seconds.

(TT Julian date - 2451545.0 days, January 1.5, 2000)

j2000TdbTime_s

Current Barycentric Dynamical Time (TDB) referenced to the J2000 epoch in seconds.

miPosition_m_X, miPosition_m_Y, miPosition_m_Z

Position of the vehicle CoM relative to MI presented in the MI coordinate system in meters.

miVelocity_m_s_X, miVelocity_m_s_Y, miVelocity_m_s_Z

Velocity of the vehicle CoM relative to MI presented in the MI frame in meters per second.

miAccel_m_s2_X, miAccel_m_s2_Y, miAccel_m_s2_Z

Total acceleration of the vehicle CoM relative to MI presented in the MI frame in meters per second squared.

quaternionWrtMi_W, quaternionWrtMi_X, quaternionWrtMi_Y, quaternionWrtMi_Z

Attitude of the vehicle body frame relative to the MI frame. Expressed as a right transformative unit quaternion

W = scalar, X,Y,Z = vector components

bodyAngularRateWrtMi_deg_s_Roll, bodyAngularRateWrtMi_deg_s_Pitch, bodyAngularRateWrtMi_deg_s_Yaw

Angular velocity of the vehicle body frame relative to the MI frame presented in the body frame in degrees per second. Where roll is a rate about the body X axis, pitch is a rate about the body Y axis, and yaw is a rate about the body Z axis.

bodyAngularAccelWrtMi_deg_s2_Roll, bodyAngularAccelWrtMi_deg_s2_Pitch, bodyAngularAccelWrtMi_deg_s2_Yaw

Angular acceleration of the vehicle body frame relative to MI frame presented in the body frame in degrees per second squared. Where roll is an acceleration about the body X axis, pitch is an acceleration about the body Y axis, and yaw is an acceleration about the body Z axis.

memPosition_m_X, memPosition_m_Y, memPosition_m_Z

Position of the vehicle CoM relative to MI presented in the MEM planet-fixed coordinate system in meters.

memLatitude_deg, memLongitude_deg

Latitude and longitude of vehicle CoM in the MEM planet-fixed coordinate system in degrees. Uses IAU 2015 defined spheroid.

Ranges: Longitude [-180,180), Latitude [-90,90]

pamPosition_m_X, pamPosition_m_Y, pamPosition_m_Z

Position of the vehicle CoM relative to MI presented in the PAM planet-fixed coordinate system in meters.

pamLatitude_deg, pamLongitude_deg

Latitude and longitude of vehicle CoM in the PAM planet-fixed coordinate system in degrees. Uses IAU 2015 defined spheroid.

Ranges: Longitude [-180,180), Latitude [-90,90]

lpsPosition_m_X, lpsPosition_m_Y

Position of the vehicle CoM in the LPS coordinate system in meters. This is calculated in either the north pole or south pole stereographic projection depending on the position of the CoM.

X and Y outputs shall be ZERO when outside the LPS region (i.e., between -80 and +80 ME latitude)

pamLocalGravitation_m_s2_X, pamLocalGravitation_m_s2_Y, pamLocalGravitation_m_s2_Z

Gravitational acceleration at the vehicle CoM due to the Moon presented in the PAM frame in meters per second squared. This variable only contains gravitational acceleration contributions from the inverse radius squared/GRAIL models. Third-body gravitation contributions are not included in this output.

quaternionWrtVo_W, quaternionWrtVo_X, quaternionWrtVo_Y, quaternionWrtVo_Z

Attitude of the vehicle body frame relative to the VO frame. Expressed as a right transformative unit quaternion.

W = scalar, X, Y, Z = vector components

altitudeIau_m

Current altitude of the vehicle CoM above reference in meters - IAU 2015 (1737.4 km spheroid). All orbital parameters are referenced to a lunar centered coordinate system.

periapsisIau_m

Orbital parameter for argument of periapsis of the orbit of the vehicle CoM in meters (altitude above reference - IAU 2015 (1737.4 km spheroid)).

apoapsisIau_m

Orbital parameter for argument of apoapsis of the orbit of the vehicle CoM in meters (altitude above reference - IAU 2015 (1737.4 km spheroid)).

eccentricity

Orbital parameter for eccentricity of the orbit of the vehicle CoM.

inclinationMi_deg

Orbital parameter for inclination of the orbit of the vehicle CoM referenced to the MI equatorial plane in degrees.

Range: [0,180] degrees

inclinationPam_deg

Orbital parameter for inclination of the orbit of the vehicle CoM referenced to the PAM equatorial plane in degrees.

Range: [0,180) degrees

semiMajorAxis_m

Orbital parameter for semi-major axis of the orbit of the vehicle CoM in meters.

trueAnomaly_deg

Orbital parameter for true anomaly of the vehicle CoM in degrees.

Range: [0,360) degrees

rightAscensionMi_deg

Orbital parameter for right ascension of the ascending node of the orbit of the vehicle CoM referenced to the MI frame in degrees.

Range: [0,360) degrees

7.4.2 Optional Output Variables

The following outputs were optional for all simulations and are recorded for all scenarios.

eulerAngleWrtMi_deg_Roll, eulerAngleWrtMi_deg_Pitch, eulerAngleWrtMi_deg_Yaw

Attitude of the vehicle body frame relative to the MI frame in degrees. Expressed as an Euler 3-2-1 rotation sequence in the order of Yaw, Pitch, Roll, or body Z,Y,X.

Ranges: Roll [-180,180), Pitch [-90,90], Yaw [0,360) degrees

eulerAngleWrtVo_deg_Roll, eulerAngleWrtVo_deg_Pitch, eulerAngleWrtVo_deg_Yaw

Attitude of the vehicle body frame with respect to the VO frame in degrees. Expressed as an Euler 3-2-1 rotation sequence in the order of Yaw, Pitch, Roll, or body Z,Y,X.

Ranges: Roll [-180,180), Pitch [-90,90], Yaw [0,360) degrees

7.4.3 Third Body Gravitation Output Variables

The following outputs are recorded for all third body gravitation cases as: follows.

- Cases: 5, 5A, 6, 6A, 7, 8, 9, 9A, 9B

miLocalGravitationSun_m_s2_X, miLocalGravitationSun_m_s2_Y, miLocalGravitationSun_m_s2_Z

Gravitational acceleration contribution at the vehicle CoM due to the Sun presented in the MI frame in meters per second squared.

miLocalGravitationEarth_m_s2_X, miLocalGravitationEarth_m_s2_Y, miLocalGravitationEarth_m_s2_Z

Gravitational acceleration contribution at the vehicle CoM due to the Earth presented in the MI frame in meters for second squared.

eulerAngleOfSunWrtBody_deg_Pitch, eulerAngleOfSunWrtBody_deg_Yaw

Pointing angle from the body frame to point the body X axis at the Sun in degrees. Expressed as an Euler 3-2 rotation sequence in the order of Yaw, Pitch, or body Z,Y. No roll rotation.

Ranges: Pitch [-90,90], Yaw [0,360) degrees

7.4.4 Case 9 Specific Output Variables – Test Points

The following outputs are recorded only for case 9 for the purposes of checking specific equations and for the evaluation of Digital Elevation Model (DEM) data through the simulation tools.

- Cases: 9 only

pamLocalGravitationOfTp1_m_s2_X, pamLocalGravitationOfTp1_m_s2_Y, pamLocalGravitationOfTp1_m_s2_Z

Gravitational acceleration at Test Point 1 due to the Moon presented in the PAM frame in meters per second squared. This variable only contains gravitational acceleration contributions from the inverse radius squared/GRAIL models. Third-body gravitation contributions are not included in this output.

pamLatitudeOfTp1_deg, pamLongitudeOfTp1_deg

Latitude and longitude of Test Point 1 in the PAM planet-fixed coordinate system in degrees. Uses IAU 2015 defined spheroid.

Ranges: Longitude [-180,180), Latitude [-90,90]

altitudeIauOfTp2_m

Current altitude of Test Point 2 above reference in meters - IAU 2015 spheroid. This is the altitude of the DEM test point above the reference spheroid and is derived from DEM elevation data. See sections 7.3.2.4 and 7.7.15.

pamLatitudeOfTp2_deg, pamLongitudeOfTp2_deg

Latitude and longitude of Test Point 2 in the PAM planet-fixed coordinate system in degrees. Uses IAU 2015 defined spheroid.

Ranges: Longitude [-180,180), Latitude [-90,90]

7.4.5 Case 9, 9A, 9B Specific Output Variables — Sensor

The following outputs are only recorded for cases 9, 9A, and 9B for the purposes of checking specific equations related to a sensor located on a vehicle.

The variables were named ‘SensedPosition’ as a reference to how the position, velocity, and acceleration would be ‘sensed’ at a particular sensor location. Since no errors are modeled, this is identical to the true position, velocity, and acceleration values (without gravitational acceleration).

- Cases: 9, 9A, and 9B only

miSensedPositionOfSensor_m_X, miSensedPositionOfSensor_m_Y, miSensedPositionOfSensor_m_Z

Sensed position of the sensor relative to MI presented in the MI frame in meters.

miSensedVelocityOfSensor_m_s_X, miSensedVelocityOfSensor_m_s_Y, miSensedVelocityOfSensor_m_s_Z

Sensed velocity of the sensor relative to MI presented in the MI frame in meters per second.

miSensedAccelOfSensor_m_s2_X, miSensedAccelOfSensor_m_s2_Y, miSensedAccelOfSensor_m_s2_Z

Sensed acceleration of the sensor relative to MI presented in the MI frame in meters per second squared. This 'sensed acceleration' does not include gravitational acceleration contributions.

7.4.6 Post-Processed Outputs

The following variables were computed in post processing. All post-processed variables are prefixed with pp_ to indicate they were not direct products of the participating simulations.

pp_eulerAngleWrtMi_deg_Roll, pp_eulerAngleWrtMi_deg_Pitch, pp_eulerAngleWrtMi_deg_Yaw

Attitude of the vehicle body frame relative to the MI frame in degrees. Expressed as an Euler 3-2-1 rotation sequence in the order of Yaw, Pitch, Roll, or body Z,Y,X.

Ranges: Roll [-180,180), Pitch [-90,90], Yaw [0,360) degrees

Purpose: to provide a more human readable version or orientation to compare vs. looking at quaternions.

Calculation: Computed from quaternionWrtMi_W/X/Y/Z

pp_eulerAngleWrtVo_deg_Roll, pp_eulerAngleWrtVo_deg_Pitch, pp_eulerAngleWrtVo_deg_Yaw

Attitude of the vehicle body frame relative to the VO frame in degrees. Expressed as an Euler 3-2-1 rotation sequence in the order of Yaw, Pitch, Roll, or body Z,Y,X.

Ranges: Roll [-180,180), Pitch [-90,90], Yaw [0,360) degrees

Purpose: to provide a more human readable version or orientation to compare vs. looking at quaternions.

Calculation: Computed from quaternionWrtVo_W/X/Y/Z

pp_miPositionOfSensorWrtCm_m_X, pp_miPositionOfSensorWrtCm_m_Y, pp_miPositionOfSensorWrtCm_m_Z

Position of the sensor relative to the vehicle CoM presented in the MI frame in meters.

Note: This is the true sensor position as represented in the EOM relative to the CoM. This is intended to be used for EOM results comparison and not to emulate any actual sensor state or output.

Calculation: miSensedPositionOfSensor_m_X/Y/Z - miPosition_m_X/Y/Z. (Note: this calculation will result in some precision loss.)

Purpose: To look for calculation discrepancies in how the miSensedPositionOfSensor outputs are calculated (e.g., drift in body relative sensor position).

7.5 Simulation Output Comparison Website

A static website, hosted through the JSC GitLab Pages feature, was developed as a tool for the simulation groups to perform quick data comparison using interactive plots, access scenario specifications, and catalogue the results. The website was continuously developed throughout the

assessment and was accessible to anyone with NASA Virtual Private Network. A full report, formatted in LaTeX, was automatically generated using Gitlab Continuous Integration and made available on the website. Each scenario had its own webpage that included a scenario description, specification table, a list of the latest results, and interactive plots.

The interactive plots were developed using the plotly.js JavaScript plotting library and were a valuable resource that avoided the need for sim groups to develop their own plotting routines for comparison to other submitted results. Through the web interface the user had the option between time histories and difference plots with respect to an average of all sim group results or an individual sim group. By default, all sim results were displayed on the plots, but the user had the option to select sims to omit. Interactive plot options included zooming to regions of the plot, hovering over the plot with the cursor to display the value at any point along each series, and hiding or showing specific plot traces. Most scenario webpages had three plots, and each could draw any of the defined output variables associated with the scenario as selected by the user. A feature was added to enable anyone with access to the website to load temporary data to compare to the existing delivered results. This allowed sim groups to compare newly generated results to the other sim groups without the need to provide a more formal delivery. It also gave the team the opportunity to compare results between difference scenarios. For example, data from Scenario 8 could be uploaded to any of the NRHO sensitivity subcases to evaluate the sensitivity inherit in a full NRHO period. A feature was added to the interactive plotting website to optionally highlight a 'family' of plot traces detected by a K-means clustering algorithm. The K-means algorithm iterates to identify clusters of similar data based on relative distance and variance of the data provided. Clusters of traces are identified using the final points in each plot trace and a primary 'family' is determined based on the largest cluster group. Once a 'family' is found, a shaded region is drawn based on the min and max values of the traces in the group. Quaternion outputs were modified in post processing by forcing the scalar element of the quaternion to always be positive, inverting the signs on the vector component as necessary. Consistent signs on the plotted quaternion elements across the sim traces made them easier to compare.

7.6 Simulation Results Interpretation

The primary output of the check-cases is a time history of each output variable, which can then be plotted with any data plotting software. For simulation comparison, the results from multiple simulations are plotted together. Most of the output data for different simulations looks to be nearly identical when plotted at the full scale of the data. A second type of plot, showing the difference between each simulation's outputs and the average value of that output, is typically more useful for comparing and contrasting data.

The simulations are presented anonymously, simply labeled SIM1 through SIM8. This ordering is unrelated to and independent of the order of the simulation descriptions in Section 7.1. Case 1 also includes the analytical solution, labeled "ref1" as described in Appendix B: Keplerian Propagation.

The appendix to this report, Appendix A, includes a wealth of output plots from the participating simulations. Both the full-scale outputs and difference plots are shown. The difference plots are scaled to easily show the differences on the plot regardless of the scale. The interactive website described in Section 7.5 provides additional options to compare simulation data.

The simulation outputs were compared for all plots. It was difficult to determine a matching criterion due to the intentional different purposes each simulation was developed to meet and their underlying differences in implementation of algorithms and numerical integration. Attempts were made to specify clear and unambiguous ground rules, assumptions, definitions, frames, sources of data and other relevant information. However, it is possible that some definitions remained unclear or misinterpreted as several plots reveal significant differences. These differences highlight the importance of continuous communication and documenting all aspects of the modeling and simulation process. NASA STD 7009 states that uncertainty quantification is: *The process of identifying all relevant sources of uncertainties, characterizing them in all models, experiments, and comparisons of M&S results and experiments, and of quantifying uncertainties in all relevant inputs and outputs of the simulation or experiment.* It was beyond the scope and resources of this assessment to characterize the sources of all uncertainties and comparisons to show favorable comparisons. However, NASA STD 7009 Section 4.1.3 indicates: Programs must determine: *the critical decisions to be addressed with M&S and to determine which M&S are in scope.* Part of the acceptance criteria to use a Model or Simulation includes specification *of what constitutes a favorable comparison for the Verification Evidence, Validation Evidence, Input Pedigree Evidence and Use History level definitions in the Credibility Assessment Scale*. While this assessment declined to assert a criterion for what is 'in family', the comparisons plots were mostly favorable across the simulations for many of the outputs. A feature was added to the static website describing simulations that were 'within family' as previously described. While no one simulation was described as 'truth', this feature is helpful to new simulation developers as a check if implementation of the check-cases matched other known sims. It is left as an exercise for the Simulation and Model provider and Program Office to determine the 'criteria' for favorable comparison to achieve the Verification and Validation goals specified.

The result of this effort is intended to provide external simulation developers a set of reasonable check-cases to support simulation validation and identify common areas where development 'bugs' can occur. Overall, simulation teams improved their tools, added new capabilities, found errors in implementation and gained an improved understanding of the details of their simulation from participating in this activity. The external site for simulation results and output upload will be maintained under the Flight Mechanics NESC Academy repository as a useful resource for simulation developers.

The simulation teams strove to match configurations and initial conditions as much as possible. The plots show remaining differences after these efforts. The plots are largely presented as-is, without speculation about the source of any particular differences. Four important sources of potential differences are described next.

During the check-cases, comparison of the VO frame (see Section 7.3.3.2) data showed up often as a source of differences. Differences were typically masked by the circular or nearly circular cases but became apparently in the VO frame output data for the elliptical orbits. This occurred due to two 'standard' definitions of VO or LVLH frames, one as used in this assessment where the X axis is in the direction of the velocity vector, and another where the X axis is coincident with the velocity vector. By the end of the assessment, it appeared the differences in this frame definition were resolved.

When originally comparing NRHO third-body gravitational acceleration differences, a difference in simulation was tracked down to at least one simulation (sim 8) including an 'indirect

oblateness acceleration' term for the Earth gravitational acceleration contribution. This difference thus applies to all scenarios with third-body gravitational acceleration terms. No corrective action was taken.

When comparing simulation time data, the sim teams discovered that at least two of the simulations (sims 7 and 8), advanced simulation time in the TDB time frame, and thus including relativistic effects, whereas most of the simulations advanced time in the UTC/TT time frame. No corrective action was taken.

The last important difference noted was for the DEM outputs from Case 9 (see Section 7.4.4 and 7.7.15), specifically altitudeIauOfTp2_m. The DEM output data was added late in the assessment and little time was available to the simulation teams to analyze and resolve differences, especially considering that some of the simulation tools had to implement new routines to handle the new output data. When originally added, it was noted that differences would likely be observed due to differences in interpolation routines. The time zero location of the test point was chosen explicitly to be a highly sloped region of the south pole, and examining the plots, it is observed that indeed this is a location with some of the largest differences between the simulations.

7.7 Check-Case Descriptions for Implementation

The complete descriptions of each check-case are given below, along with additional descriptions and discussions of each check-case. The tables contain all the information as it was presented during the study. Certain scenario descriptions, e.g., the inclusion of gravity gradient effects or the output format, are the same for every scenario but are included to preserve the original check-case description.

Throughout the assessment, it was observed that some additional clarifying information regarding the vehicle body frame orientation and other aspects was useful for implementing the check-cases. The tables contain this additional information to aid in implementing the cases.

For convenience, many of the orbits are described as 'circular orbits'. As with any real orbit, especially in the presence of non-Keplerian gravity and third-body perturbations, the orbits could more properly be called 'near-circular orbits'. This detail can detract from the discussion, so they will be called circular orbits.

7.7.1 Check-case 1 – Keplerian Propagation

Scenario 1 tests simple Keplerian propagation in an inverse square gravity field. It uses the cylinder with no initial rotational velocity with respect to the VO frame. The body axes initially align with the VO coordinate system. The angular rotation rate in the body frame is matched to the orbital period so that the vehicle maintains an approximately constant orientation relative in the VO frame. Over the course of the simulation there is a small amount of drift, as the initial conditions for the angular rate are given to the nearest 1e-5 degrees/second.

Scenario 1 serves as an introductory case with minimal implementation details beyond orbital motion. The simpler cylinder model is used, and it is the only case which uses inverse square gravity for the Moon gravitation. This case is useful for testing orientation and rotation conventions, as for all subsequent cases the gravity model will have effects on the vehicle's rotation.

Finally, the simple Keplerian gravity model has an analytic solution, and propagated solutions can be compared to the ideal Keplerian solution. A Python script was developed to allow easy output of orbital states and parameters. The outputs of the script are available [ref. Appendix B] along with the outputs from the rest of the simulation tools. In addition to comparing outputs from scenario 1, note that scenarios 2 and 3 use the same initial conditions with different gravitational models. The analytical outputs can be compared with scenarios 2 and 3 to see the effects of the higher-order gravity models.

Constants and Models			
The gravitational constant, as defined in DE440, is GM = 4902.8001184575496km3/s2. The ephemerides are: DE440. Time is expressed in UTC. Although test case initial conditions may be in the future, the following constants are used: TAI − UTC = 37 sec; DUT1 = 0.0 sec			
Initial Conditions			
Scenario	1. Keplerian Propagation Ref 1 ICs		
Vehicle	Cylinder		
Orientation	Body frame aligned with VO frame quaternionWrtVo_W/X/Y/Z should be [1,0,0,0] respectively		
Rotation	bodyAngularRateWrtMi: [0, -0.05008, 0] deg/s (angular rotation rate of the body frame relative to MI presented in the body frame)		
Gravitation	Inverse square	**Gravity Gradient Effects**	Off
Sun/Earth Perturbations	Off	**Duration**	28,800 s
Output Interval	60 s	**Output Format**	CSV

Initial States		
Time	2025-7-15 11:23:27.30 UTC corresponds to output j2000UtcTime_s J2000 epoch of January 1.5, 2000 = -2451545.0 days Using 37 leap seconds j2000TtTime_s = j2000UtcTime + 32.184 + 37 = j2000UtcTime + 69.184 (s) Note that delta UT1/UTC is not relevant for check-case and is thus not specified	
Frame	Planet-Centered Inertial	
Position (m)	miPosition_m_X miPosition_m_Y miPosition_m_Z	-194338.26150101773 824894.7002999065 1653703.391999927

Velocity (m/s)	miVelocity_m_s_X miVelocity_m_s_Y miVelocity_m_s_Z	80.94043359034313 -1447.8938749999684 731.723312100025

7.7.2 Check-case 2 — Low-Fidelity 8x8 GRAIL

Scenario 2 is identical to Scenario 1 except for the lunar gravity model. This case uses the GRAIL model to degree and order 8 instead of inverse square gravity. This case tests the basic implementation of the GRAIL model, using enough terms to clearly see the effect of realistic lunar gravity without being too computationally complex. The next case (check-case 3) will implement a higher-order model, otherwise the 8x8 GRAIL model is used for all subsequent check-cases.

In practice, the lunar gravity field is complex, and a higher degree and order should be used for most simulation work. The exact degree and order depend on the length of the simulation and the effects under study. Check-case 2 can be compared with check-case 3 to see how much difference the gravity model makes for even a short, 8-hour propagation.

Constants and Models			
The gravitational constant is defined in the GRAIL model specified in Section 7.3.4.2. The ephemerides are: DE440. Time is expressed in UTC. Although test case initial conditions may be in the future, the following constants are used: TAI − UTC = 37 sec; DUT1 = 0.0 sec Low-Fidelity GRAIL uses the GRAIL gravity model of degree and order 8 (i.e., 8x8 GRAIL).			
Initial Conditions			
Scenario	2. Low-Fidelity 8x8 GRAIL Ref 1 ICs		
Vehicle	Cylinder		
Orientation	Body frame aligned with VO frame quaternionWrtVo_W/X/Y/Z should be [1,0,0,0] respectively		
Rotation	bodyAngularRateWrtMi: [0, -0.05008, 0] deg/s (angular rotation rate of the body frame relative to MI presented in the body frame)		
Gravitation	Low-Fidelity GRAIL	**Gravity Gradient Effects**	Off
Sun/Earth Perturbations	Off	**Duration**	28,800 s
Output Interval	60 s	**Output Format**	CSV
Initial States			
Time	2025-7-15 11:23:27.30 UTC corresponds to output j2000UtcTime_s J2000 epoch of January 1.5, 2000 = -2451545.0 days Using 37 leap seconds j2000TtTime_s = j2000UtcTime + 32.184 + 37 = j2000UtcTime + 69.184 (s) Note that delta UT1/UTC is not relevant for check-case and is thus not specified		
Frame	Planet-Centered Inertial		

Position (m)	miPosition_m_X miPosition_m_Y miPosition_m_Z	-194338.26150101773 824894.7002999065 1653703.391999927
Velocity (m/s)	miVelocity_m_s_X miVelocity_m_s_Y miVelocity_m_s_Z	80.94043359034313 -1447.8938749999684 731.723312100025

7.7.3 Check-case 3 – High-Fidelity 320x320 GRAIL

Scenario 3 is identical to Scenarios 1 and 2 except for the lunar gravity model. This case uses the GRAIL gravity model to degree and order 320. This case tests the implementation of a high-fidelity gravity model. This case can be compared to check-case 2 to see the effects of the Moon's complex gravitational field and provides simulation developers an opportunity to check the implementation of a computationally intense function requiring a large amount of input data across a range of significant figures.

Constants and Models			
The gravitational constant is defined in the GRAIL model specified in Section 7.3.4.2. The ephemerides are: DE440. Time is expressed in UTC. Although test case initial conditions may be in the future, the following constants are used: TAI − UTC = 37 sec; DUT1 = 0.0 sec High-Fidelity GRAIL uses the GRAIL gravity model of degree and order 320 (i.e., 320x320 GRAIL).			
Initial Conditions			
Scenario	3: High-Fidelity 320x320 GRAIL Ref 1 ICs		
Vehicle	Cylinder		
Orientation	Body frame aligned with VO frame quaternionWrtVo_W/X/Y/Z should be [1,0,0,0] respectively		
Rotation	bodyAngularRateWrtMi: [0, -0.05008, 0] deg/s (angular rotation rate of the body frame relative to MI presented in the body frame)		
Gravitation	High-Fidelity GRAIL	**Gravity Gradient Effects**	Off
Sun/Earth Perturbations	Off	**Duration**	28,800 s
Output Interval	60 s	**Output Format**	CSV
Initial States			
Time	2025-7-15 11:23:27.30 UTC corresponds to output j2000UtcTime_s J2000 epoch of January 1.5, 2000 = -2451545.0 days Using 37 leap seconds j2000TtTime_s = j2000UtcTime + 32.184 + 37 = j2000UtcTime + 69.184 (s) Note that delta UT1/UTC is not relevant for check-case and is thus not specified		
Frame	Planet-Centered Inertial		

Position (m)	miPosition_m_X miPosition_m_Y miPosition_m_Z	-194338.26150101773 824894.7002999065 1653703.391999927
Velocity (m/s)	miVelocity_m_s_X miVelocity_m_s_Y miVelocity_m_s_Z	80.94043359034313 -1447.8938749999684 731.723312100025

7.7.4 Check-case 4 – High Circular Orbit

Scenario 4 introduces two new elements, a high-altitude circular orbit and the Apollo vehicle model. The vehicle model is based directly on the Apollo vehicle, using contemporary documentation. The Apollo vehicle is asymmetric, which tests implementation of some mass properties and conventions for vehicle orientation. The altitude for the circular orbit was chosen to be where third-body perturbation effects start to become important, and these effects will be introduced in another check-case (Check-case 4A).

Scenario 4, as with all subsequent scenarios, will use the 8x8 GRAIL gravity model.

Constants and Models			
The gravitational constant is defined in the GRAIL model specified in Section 7.3.4.2. The ephemerides are: DE440. Time is expressed in UTC. Although test case initial conditions may be in the future, the following constants are used: TAI − UTC = 37 sec; DUT1 = 0.0 sec Low-Fidelity GRAIL uses the GRAIL gravity model of degree and order 8 (i.e., 8x8 GRAIL).			
Initial Conditions			
Scenario	4. High Circular Orbit		
Vehicle	Apollo Vehicle Model		
Orientation	Body frame aligned with VO frame quaternionWrtVo_W/X/Y/Z should be [1,0,0,0] respectively		
Rotation	bodyAngularRateWrtMi: [0, -0.03788, 0] deg/s (angular rotation rate of the body frame relative to MI presented in the body frame)		
Gravitation	Low-Fidelity GRAIL	**Gravity Gradient Effects**	Off
Sun/Earth Perturbations	Off	**Duration**	28,800 s
Output Interval	60 s	**Output Format**	CSV
Initial States			
Time	2026-1-28 6:42:03.51 UTC corresponds to output j2000UtcTime_s J2000 epoch of January 1.5, 2000 = -2451545.0 days Using 37 leap seconds j2000TtTime_s = j2000UtcTime + 32.184 + 37 = j2000UtcTime + 69.184 (s) Note that delta UT1/UTC is not relevant for check-case and is thus not specified		
Frame	Planet-Centered Inertial		

Position (m)	miPosition_m_X miPosition_m_Y miPosition_m_Z	-842357.1179795 -2062423.44196825 206888.355905013
Velocity (m/s)	miVelocity_m_s_X miVelocity_m_s_Y miVelocity_m_s_Z	1255.71312544442 -448.293487863427 643.766975306731

7.7.5 Check-case 5 – High Circular Orbit with Perturbations

Scenario 5 is identical to Scenario 4 with the exception of added third-body perturbations from the Earth and the Sun. The Earth and Sun are modeled as point masses, and are the only third bodies included. In practice, additional third bodies are typically included for propagation. For this assessment, the effects of the Earth and the Sun dominate and are more than sufficient to test the implementation of third-body perturbation effects.

Constants and Models			
The gravitational constant is defined in the GRAIL model specified in Section 7.3.4.2. The ephemerides are: DE440. Time is expressed in UTC. Although test case initial conditions may be in the future, the following constants are used: TAI − UTC = 37 sec; DUT1 = 0.0 sec Low-Fidelity GRAIL uses the GRAIL gravity model of degree and order 8 (i.e., 8x8 GRAIL).			
Initial Conditions			
Scenario	Lunar Case 5: High Circular Orbit with Perturbations		
Vehicle	Apollo Vehicle Model		
Orientation	Body frame aligned with VO frame quaternionWrtVo_W/X/Y/Z should be [1,0,0,0] respectively		
Rotation	bodyAngularRateWrtMi: [0, -0.03788, 0] deg/s (angular rotation rate of the body frame relative to MI presented in the body frame)		
Gravitation	Low-Fidelity GRAIL	**Gravity Gradient Effects**	Off
Sun/Earth Perturbations	On	**Duration**	28,800 s
Output Interval	60 s	**Output Format**	CSV
Initial States			
Time	2026-1-28 6:42:03.51 UTC corresponds to output j2000UtcTime_s J2000 epoch of January 1.5, 2000 = -2451545.0 days Using 37 leap seconds j2000TtTime_s = j2000UtcTime + 32.184 + 37 = j2000UtcTime + 69.184 (s) Note that delta UT1/UTC is not relevant for check-case and is thus not specified		
Frame	Planet-Centered Inertial		
Position (m)	miPosition_m_X miPosition_m_Y miPosition_m_Z	-842357.1179795 -2062423.44196825 206888.355905013	

Velocity (m/s)	miVelocity_m_s_X miVelocity_m_s_Y miVelocity_m_s_Z	1255.71312544442 -448.293487863427 643.766975306731

7.7.6 Check-case 5A – High Circular Orbit with Perturbations and Tumbling

Scenario 5A is identical to Scenario 5 with the exception of the initial vehicle angular rate. This case includes initial rates about all three body axes, testing the implementation of the asymmetric Apollo vehicle. The pitch rate, matched to the orbital rate, is maintained. In addition, initial roll and yaw rates are given.

Constants and Models			
The gravitational constant is defined in the GRAIL model specified in Section 7.3.4.2. The ephemerides are: DE440. Time is expressed in UTC. Although test case initial conditions may be in the future, the following constants are used: TAI − UTC = 37 sec; DUT1 = 0.0 sec Low-Fidelity GRAIL uses the GRAIL gravity model of degree and order 8 (i.e., 8x8 GRAIL).			
Initial Conditions			
Scenario	Lunar Case 5A: High Circular Orbit with Perturbations and Tumbling		
Vehicle	Apollo Vehicle Model		
Orientation	Body frame aligned with VO frame quaternionWrtVo_W/X/Y/Z should be [1,0,0,0] respectively		
Rotation	bodyAngularRateWrtMi: [0.1, -0.03788, 0.001] deg/s (angular rotation rate of the body frame relative to MI presented in the body frame)		
Gravitation	Low-Fidelity GRAIL	**Gravity Gradient Effects**	Off
Sun/Earth Perturbations	On	**Duration**	28,800 s
Output Interval	60 s	**Output Format**	CSV
Initial States			
Time	2026-1-28 6:42:03.51 UTC corresponds to output j2000UtcTime_s J2000 epoch of January 1.5, 2000 = -2451545.0 days Using 37 leap seconds j2000TtTime_s = j2000UtcTime + 32.184 + 37 = j2000UtcTime + 69.184 (s) Note that delta UT1/UTC is not relevant for check-case and is thus not specified		
Frame	Planet-Centered Inertial		
Position (m)	miPosition_m_X miPosition_m_Y miPosition_m_Z	-842357.1179795 -2062423.44196825 206888.355905013	
Velocity (m/s)	miVelocity_m_s_X miVelocity_m_s_Y miVelocity_m_s_Z	1255.71312544442 -448.293487863427 643.766975306731	

7.7.7 Check-case 6 – Highly Elliptical Orbit

Scenario 6 introduces a new orbit and returns to the simpler cylinder model. The new orbit is highly elliptical, with an eccentricity just below 0.7. Near perilune, the vehicle is at its highest orbital velocity, and the detailed lunar gravity model is at its highest significance. In contrast, near apolune the vehicle is at its lowest orbital velocity, and the third-body perturbations are at their highest significance. The highly elliptical orbit has a much longer period than the circular orbits, so the propagation time is increased to 28 hours to ensure that more than two full orbits are simulated.

As with other cases, the initial pitch rate is matched to the orbital period. However, for an elliptical orbit, the vehicle's orientation in the VO frame will change, with significant nodding forward and back over a given orbital period. For the highly elliptical orbit, the pitch varies by more than +/- 80 degrees compared to the initial value.

Constants and Models			
The gravitational constant is defined in the GRAIL model specified in Section 7.3.4.2. The ephemerides are: DE440. Time is expressed in UTC. Although test case initial conditions may be in the future, the following constants are used: TAI − UTC = 37 sec; DUT1 = 0.0 sec Low-Fidelity GRAIL uses the GRAIL gravity model of degree and order 8 (i.e., 8x8 GRAIL).			
Initial Conditions			
Scenario	Lunar Case 6: Highly Elliptical Orbit		
Vehicle	Cylinder		
Orientation	Body frame aligned with VO frame quaternionWrtVo_W/X/Y/Z should be [1,0,0,0] respectively		
Rotation	bodyAngularRateWrtMi: [0, -0.00756, 0] deg/s (angular rotation rate of the body frame relative to MI presented in the body frame)		
Gravitation	Low-Fidelity GRAIL	**Gravity Gradient Effects**	Off
Sun/Earth Perturbations	On	**Duration**	100,800 s
Output Interval	60 s	**Output Format**	CSV
Initial States			
Time	2026-1-28 6:42:03.51 UTC corresponds to output j2000UtcTime_s J2000 epoch of January 1.5, 2000 = -2451545.0 days Using 37 leap seconds j2000TtTime_s = j2000UtcTime + 32.184 + 37 = j2000UtcTime + 69.184 (s) Note that delta UT1/UTC is not relevant for check-case and is thus not specified		
Frame	Planet-Centered Inertial		
Position (m)	miPosition_m_X miPosition_m_Y miPosition_m_Z	1830456.52713665 -711552.31509966 -304763.785806531	

Velocity (m/s)	miVelocity_m_s_X miVelocity_m_s_Y miVelocity_m_s_Z	457.748608640399 335.646153516028 1965.6506408711

7.7.8 Check-case 6A – Highly Elliptical Orbit, Zero Inertial Rotation

Scenario 6A is identical to Scenario 6, except for the initial vehicle angular rate. This case includes an initial vehicle angular rate of identically zero relative to the inertial frame. Most simulations propagate in an inertial frame, and in principle this case represents integrating zero. In practice, a high-fidelity simulation will not necessarily integrate the rotational states in isolation, and different simulations are expected to exhibit small differences.

Constants and Models			
The gravitational constant is defined in the GRAIL model specified in Section 7.3.4.2. The ephemerides are: DE440. Time is expressed in UTC. Although test case initial conditions may be in the future, the following constants are used: TAI − UTC = 37 sec; DUT1 = 0.0 sec Low-Fidelity GRAIL uses the GRAIL gravity model of degree and order 8 (i.e., 8x8 GRAIL).			
Initial Conditions			
Scenario	Lunar Case 6A: Highly Elliptical Orbit – Cylinder, zero inertial rotation		
Vehicle	Cylinder		
Orientation	Body frame aligned with VO frame quaternionWrtVo_W/X/Y/Z should be [1,0,0,0] respectively		
Rotation	bodyAngularRateWrtMi: [0, 0, 0] deg/s (angular rotation rate of the body frame relative to MI presented in the body frame)		
Gravitation	Low-Fidelity GRAIL	**Gravity Gradient Effects**	Off
Sun/Earth Perturbations	On	**Duration**	100,800 s
Output Interval	60 s	**Output Format**	CSV
Initial States			

Time	2026-1-28 6:42:03.51 UTC corresponds to output j2000UtcTime_s J2000 epoch of January 1.5, 2000 = -2451545.0 days Using 37 leap seconds j2000TtTime_s = j2000UtcTime + 32.184 + 37 = j2000UtcTime + 69.184 (s) Note that delta UT1/UTC is not relevant for check-case and is thus not specified
Frame	Planet-Centered Inertial

Position (m)	miPosition_m_X miPosition_m_Y miPosition_m_Z	1830456.52713665 -711552.31509966 -304763.785806531
Velocity (m/s)	miVelocity_m_s_X miVelocity_m_s_Y miVelocity_m_s_Z	457.748608640399 335.646153516028 1965.6506408711

7.7.9 Check-case 7 — Highly Elliptical Orbit, Apollo

Scenario 7 is identical to Scenario 6, with the exception of using the Apollo vehicle model. The back-and-forth nodding effect noted in Scenario 6 is still seen. In addition, the roll and yaw show more variation due to the asymmetric nature of the Apollo vehicle.

Constants and Models			
The gravitational constant is defined in the GRAIL model specified in Section 7.3.4.2. The ephemerides are: DE440. Time is expressed in UTC. Although test case initial conditions may be in the future, the following constants are used: TAI − UTC = 37 sec; DUT1 = 0.0 sec Low-Fidelity GRAIL uses the GRAIL gravity model of degree and order 8 (i.e., 8x8 GRAIL).			
Initial Conditions			
Scenario	Lunar Case 7: Highly Elliptical Orbit — Apollo		
Vehicle	Apollo Vehicle Model		
Orientation	Body frame aligned with VO frame quaternionWrtVo_W/X/Y/Z should be [1,0,0,0] respectively		
Rotation	bodyAngularRateWrtMi: [0, -0.00756, 0] deg/s (angular rotation rate of the body frame relative to MI presented in the body frame)		
Gravitation	Low-Fidelity GRAIL	**Gravity Gradient Effects**	Off
Sun/Earth Perturbations	On	**Duration**	100,800 s
Output Interval	60 s	**Output Format**	CSV
Initial States			

Time	2026-1-28 6:42:03.51 UTC corresponds to output j2000UtcTime_s J2000 epoch of January 1.5, 2000 = -2451545.0 days Using 37 leap seconds j2000TtTime_s = j2000UtcTime + 32.184 + 37 = j2000UtcTime + 69.184 (s) Note that delta UT1/UTC is not relevant for check-case and is thus not specified	
Frame	Planet-Centered Inertial	
Position (m)	miPosition_m_X miPosition_m_Y miPosition_m_Z	1830456.52713665 -711552.31509966 -304763.785806531

Velocity (m/s)	miVelocity_m_s_X miVelocity_m_s_Y miVelocity_m_s_Z	457.748608640399 335.646153516028 1965.6506408711

7.7.10 Check-case 8 — NRHO, Apollo

Scenario 8 introduces the NRHO, which is of great scientific interest, especially for the Artemis program. The NRHO chosen for this study is derived from a reference NRHO orbit from a 2019 white paper for the Artemis program [ref. 11]. While a wealth of information on NRHOs is available in the literature for the interested reader, only the salient points for this assessment are discussed here.

The NRHO is a non-Keplerian orbit with a period of approximately 6.5 days. The period of simulation is significantly longer for this case than for the other check-cases, lasting seven days rather than the eight hours for the circular orbits, or the 28 hours for the elliptical orbit. Due to the long orbital period and non-Keplerian geometry, the vehicle's rotation rate is simply chosen to be some small rate, without a specific effort to match the rate to the period. Because the NRHO is a three-body orbit and not a Keplerian orbit, correct implementation of third-body perturbations is essential. Case 8 (and all its variants) is initially oriented similarly to the other cases, but the quaternion representing the vehicle orientation relative to the MI frame is given explicitly. This ensures that some check-cases start with an identical rotational state in the base inertial EOM states for better comparison.

This assessment uses a common set of outputs for all cases, including Keplerian orbital elements. In practice, Keplerian orbital elements are not suitable for characterizing or describing an NRHO. Orbital elements — eccentricity and semi-major axis, for example — are constant for an ideal Keplerian orbit, and nearly constant for real two-body orbits. These values are decidedly not constant for the NRHO. This assessment is focused on comparing implementations of 6DOF simulations, including automated calculation of derived parameters and coordinate transformations. Every additional simulation output provides additional opportunities for comparison and testing simulation implementation. Therefore, Keplerian orbital elements are output for the NRHO *only* to aid in simulation comparison.

Constants and Models	
The gravitational constant is defined in the GRAIL model specified in Section 7.3.4.2. The ephemerides are: DE440. Time is expressed in UTC. Although test case initial conditions may be in the future, the following constants are used: TAI − UTC = 37 sec; DUT1 = 0.0 sec Low-Fidelity GRAIL uses the GRAIL gravity model of degree and order 8 (i.e., 8x8 GRAIL).	
Initial Conditions	
Scenario	Lunar Case 8: NRHO — Apollo
Vehicle	Apollo Vehicle Model
Orientation	Major axis is approximately along the velocity vector quaternionWrtMi = [0.646080387557766 0.334383101748747 0.685546322693561 0.0281835682522564] Where the components are ordered [W X Y Z]'

Rotation	bodyAngularRateWrtMi: [0.001, 0, 0] deg/s (angular rotation rate of the body frame relative to MI presented in the body frame)		
Gravitation	Low-Fidelity GRAIL	**Gravity Gradient Effects**	Off
Sun/Earth Perturbations	On	**Duration**	604,800 s
Output Interval	60 s	**Output Format**	CSV
Initial States			
Time	2026-1-28 6:42:03.51 UTC corresponds to output j2000UtcTime_s J2000 epoch of January 1.5, 2000 = -2451545.0 days Using 37 leap seconds j2000TtTime_s = j2000UtcTime + 32.184 + 37 = j2000UtcTime + 69.184 (s) Note that delta UT1/UTC is not relevant for check-case and is thus not specified		
Frame	Planet-Centered Inertial		
Position (m)	miPosition_m_X miPosition_m_Y miPosition_m_Z	-5838140.15098070377 2538924.86620857380 1055566.19690058869	
Velocity (m/s)	miVelocity_m_s_X miVelocity_m_s_Y miVelocity_m_s_Z	-685.576059820879546 751.297639180154464 -621.913991301902946	

7.7.11 Check-case 8A — NRHO, True Anomaly 180 Degrees

Given the interest in the NRHO, several variations of the orbit were defined to explore the sensitivity and repeatability of the NRHO orbit. The first of these variations, Scenario 8A, advances the orbit from Scenario 8 by approximately half of an orbital period, with a new starting time, and new initial position and velocity. The vehicle's orientation is also reset to align with the local VO frame, with the orientation quaternion given exactly. The new initial conditions were defined by simply averaging the positions and velocities of the various participating simulations at the time when the true anomaly was closest to 180 degrees. These ICs were found during the process of the assessment, and due to subsequent data updates, these ICs do not correspond to averaging the states from the final assessment files.

This case can be used to explore the effects of re-initializing a propagation, for example due to a navigation re-set or to re-synchronize to a reference orbit. A true anomaly of 180 degrees corresponds to the largest distance from the Moon, where the orbital velocity is at its lowest and the third-body perturbations are at their strongest. Differences in modeling third-body perturbations are expected to have a strong influence on differences between simulations, as early deviations propagate.

Constants and Models
The gravitational constant is defined in the GRAIL model specified in Section 7.3.4.2. The ephemerides are: DE440. Time is expressed in UTC. Although test case initial conditions may

be in the future, the following constants are used: TAI − UTC = 37 sec; DUT1 = 0.0 sec Low-Fidelity GRAIL uses the GRAIL gravity model of degree and order 8 (i.e., 8x8 GRAIL).			
Initial Conditions			
Scenario	Lunar Case 8a: NRHO — True Anomaly 180		
Vehicle	Apollo Vehicle Model		
Orientation	Major axis is approximately along the velocity vector quaternionWrtMi = [0.930023544773565 0.319852453640438 -0.0390525111437773 0.176707428933728] Where the components are ordered [W X Y Z]'		
Rotation	bodyAngularRateWrtMi: [0.001, 0, 0] deg/s (angular rotation rate of the body frame relative to MI presented in the body frame)		
Gravitation	Low-Fidelity GRAIL	**Gravity Gradient Effects**	Off
Sun/Earth Perturbations	On	**Duration**	604,800 s
Output Interval	60 s	**Output Format**	CSV
Initial States			
Time	2026-1-31 9:42:03.51 UTC corresponds to output j2000UtcTime_s J2000 epoch of January 1.5, 2000 = -2451545.0 days Using 37 leap seconds j2000TtTime_s = j2000UtcTime + 32.184 + 37 = j2000UtcTime + 69.184 (s) Note that delta UT1/UTC is not relevant for check-case and is thus not specified		
Frame	Planet-Centered Inertial		
Position (m)	miPosition_m_X miPosition_m_Y miPosition_m_Z	-2845919.11896933 42880801.2794537 -55813619.8570753	
Velocity (m/s)	miVelocity_m_s_X miVelocity_m_s_Y miVelocity_m_s_Z	66.3176950654487 21.5521352604019 13.1774942999742	

7.7.12 Check-case 8B — NRHO, True Anomaly 0 Degrees

The second NRHO variation is similar to the first. Scenario 8B advances the orbit from scenario 8 by approximately one complete orbital period. This case uses a new starting time and new initial position and velocity. The vehicle's orientation is re-set to align with the local VO frame, with the orientation quaternion given exactly. Similar to the previous case, the new position and velocity were found by averaging values from Scenario 8. As with the previous case, due to subsequent data updates the ICs do not correspond to averaging the states from the final assessment files.

Like Scenario 8A, this case can be used to explore the effects of re-initialization. A true anomaly of 0 degrees corresponds to the closest approach to the Moon, where the orbital velocity is at its

highest and third-body perturbations are at their weakest. Differences in modeling the 8x8 GRAIL gravity model are expected to have a strong influence on differences between simulations, as early deviations propagate. The NRHO is quasi-stable, and a real vehicle will require occasional delta-v maneuvers to stay in the neighborhood of the orbit. Scenario 8B approximately corresponds to a second revolution of the NRHO with no delta-v or station-keeping applied.

Constants and Models			
The gravitational constant is defined in the GRAIL model specified in Section 7.3.4.2. The ephemerides are: DE440. Time is expressed in UTC. Although test case initial conditions may be in the future, the following constants are used: TAI − UTC = 37 sec; DUT1 = 0.0 sec Low-Fidelity GRAIL uses the GRAIL gravity model of degree and order 8 (i.e., 8x8 GRAIL).			
Initial Conditions			
Scenario	Lunar Case 8b: NRHO — True Anomaly 0		
Vehicle	Apollo Vehicle Model		
Orientation	Major axis is approximately along the velocity vector quaternionWrtMi = [0.12713225279123 -0.533645080051352 0.809006527171233 0.211113139883695] Where the components are ordered [W X Y Z]'		
Rotation	bodyAngularRateWrtMi: [0.001, 0, 0] deg/s (angular rotation rate of the body frame relative to MI presented in the body frame)		
Gravitation	Low-Fidelity GRAIL	**Gravity Gradient Effects**	Off
Sun/Earth Perturbations	On	**Duration**	604,800 s
Output Interval	60 s	**Output Format**	CSV
Initial States			

Time	2026-2-3 14:02:03.51 UTC corresponds to output j2000UtcTime_s J2000 epoch of January 1.5, 2000 = -2451545.0 days Using 37 leap seconds j2000TtTime_s = j2000UtcTime + 32.184 + 37 = j2000UtcTime + 69.184 (s) Note that delta UT1/UTC is not relevant for check-case and is thus not specified	
Frame	Planet-Centered Inertial	
Position (m)	miPosition_m_X miPosition_m_Y miPosition_m_Z	64480.6573661784 -1568752.35550803 2887684.20305855

Velocity (m/s)	miVelocity_m_s_X miVelocity_m_s_Y miVelocity_m_s_Z	-675.900720893134 -1376.03922351634 -729.640352061474

7.7.13 Check-case 8C — NRHO, Delta Radius

Scenario 8C is identical to the basic NRHO case, Scenario 8, except for the initial position vector. The magnitude of the initial position is increased by 10 meters. An exhaustive analysis of initial condition dispersions was considered to be prohibitive in terms of simulation time and data output, so the 10-meter delta-radius was selected as one of two small initial condition dispersions. In addition to comparing simulations with each other, case 8C can be used to study the effects of small initial condition dispersions for a given simulation.

Constants and Models			
The gravitational constant is defined in the GRAIL model specified in Section 7.3.4.2. The ephemerides are: DE440. Time is expressed in UTC. Although test case initial conditions may be in the future, the following constants are used: TAI − UTC = 37 sec; DUT1 = 0.0 sec Low-Fidelity GRAIL uses the GRAIL gravity model of degree and order 8 (i.e., 8x8 GRAIL).			
Initial Conditions			
Scenario	Lunar Case 8c: NRHO — Delta Radius		
Vehicle	Apollo Vehicle Model		
Orientation	Major axis is approximately along the velocity vector quaternionWrtMi = [0.646080387557766 0.334383101748747 0.685546322693561 0.0281835682522564] Where the components are ordered [W X Y Z]'		
Rotation	bodyAngularRateWrtMi: [0.001, 0, 0] deg/s (angular rotation rate of the body frame relative to MI presented in the body frame)		
Gravitation	Low-Fidelity GRAIL	**Gravity Gradient Effects**	Off
Sun/Earth Perturbations	On	**Duration**	604,800 s
Output Interval	60 s	**Output Format**	CSV
Initial States			
Time	2026-1-28 6:42:03.51 UTC corresponds to output j2000UtcTime_s J2000 epoch of January 1.5, 2000 = -2451545.0 days Using 37 leap seconds j2000TtTime_s = j2000UtcTime + 32.184 + 37 = j2000UtcTime + 69.184 (s) Note that delta UT1/UTC is not relevant for check-case and is thus not specified		
Frame	Planet-Centered Inertial		

Position (m)	miPosition_m_X miPosition_m_Y miPosition_m_Z	-5838149.197823560 2538928.800553022 1055567.832616975
Velocity (m/s)	miVelocity_m_s_X miVelocity_m_s_Y miVelocity_m_s_Z	-685.576059820879546 751.297639180154464 -621.913991301902946

7.7.14 Check-case 8D — NRHO, Delta Velocity

Scenario 8D is identical to the basic NRHO case, Scenario 8, except for the initial velocity vector. The magnitude of the initial velocity is increased by 0.1 m/s. Scenario 8D is the second small initial condition dispersion. As with case 8C, case 8D can be used to compare simulations with each other, as well asand to study the effects of small initial condition dispersions for a given simulation.

Constants and Models			
The gravitational constant is defined in the GRAIL model specified in Section 7.3.4.2. The ephemerides are: DE440. Time is expressed in UTC. Although test case initial conditions may be in the future, the following constants are used: TAI − UTC = 37 sec; DUT1 = 0.0 sec Low-Fidelity GRAIL uses the GRAIL gravity model of degree and order 8 (i.e., 8x8 GRAIL).			
Initial Conditions			
Scenario	Lunar Case 8d: NRHO — Delta Velocity		
Vehicle	Apollo Vehicle Model		
Orientation	Major axis is approximately along the velocity vector quaternionWrtMi = [0.646080387557766 0.334383101748747 0.685546322693561 0.0281835682522564] Where the components are ordered [W X Y Z]'		
Rotation	bodyAngularRateWrtMi: [0.001, 0, 0] deg/s (angular rotation rate of the body frame relative to MI presented in the body frame)		
Gravitation	Low-Fidelity GRAIL	**Gravity Gradient Effects**	Off
Sun/Earth Perturbations	On	**Duration**	604,800 s
Output Interval	60 s	**Output Format**	CSV
Initial States			
Time	2026-1-28 6:42:03.51 UTC corresponds to output j2000UtcTime_s J2000 epoch of January 1.5, 2000 = -2451545.0 days Using 37 leap seconds j2000TtTime_s = j2000UtcTime + 32.184 + 37 = j2000UtcTime + 69.184 (s) Note that delta UT1/UTC is not relevant for check-case and is thus not specified		
Frame	Planet-Centered Inertial		

Position (m)	miPosition_m_X miPosition_m_Y miPosition_m_Z	-5838140.15098070377 2538924.86620857380 1055566.19690058869
Velocity (m/s)	miVelocity_m_s_X miVelocity_m_s_Y miVelocity_m_s_Z	-685.6335669887992 751.3606591746176 -621.9661583981967

7.7.15 Check-case 9 — Polar Orbit, Sensor Position A

Scenario 9 implements an orbit similar to the low lunar orbit from the first three cases but aligned to be nearly polar in the PA frame. The PA frame was chosen as the GRAIL model is implemented in the PA frame, and a polar orbit comes close to the origin of this frame. As with Scenario 8, the initial orientation quaternion is specified explicitly.

Scenario 9 introduces a sensor station that is offset from the vehicle CoM, testing the translation of states to a position away from the CoM. (see Section 7.4.5). The sensor is used in cases 9, 9A, and 9B, which implement a combination of sensor positions and moment profiles applied to the vehicle to excite the sensor outputs.

Two sensor positions are used as follows for the three cases as shown in Table F below. Note that since all cases have a fixed CoM, the body frame (centered at the CoM) was used as if it were a structural frame. More complicated simulations with a varying CoM locations would define a separate structural frame fixed to the structure to define sensor positions.

Table F. Sensor Positions by Case Number

	Sensor Position in body coordinates (m)		
Case	**X**	**Y**	**Z**
9	1.0	0.0	0.0
9A	0.5	1.0	-2.0
9B	0.5	1.0	-2.0

Finally, Scenario 9 includes two test points to test the PA and ME planet-fixed frame conversions and the ingestion of a DEM. DEMs are of great interest to simulate the local topography of the Moon for landing and surface missions. Test point 1 is given as a constant location, while test point 2 moves as a function of time. Test points 1 and 2 are not tied to the vehicle's motion. Test point 1 was added specifically to test conversion of data from ME to PA coordinates. Test point 2 is a simple means of testing DEM ingestion, and it was convenient to tie the moving test point to a simulation that already moves forward through time. (see Section 7.4.4)

Test point 1 is defined at a fixed location for the entire elapsed time and is defined in MEM coordinates as follows. Test point 1 related outputs should be fixed values for the entire run.

Latitude MEM = -3.64530 (deg)
Longitude MEM = -17.47136 (deg)
Altitude = 6000.0 (m) – Altitude above IAU 2015 reference Spheroid
Reference IAU 2015 spheroid for latitude and longitude calculations.
Latitude positive north, Longitude positive east

Test point 2 is defined at a location in MEM coordinates which changes with elapsed time. The test point position is completely independent of the vehicle location and is used as a look into the

DEM data (see Section 7.3.2.4). The starting point of test point 2 at elapsed time 0 was explicitly chosen as a location with a high slope in the south pole region to expose differences in interpolation techniques or other DEM lookup issues. The following calculation for the location of test point 2 in MEM coordinates results in migration across 2 degrees of latitude and 3 circles of longitude for a 28,800 second run:

Latitude MEM = -89.91137 + 2.0 * t/28800.0 (deg)
Longitude MEM = 127.26573 + 3.0 * 360.0 * t/28800.0 (deg)

Where:

t=elapsedTime_s
Reference IAU 2015 spheroid for latitude and longitude calculations.
Latitude positive north, Longitude positive east

Constants and Models			
The gravitational constant is defined in the GRAIL model specified in Section 7.3.4.2. The ephemerides are: DE440. Time is expressed in UTC. Although test case initial conditions may be in the future, the following constants are used: TAI − UTC = 37 sec; DUT1 = 0.0 sec Low-Fidelity GRAIL uses the GRAIL gravity model of degree and order 8 (i.e., 8x8 GRAIL).			
Initial Conditions			
Scenario	Lunar Case 9: Polar Orbit— Sensor Position A		
Vehicle	Apollo Vehicle Model		
Orientation	Major axis is approximately along the velocity vector quaternionWrtMi = [0.137636538714161 -0.722405126130431 0.66454559455915 0.132537427556468] Where the components are ordered [W X Y Z]'		
Rotation	bodyAngularRateWrtMi: [0, -0.05008, 0] deg/s (angular rotation rate of the body frame relative to MI presented in the body frame)		
Sensor Position	Position A Body Coordinates = [1.0, 0.0, 0.0] m		
Gravitation	Low-Fidelity GRAIL	**Gravity Gradient Effects**	Off
Sun/Earth Perturbations	On	**Duration**	28,800 s
Output Interval	60 s	**Output Format**	CSV
Initial States			
Time	2026-1-28 6:42:03.51 UTC corresponds to output j2000UtcTime_s J2000 epoch of January 1.5, 2000 = -2451545.0 days Using 37 leap seconds j2000TtTime_s = j2000UtcTime + 32.184 + 37 = j2000UtcTime + 69.184 (s) Note that delta UT1/UTC is not relevant for check-case and is thus not specified		

Frame	Planet-Centered Inertial	
Position (m)	miPosition_m_X miPosition_m_Y miPosition_m_Z	16399.51954901463 -696336.1908584647 1722730.536496220
Velocity (m/s)	miVelocity_m_s_X miVelocity_m_s_Y miVelocity_m_s_Z	132.0661535881654 -1500.512866792463 -607.7719139949826

7.7.16 Check-case 9A — Polar Orbit, Sensor Position B

Scenario 9A is identical to Scenario 9, except for the sensor station and lack of test point data. The sensor station for Scenario 9A is offset from the CoM in all three axes as discussed in Section 7.7.15.

Constants and Models			
The gravitational constant is defined in the GRAIL model specified in Section 7.3.4.2. The ephemerides are: DE440. Time is expressed in UTC. Although test case initial conditions may be in the future, the following constants are used: TAI − UTC = 37 sec; DUT1 = 0.0 sec Low-Fidelity GRAIL uses the GRAIL gravity model of degree and order 8 (i.e., 8x8 GRAIL).			
Initial Conditions			
Scenario	Lunar Case 9a: Polar Orbit— Sensor Position B		
Vehicle	Apollo Vehicle Model		
Orientation	Major axis is approximately along the velocity vector quaternionWrtMi = [0.137636538714161 -0.722405126130431 0.66454559455915 0.132537427556468] Where the components are ordered [W X Y Z]'		
Rotation	bodyAngularRateWrtMi: [0, -0.05008, 0] deg/s (angular rotation rate of the body frame relative to MI presented in the body frame)		
Sensor Position	Position B Body Coordinates = [0.5, 1.0, -2.0] m		
Gravitation	Low-Fidelity GRAIL	**Gravity Gradient Effects**	Off
Sun/Earth Perturbations	On	**Duration**	28,800 s
Output Interval	60 s	**Output Format**	CSV
Initial States			
Time	2026-1-28 6:42:03.51 UTC corresponds to output j2000UtcTime_s J2000 epoch of January 1.5, 2000 = -2451545.0 days Using 37 leap seconds j2000TtTime_s = j2000UtcTime + 32.184 + 37 = j2000UtcTime + 69.184 (s) Note that delta UT1/UTC is not relevant for check-case and is thus not specified		

Frame	Planet-Centered Inertial	
Position (m)	miPosition_m_X miPosition_m_Y miPosition_m_Z	16399.51954901463 -696336.1908584647 1722730.536496220
Velocity (m/s)	miVelocity_m_s_X miVelocity_m_s_Y miVelocity_m_s_Z	132.0661535881654 -1500.512866792463 -607.7719139949826

7.7.17 Check-case 9B, Polar Orbit, Sensor Position B with Moment Profile

Scenario 9B is identical to Scenario 9A, except for an added open-loop moment profile. The moment profile, which applies moments about various axes and combinations of axes, is specified in the descriptions of the output variables.

Additionally, case 9B adds a moment profile applied to the vehicle as shown in Table G below. This profile applies a repeating pattern of 12 moments of 60 seconds each as specified below, with the first moment profile starting at 30 seconds elapsed time. No moments are applied until 30 seconds into the simulation. The following table outlines the moment profiles. Below Table G, additional calculations are listed to assist in generating the profiles in a more generalized form.

The first column identifies the profile number (p). The second column lists the range of times for each profile for the first instance of each profile. The third column gives a more general form continuing the ranges over the remaining profile groupings. The final column lists what moments are applied to the vehicle about the body frame axes as plus or minus moments about the x, y, and z axes as noted.

Table G. Sensor Case Moment Profile

Profile (p)	Time Range (r=0) (seconds)	Time Range (r=[0,…]) (seconds)	Moment Application (about body frame axes)
1	[30,90)	[30+r*720, 90+r*720)	+Mx
2	[90,150)	[90+r*720, 150+r*720)	-Mx
3	[150,210)	[150+r*720, …)	+My
4	[210,270)	…	-My
5	[270,330)		+Mz
6	[330,390)		-Mz
7	[390,450)		+Mx +My
8	[450,510)		-Mx -My
9	[510,570)		+My +Mz
10	[570,630)		-My -Mz
11	[630,690)		+Mx +My +Mz
12	[690,750)		-Mx -My -Mz

Where:

Mx = +5.0 N-m
My = +4.0 N-m
Mz = +2.5 N-m

The time ranges corresponding to the third column can also be expressed as:

[y, y+60)

Where:

y = p*60+r*720-30
p = profile number corresponding to the current time in the range [1,12]
r = current time range index. 0 for the first set of profiles, 1 for the second set of profiles, etc.

Furthermore, a more generalized and useful calculation for determining the profile number in C pseudo code is as follows, recalling that no moments should be generated before 30 seconds:

```
p = int(floor(fmod((time+eps-30.0)/60.0,12.0))) + 1
```

Where:

time = simulation elapsed time in seconds, elapsedTime_s
eps = time epsilon value in seconds
int() = C style cast of value to an integer
floor() = function returning the largest integral value not greater than the argument
fmod(x,y) = floating-point remainder function. Floating-point remainder of dividing argument x by argument y.

For fixed-step solvers, the recommended epsilon value (eps) should be equal to one-half of the fixed-step solver time step size. This is to ensure that the profile starts at the correct time if there is any drift in the elapsedTime_s value due to numerical round-off. Variable-step solvers should use a small enough epsilon value to start the profiles at the correct time while not erroneously starting one frame too early.

<table>
<tr><th colspan="4">Constants and Models</th></tr>
<tr><td colspan="4">The gravitational constant is defined in the GRAIL model specified in Section 7.3.4.2. The ephemerides are: DE440. Time is expressed in UTC. Although test case initial conditions may be in the future, the following constants are used: TAI − UTC = 37 sec; DUT1 = 0.0 sec Low-Fidelity GRAIL uses the GRAIL gravity model of degree and order 8 (i.e., 8x8 GRAIL).</td></tr>
<tr><td colspan="4">This scenario uses an open-loop moment profile</td></tr>
<tr><th colspan="4">Initial Conditions</th></tr>
<tr><td>Scenario</td><td colspan="3">Lunar Case 9b: Polar Orbit - Sensor Position B with Moment Profile</td></tr>
<tr><td>Vehicle</td><td colspan="3">Apollo Vehicle Model</td></tr>
<tr><td>Orientation</td><td colspan="3">Major axis is approximately along the velocity vector
quaternionWrtMi =
[0.137636538714161 -0.722405126130431 0.66454559455915 0.132537427556468]
Where the components are ordered [W X Y Z]'</td></tr>
<tr><td>Rotation</td><td colspan="3">bodyAngularRateWrtMi: [0, -0.05008, 0] deg/s
(angular rotation rate of the body frame relative to MI presented in the body frame)</td></tr>
<tr><td>Sensor Position</td><td colspan="3">Position B
Body Coordinates = [0.5, 1.0, -2.0] m</td></tr>
<tr><td>Gravitation</td><td>Low-Fidelity GRAIL</td><td>Gravity Gradient Effects</td><td>Off</td></tr>
</table>

Sun/Earth Perturbations	On	**Duration**	28,800 s
Output Interval	60 s	**Output Format**	CSV
Initial States			
Time	2026-1-28 6:42:03.51 UTC corresponds to output j2000UtcTime_s J2000 epoch of January 1.5, 2000 = -2451545.0 days Using 37 leap seconds j2000TtTime_s = j2000UtcTime + 32.184 + 37 = j2000UtcTime + 69.184 (s) Note that delta UT1/UTC is not relevant for check-case and is thus not specified		
Frame	Planet-Centered Inertial		
Position (m)	miPosition_m_X miPosition_m_Y miPosition_m_Z	16399.51954901463 -696336.1908584647 1722730.536496220	
Velocity (m/s)	miVelocity_m_s_X miVelocity_m_s_Y miVelocity_m_s_Z	132.0661535881654 -1500.512866792463 -607.7719139949826	

8.0 Findings, Observations, and NESC Recommendations

8.1 Findings

F-1. Scenarios must be precisely defined and documented prior to implementation to minimize potential misinterpretation.

F-2. A checksum for each output file is useful to verify the integrity of output data.

F-3. The ability to upload results and compare parameters with other simulations is helpful in troubleshooting model implementation and interpretation of ground rules and assumptions.

F-4. Evaluating results to be 'in family' (e.g., NASA-STD-7009) depends on a Program definition of what is acceptable for the simulation's intended purpose. When assessing a given simulation's credibility using the methodology established in NASA STD 7009 for Models and Simulations, comparisons with other simulations provide the basis for increasing the relative score for model validation.

F-5. As in the previous 2015 assessment [ref. 12], many developers participating in this activity made improvements to their simulations to do comparisons; and generally resulted in improved understanding for new users unfamiliar with simulation heritage code.

F-6. This assessment contributes to increased confidence when using the simulation results in decision-making by providing partial verification and validation of simulations for lunar exploration including equations of motion, environment models, integrators, and frames, per NASA STD 7009.

F-7. The existence of two planet-fixed frames (ME and PA) for the Moon and their implementation in simulation tools is a common source of differences in simulation results. The small difference between the two frames can easily be overlooked if not examined closely.

8.2 Observation

O-1. It is helpful to have a direct display of differences in data at the initial frame on the plotting tool. Evaluating raw simulation output data or differences (e.g., plotting) between simulation parameters at the initial frame can reveal discrepancies that will propagate into larger differences.

O-2. Certain frame conventions can have multiple definitions and interpretations that may present a challenge when comparing simulations or ingesting data from different sources.

O-3. Using SPICE (Spacecraft, Planet, Instrument, C-Matrix, Events) as a tool requires awareness of the order of kernel usage, proper kernel usage, naming convention of kernels, time frame definitions, and value precision to minimize conversions from/to SPICE. Insufficient understanding of these definitions may lead to numerical errors and/or SPICE precision accuracy issues.

O-4. Maintaining check-case definition configuration control ensures accurate simulation comparisons (e.g., using identical data inputs and common ground rules and assumptions).

O-5. Commercial off-the-shelf tools can change numerical precision; therefore, care should be taken to avoid unintentionally modifying data.

O-6. Each simulation is designed for specific purposes and therefore may not be suitable to execute a given check-case.

O-7. Developing even simple vehicle models to share between simulations is problematic given the differences in model implementation (e.g., implementing a closed-loop GNC to share between simulations was a challenge in short time frames, and was ultimately not included in this assessment).

O-8. For this assessment, all participating simulations used SPICE.

O-9. Using the previous assessment [ref. 12], *Six Degree of Freedom Flight Simulation Check-Cases Findings Observations and Recommendations* positively influenced this assessment as guidance for assessment and helped reduce potential errors.

8.3 NESC Recommendations

The following recommendations are made to NASA's simulation development teams.

R-1. Use these check cases to help minimize errors in lunar simulation tools.

R-2. Seek agreement with Programs on a standard for sharing models and parameter data for ease of collaboration and dynamic model data exchange. [*F-1, F-2, O-2, O-3, O-4, O-8*]

R-3. Verify the intended use of the simulation before comparing simulation data to the check-cases in this assessment, or any generic check-case, to avoid invalidating a potentially useful simulation. [*F-3, O-6*]

The following recommendation is directed to the Office of Chief Engineer and Mission Directorates (ESDMD, SMD, ARMD)

R-4. Maintaining an online interactive repository of simulation raw output data to allow users to upload simulation outputs for comparisons is valuable. This online interactive repository should be maintained and hosted by an external organization (external to simulation developers). [*F-5, O-1*]

R-5. The Agency should adopt standards for defining and using coordinate systems and reference frames and, for specifying how data exchange occurs between simulations and models. [*F-1, O-2, O-4*]

9.0 Alternate Technical Opinion(s)

No alternate technical opinions were identified during the course of this assessment by the NESC assessment team or the NESC Review Board.

10.0 Other Deliverables

The outcome of this investigation will be published as a NASA TM and made available without restriction.

Check-case data will be provided as comma-separated values on the NESC Flight Simulation Check-case website.

11.0 Definition of Terms

[Use the template definitions provided below, and list additional terms that need to be defined to the readers/reviewers of this report. The Technical Editors will help identify terms unique to your assessment. Any unused terms from the list below may be deleted.]

Corrective Action	Changes to design processes, work instructions, workmanship practices, training, inspections, tests, procedures, specifications, drawings, tools, equipment, facilities, resources, or material that result in preventing, minimizing, or limiting the potential for recurrence of a problem.
Finding	A relevant factual conclusion and/or issue that is within the assessment scope and that the team has rigorously based on data from their independent analyses, tests, inspections, and/or reviews of technical documentation.
Lesson Learned	Knowledge, understanding, or conclusive insight gained by experience that may benefit other current or future NASA programs and projects. The experience may be positive e.g., a successful test or mission, or negative, as in a mishap or failure.
Observation	A noteworthy fact, issue, and/or risk, which is not directly within the assessment scope, but could generate a separate issue or concern if not addressed. Alternatively, an observation can be a positive acknowledgement of a Center/Program/Project/Organization's operational structure, tools, and/or support.
Problem	The subject of the independent technical assessment.

Proximate Cause	The event(s), including any condition(s) that existed immediately before the undesired outcome, that directly resulted in its occurrence.
Recommendation	A proposed measurable stakeholder action directly supported by specific Finding(s) and/or Observation(s) that will correct or mitigate an identified issue or risk.
Root Cause	One or multiple causes (including adverse or unplanned events, conditions, or organizational factors) that contributed to or created the proximate cause(s) and subsequent undesired outcome and, if eliminated or modified, should have prevented the undesired outcome.
Supporting Narrative	A paragraph, or section, in an NESC final report that provides a detailed explanation of a succinctly worded finding or observation. For example, the logical deduction that led to a finding or observation, descriptions of assumptions, exceptions, clarifications, and boundary conditions.

12.0 Acronyms, Abbreviations, and Nomenclature List

Acronyms and Abbreviations

3DOF	3 degrees of freedom
6DOF	6 degrees of freedom
AIAA	American Institute of Aeronautics and Astronautics
ARC	Ames Research Center
CHIP-WG	Crewed Human Landing System Interfaces for Piloting Working Group
CoM	Center of Mass
CrewCo	Crew Compartment
CSV	Comma Separated Values
DARTS	Dynamics Algorithms for Real Time Simulation
DDL	Deorbit, Descent, and Landing
DE	Development Ephemerides
DEM	Digital Elevation Model
DOF	Degrees-of-freedom
DSENDS	Dynamics Simulator for Entry, Descent, and Surface Landing
EMEJ2K	Earth Mean Equator Mean Equinox J2000
GDR	Gridded Data Record
GLASS	GeneraLized Aerospace Simulation in Simulink
GNC	Guidance, Navigation, and Control
GRAIL	Gravity Recovery and Interior Laboratory
GRC	Glenn Research Center
HLO	High Lunar Orbit
HLS	Human Landing Systems
IAU	International Astronomical Union
J2000	Earth-centered Inertial Frame for Epoch 2000
JEOD	JSC Engineering Orbital Dynamics
JPL	Jet Propulsion Laboratory
JSC	Johnson Space Center
KSC	Kennedy Space Center

LaRC	Langley Research Center
LLO	low lunar orbit
LOLA	Lunar Orbital Laser Altimeter
LPO	lunar polar orbit
LPS	Lunar Polar Stereographic
LVLH	Local Vertical, Local Horizontal
M&S	Modeling and Simulation
MAVERIC	Marshall Aerospace Vehicle Representation in C
ME	Mean Earth
MI	Moon-centered Inertial
MSFC	Marshall Space Flight Center
NASA	National Aeronautics and Space Administration
NESC	NASA Engineering and Safety Center
NPR	NASA Procedural Requirement
NRHO	Near-rectilinear Halo Orbit
ODE	Ordinary Differential Equation
PA	Principal Axis
PDS	Planetary Data Sciences
POST2	Program to Optimize Simulated Trajectories II
RPOD	Rendezvous, Proximity Operations, and Docking
SPICE	Spacecraft, Planet, Instrument, C-Matrix, Events
STARS	Space Transportation and Aeronautics Research Simulation
TAI	International Atomic Time
TFrames	Tools to Facilitate the Rapid Assembly of Missile Engagement Simulations
TDB	Barycentric Dynamical Time
TDT	Technical Discipline Team
TT	Terrestrial Time
UTC	Coordinated Universal Time
VO	Vehicle-carried Orbit-defined

Nomenclature

[x,y]	Mathematical inclusion, e.g., range is >= x and <=y
(x,y)	Mathematical exclusion, e.g., range is > x and <y
DUT1	Delta between UTC and Universal Time (UT1) (UT1-UTC)
Mx, My, Mz	Moments applied about the body x,y, and z axis respectively
dm	Differential of variable m (mass)
I_{xz}	Moment or product of inertia (e.g., x-z product of inertia)
$k0$	LPS central scale factor
m	Mass
r_M	Lunar radius for spherical moon reference model (IAU)
φ	Planetocentric latitude
$\varphi 0$	LPS center latitude

13.0 References

1. Margolis, B. W. L., and Lyons, K. R.,“Condor,” New Technology Report ARC-18996-1, NASA Ames Research Center, November 2023.
2. B. W. L. Margolis, "SimuPy: A Python framework for modeling and simulating dynamical systems," The Journal of Open Source Software, vol. 2, p. 396, September 2017.
3. B. W. L. Margolis and K. R. Lyons, "SimuPy Flight Vehicle Toolkit," Journal of Open Source Software, vol. 7, p. 4299, July 2022.
4. J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," Mathematical Programming Computation, vol. 11, p. 1–36, 2019.
5. A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker and C. S. Woodward, "SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers," ACM Transactions on Mathematical Software (TOMS), vol. 31, p. 363–396, 2005.
6. E. Hairer, S. P. Norsett and G. Wanner, “Solving Ordinary Differential Equations I: Nonstiff Problems”, Springer Berlin Heidelberg, 1993.
7. B. W. L. Margolis, "A Sweeping Gradient Method for Ordinary Differential Equations with Events," Journal of Optimization Theory and Applications, vol. 199, p. 600–638, October 2023.
8. Dynamics Algorithms for Real Time Simulation (DARTS) Lab web site https://dartslab.jpl.nasa.gov.
9. Cameron, J.; Jain, A.; Burkhart, D.; Bailey, E.; Balaram, B.; Bonfiglio, E.; Grip, H; Ivanov, M.; Sklyanskiy, E: “DSENDS: Multi-mission Flight Dynamics Simulator for NASA Missions”. In Proceedings of the AIAA SPACE Conference, pp. 5421, Long Beach, CA, Sep. 2016.
10. National Aeronautics and Space Administration. 1971. “CSM/LM Spacecraft Operational Data Book. Volume III: Mass Properties. Revision 3” Pg. 2-15 and Pg. 3.1-10. April 1, 1971
11. Whitley, Ryan J. et al. “Earth-Moon Near Rectilinear Halo and Butterfly Orbits for Lunar Surface Exploration.” AAS 18-406.
12. NASA/TM-2015-218675: "Check-Cases for Verification of 6-Degree-of-Freedom Flight Vehicle Simulations
13. “Report of the IAU Working Group on Cartographic Coordinates and Rotational Elements 2015”, https://doi.org/10.1007/s10569-017-9805-5
14. J.P. Synder, “Map Projections - A Working Manual” 1987 USGPP 1395, p. 155.
15. SISO-STD-018-2020 “Standard for Space Reference Federation Object Model (SpaceFOM)” 25 October 2019, Appendix E
16. Flight Dynamics Model Exchange Standard (ANSI/AIAA S-119-2011)
17. Lunar Grid Reference System for Artemis Mission Navigation and Lunar Surface Science, M.T. McClernan, B.A. Archinal, and T.M. Hare; U.S. Geological Survey
18. The JPL Planetary and Lunar Ephemerides DE440 and DE441; Ryan S. Park, William M. Folkner, James G. Williams, and Dale H. Boggs. https://iopscience.iop.org/article/10.3847/1538-3881/abd414
19. JSC Engineering Orbital Dynamics (JEOD),” JEOD GitHub Site: https://github.com/nasa/jeod , September 2023
20. NASA Langley Research Center Flight Simulation Facilities https://flightsimulation.larc.nasa.gov