Verification of SpaceAGORA.jl 6-DoF Dynamics Using University of Michigan SmallSats Telemetry

Evan Yu
Dept. of Aerospace Engineering
University of Michigan
Ann Arbor, United States
evany@umich.edu

James Cutler
Dept. of Aerospace Engineering
University of Michigan
Ann Arbor, United States
jwcutler@umich.edu

Giusy Falcone
Dept. of Aerospace Engineering
University of Michigan
Ann Arbor, United States
falconeg@umich.edu

Abstract — High-fidelity six-degree-of-freedom simulation is a key enabler for developing and verifying autonomous guidance, navigation, and control algorithms. This paper validates SpaceAGORA.jl, an open-source Julia-based simulator that propagates fully coupled orbital and rigid-body attitude dynamics with modular environmental and actuator models. Validation is performed by comparing orbital and attitude dynamics with Basilisk simulations and flight telemetry from operational University of Michigan SmallSat missions (CYGNSS and GRIFEX). Results demonstrate a maximum position error compared to Basilisk within approximately 65 meters and attitude error on the order of 1e-2 radians. Comparisons with real-world CYGNSS and GRIFEX telemetry indicate that SpaceAGORA.jl accurately replicates spacecraft translational dynamics, achieving relative position differences within 1% or less, relative to the total position magnitude. In addition, a telemetry-driven replay of a CYGNSS slew demonstrates that the simulator reproduces measured attitude evolution when supplied with a wheel torque/momentum history consistent with telemetry. These validation results establish SpaceAGORA.jl as a reliable simulation environment for developing and validating autonomy and adaptive GNC algorithms under realistic operational conditions.

Table of Contents

1. Introduction

High-fidelity simulation is central to the development, verification, and risk reduction of spacecraft guidance, navigation, and control (GNC) and autonomy algorithms [1]. For modern missions, end-to-end performance often depends on the coupled evolution of orbit and attitude: environmental torques affect pointing and momentum management, actuator limits constrain achievable control authority, and attitude motion can modulate disturbance forces (e.g., drag and solar radiation pressure), particularly for small spacecraft with large area-to-mass ratios. As a result, six-degree-of-freedom (6-DOF) propagation is routinely required to evaluate closed-loop GNC behavior and to support telemetry-driven analysis and replay workflows [2, 3].

A broad ecosystem of spacecraft simulation tools exists, spanning government-developed software (e.g., GMAT [4]), commercial suites (e.g., STK [5] and FreeFlyer [6]), and open-source frameworks. While commercial and government tools provide mature capabilities, licensing costs, access restrictions, and limited extensibility can hinder reproducible research and rapid model iteration. Open-source frameworks such as Basilisk provide a modular, high-fidelity 6-DOF environment with established adoption in the research community [7]. However, autonomy-centric workflows often benefit from (i) a single-language implementation that simplifies direct extension of core dynamics and models, and (ii) configuration-driven orchestration that makes simulation settings, input data paths, and outputs reproducible and directly comparable to flight telemetry.

SpaceAGORA.jl [8] is an open-source 6-DOF simulation platform developed in Julia to support these needs. It propagates coupled orbital and rigid-body attitude dynamics and includes modular models for gravity harmonics and third-body gravity, solar radiation pressure, atmospheric drag, geomagnetic torques, and common actuators (reaction wheels, magnetic torque rods, and thrusters). SpaceAGORA.jl also interfaces with NAIF SPICE kernels for ephemerides and frame transformations and can use the GRAM suite for atmospheric density; analytic rarefied-flow coefficient models are available to support future atmospheric-flight studies.

The focus of this paper is validation of SpaceAGORA.jl in the orbit-and-attitude regime relevant to autonomy and GNC development. Specifically, this paper contributes: (i) cross-validation of SpaceAGORA.jl against Basilisk across representative uncontrolled and controlled attitude scenarios, including environmental torque cases and actuator-driven detumbling with re-

979-8-3315-7360-7/26/$31.00

played command histories. (ii) On-orbit consistency assessment using operational University of Michigan SmallSat datasets: CYGNSS [9] navigation and attitude telemetry and GRIFEX [10] orbit reconstruction from public two-line elements (TLEs), including a telemetry-driven replay of a representative CYGNSS slew maneuver.

The remainder of the paper is structured as follows: Sec. 2 summarizes the CYGNSS and GRIFEX missions and the available telemetry products. Sec. 3 describes the SpaceAGORA.jl software architecture and configuration-driven workflow. Sec. 4 details the translational and rotational dynamics models, environmental perturbations, and actuator representations. Sec. 5 outlines the Basilisk and telemetry comparison methodology and error metrics. Sec. 6 presents validation results, and Sec. 7 summarizes validation outcomes.

2. Background

This study validates SpaceAGORA.jl using telemetry from two University of Michigan SmallSat missions characterized by complementary operational modes: the Cyclone Global Navigation Satellite System (CYGNSS) and the GEO-CAPE ROIC In-Flight Performance Experiment (GRIFEX). CYGNSS, launched on December 15, 2016, comprises an eight-satellite constellation utilizing Global Navigation Satellite System reflectometry (GNSS-R) to measure tropical cyclone wind speeds and ocean surface conditions [9]. GRIFEX, launched on January 31, 2015, is a 3U CubeSat technology demonstrator developed to verify on-orbit performance of a high-frame-rate Readout Integrated Circuit (ROIC) coupled with a Focal Plane Array (FPA) payload [10]. Previous validation studies have leveraged telemetry data from other university-operated spacecraft, including ASTERIA [11], MicroMAS-1 [12], UWE-3 [13], ESTCube-1 [14], and PicSat [15], demonstrating the importance and utility of flight data for high-fidelity simulator validation.

Each CYGNSS satellite is three-axis stabilized, employing attitude determination sensors including a star tracker, sun sensors, and a three-axis magnetometer. Attitude control is executed through a combination of reaction wheels and magnetic torque rods. The integrated sensor suite achieves an attitude determination accuracy of approximately 2.1° (3 standard deviations), with control actuators providing attitude precision of approximately 2.8° (3 standard deviations) during nadir-pointing science operations. CYGNSS engineering telemetry data, specifically attitude quaternions, angular body rates, and reaction wheel speeds, form the basis for validating the translational and rotational dynamics computed by SpaceAGORA.jl.

GRIFEX is a magnetometer-based attitude sensing mission jointly developed by NASA's Jet Propulsion Laboratory (JPL) and the Michigan eXploration Laboratory (MXL). Launched as a secondary payload accompanying NASA's Soil Moisture Active Passive (SMAP) mission, GRIFEX lacks active attitude control systems, relying instead solely on passive magnetometer-based attitude estimation. Due to the absence of precise onboard navigation instrumentation, GRIFEX's translational states are reconstructed post-facto using the Simplified General Perturbations-4 (SGP4) algorithm from publicly available TLE datasets published by the U.S. Space Force and SAIC's Space-Track. Such TLE-based orbit reconstructions typically yield position uncertainties of approximately one kilometer at epoch, which increase over time.

3. SpaceAGORA.jl Architecture

SpaceAGORA.jl is developed in Julia to leverage its near-compiled execution speed, ease-of-use, and readability, enabling researchers and students to directly modify core dynamics and physical models without resorting to lower-level languages, e.g., C or C++. This direct editability significantly accelerates the development cycle. External dependencies include GRAM for atmospheric density and variability, interfaced through the Julia-Python bridge `PythonCall.jl`, and NAIF SPICE kernels for planet ephemerides, body orientation, and frame transformations. The GRAM/SPICE integration follows previous work in aerobraking trajectory simulations, facilitating consistent environmental modeling across multiple celestial bodies and reuse of existing analysis pipelines [8].

SpaceAGORA.jl employs a configuration-driven orchestration, where users define each simulation scenario through a single configuration file specifying vehicle properties, mission timelines, numerical integrator tolerances, model selections, data storage paths, and output cadence, as shown in Figure 1. Double-headed arrows indicate input and output. At runtime, the simulator initializes required GRAM and SPICE modules, assembles the chosen physical models (orbital and attitude dynamics, gravitational fields, atmospheric interactions, solar radiation pressure, actuator dynamics), and executes the 6-DOF trajectory propagation from user-defined initial conditions.

To optimize memory usage and ensure data robustness during extended simulations, SpaceAGORA.jl partitions execution by orbit, delineated by true anomaly intervals, saving simulation data incrementally at the completion of each orbit. By default, results are written in the Apache Arrow format, providing a minimal-overhead, language-agnostic data solution. SpaceAGORA.jl also provides the option to save to a CSV file, ensuring transparency and ease of analysis.

Within the scope of this paper, SpaceAGORA.jl integrates fully coupled orbital and attitude dynamics, central-body and third-body gravitational modeling, solar radiation pressure (SRP) effects, GRAM-based atmospheric density modeling, and detailed actuator dynamics for reaction wheels, thrusters, and magnetic torque rods. The simulator architecture ensures these models are modular and easily interchangeable or upgradeable without requiring extensive refactoring of the core integration or data handling frameworks.

SpaceAGORA.jl is publicly accessible on GitHub at github.com/Space-FALCON-Lab/SpaceAGORA.jl and includes a Docker container setup to ensure reproducibility and portability across computing environments by explicitly managing compiler versions, Python/GRAM wheels, and SPICE kernel versions. Currently, GRAM dependency on Python distributions limits compatibility with ARM64-based Mac; the provided Docker container

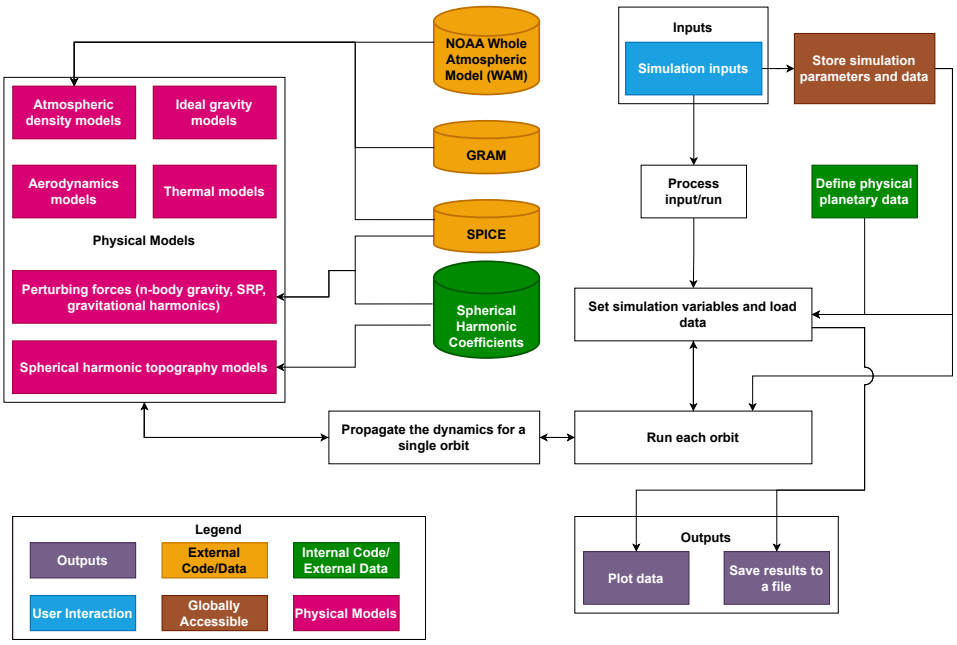


Figure 1. SpaceAGORA.jl high-level architecture and control flow. A single configuration file selects vehicle shape, physical models (gravity, GRAM-based atmospheres, solar radiation pressure, actuators), defines numerical tolerances, output cadence, and specifies paths to SPICE kernels and data directories.

mitigates this issue, offering cross-platform consistency until full native ARM64 compatibility is available.

For the results presented in the paper, GRAM Suite Version 2.0 is used with the 2023 MERRA-2 model for EarthGRAM. The SPICE kernels used in this analysis are: `pck00010.tpc`, `naif0012.tls`, `de430.bsp`. For gravity harmonics, the GGM05 model is used up to degree and order 360. The WMM2025 model is used for the geomagnetic model.

Integrator

The default numerical integrator employed in SpaceAGORA.jl is Tsitouras' adaptive-timestep, fifth-order Runge-Kutta method [16]. Default integration tolerances are set to a relative tolerance of 1e-9 and absolute tolerance of 1e-11. Within each integration step, all relevant force and torque models are updated, as described in Sec. 4. After each timestep, stopping conditions are verified, and if applicable, control inputs for the next timestep are computed.

For non-atmospheric flight, such as the cases presented in this paper, the integration tolerances are fixed for the duration of the simulation. In addition, to accommodate long or small-timestep simulations in which the stored data may exceed available memory, SpaceAGORA.jl may be configured to write data to an output file at a fixed, custom rate rather than at the end of each orbit. This limits the amount of data being stored in memory at any given time, providing a much higher limit on the simulation length.

4. SpaceAGORA.jl Dynamics

This section discusses the dynamics, disturbance, and actuator models implemented in SpaceAGORA.jl to model full translational and rotational motion. The spacecraft is modeled as a kinematic chain of links connected by joints. For instance, a spacecraft comprising a central hub and two solar panels is modeled with three links (one hub and two panels) and two joints connecting the panels to the hub. The contribution of forces and torques from each body can then be calculated to determine the net force and torque acting on the spacecraft. By treating the spacecraft as a combination of basic geometries such as flat plates and boxes, analytical solutions for drag and lift coefficients from [17] can be applied directly. This decomposition simplifies the calculation of the aerodynamic properties and facilitates modification in the spacecraft geometry, making it particularly suited for analyzing articulated drag modulation strategies. In the current implementation, joint states are not dynamically integrated; they are assumed rigid and can only be updated kinematically.

Notation and Frames

SpaceAGORA.jl defines three primary reference frames. The inertial frame is the planet-centric inertial frame described in [8]. The planet-fixed frame follows the International Astronomical Union (IAU) body-fixed conventions [18]. The spacecraft body-fixed frame is attached to the root link (the main bus or central hub), while link frames are attached to individual, non-root links, e.g., the solar panels.

Vectors expressed in the inertial (planet–centered) frame use a superscript (R); vectors in the spacecraft body frame use (B). A direction–cosine matrix $\boldsymbol{\mathcal{O}}_{B/R}$ maps

$\boldsymbol{a}^{(R)}$ to $\boldsymbol{a}^{(B)}$ via $\boldsymbol{a}^{(B)} = \boldsymbol{\mathcal{O}}_{B/R}\boldsymbol{a}^{(R)}$, with the transpose mapping back: $\boldsymbol{\mathcal{O}}_{R/B} = \boldsymbol{\mathcal{O}}_{B/R}^{T}$.

The spacecraft is modeled as a set of links $\mathcal{L}$; each link $\ell \in \mathcal{L}$ contains a set of flat facets $\mathcal{F}_\ell$. Facet $k \in \mathcal{F}_\ell$ has area A_k, unit normal $\hat{\boldsymbol{n}}_k$ (defined in the facet or link frame and rotated as needed), and center-of-pressure location $\boldsymbol{r}_k^{(B)}$ relative to the vehicle center of mass. Throughout this paper, unit vectors carry a hat (e.g., $\hat{\boldsymbol{d}}$). Scalars are italic, vectors are bold, matrices are capitalized and bold, and frame superscripts are shown only when necessary to avoid ambiguity.

Translational Dynamics

The translational dynamics in SpaceAGORA.jl use $\boldsymbol{x} \in \mathbb{R}^6$ to describe the translational state, with the first three components describing the position and the second three describing the velocity, all expressed in the inertial frame. The state vector and its derivative are:

$$\boldsymbol{x} = [p_x \quad p_y \quad p_z \quad v_x \quad v_y \quad v_z]^T \tag{1}$$

$$\dot{\boldsymbol{x}} = \begin{bmatrix} v_x & v_y & v_z & F_{\text{net},x}^{(R)}/m & F_{\text{net},y}^{(R)}/m & F_{\text{net},z}^{(R)}/m \end{bmatrix}^T \tag{2}$$

Here, p_x, p_y, p_z are the position components; v_x, v_y, v_z are the velocity components; $F_{net,x}$, $F_{net,y}$, $F_{net,z}$ are the component of the total external force in the inertial frame; and m is the total mass of the spacecraft.

The net force includes central-body gravity and all modeled perturbations (third-body gravity, atmospheric drag, solar radiation pressure, actuator forces). Its calculation follows [8], with updates to the solar radiation pressure and aerodynamic models described in Sec. 4.

Rotational Dynamics

SpaceAGORA.jl represents spacecraft attitude using the scalar-last quaternion convention. Quaternions were selected due to their singularity-free representation and widespread use in spacecraft simulations and missions. The attitude quaternion and spacecraft angular velocity are propagated using the standard quaternion kinematic equation, given in (3), and Euler's rotational dynamics equation provided in (4), which explicitly includes the angular momentum contributions of onboard reaction wheels:

$$\dot{\boldsymbol{q}} = \frac{1}{2}\Xi(\boldsymbol{q})\boldsymbol{\omega} \tag{3}$$

$$\dot{\boldsymbol{\omega}} = \boldsymbol{J}^{-1}\Big(\boldsymbol{\tau}_{\text{ext}}^{(B)} - \dot{\boldsymbol{h}}_{rw} - \boldsymbol{\omega} \times (\boldsymbol{J}\boldsymbol{\omega} + \boldsymbol{h}_{rw})\Big) \tag{4}$$

In these equations, $\boldsymbol{q}$ is the quaternion representing the rotation from the inertial frame to the spacecraft body-fixed frame and is defined as $\boldsymbol{q} = [q_1, q_2, q_3, q_4]^T$, where q_4 is the scalar component. The vector $\boldsymbol{\omega}$ is the spacecraft angular velocity expressed in the body frame, and $\boldsymbol{J}$ is the constant inertia matrix expressed in the body-fixed frame. The reaction wheel angular momentum vector $\boldsymbol{h}_{rw}$ and its time derivative $\dot{\boldsymbol{h}}_{rw}$ represent internal angular momentum exchange due to reaction wheel torques.

The matrix $\Xi(\boldsymbol{q})$ in (3) is defined as:

$$\Xi(\boldsymbol{q}) = \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \tag{5}$$

To maintain numerical stability and accuracy, the quaternion $\boldsymbol{q}$ is normalized at each integration step to preserve the unit-norm constraint. Additionally, the spacecraft inertia matrix $\boldsymbol{J}$ remains constant in the body-fixed frame and is not updated when applying kinematic updates to joint states.

Perturbation Models

Perturbation models in SpaceAGORA.jl are separated into perturbing forces and perturbing torques. This subsection summarizes the key perturbations implemented in SpaceAGORA.jl, highlighting updates to models since [8]. The implementation of third-body gravity and gravitational harmonics remains unchanged from [8] and is not discussed further here.

Solar Radiation Pressure— The SRP model uses the aforementioned facet-based geometry approach to calculate forces and torques on complex spacecraft shapes, following the methodology of [19]. Each facet, in addition to the prior geometric properties, has surface optical coefficients (specular reflection coefficient δ_k and diffuse reflection coefficient ρ_k). Self-shadowing effects and non-uniform surface properties within a facet are not modeled, but facets can differ from one another. The SRP model also supports time-varying spacecraft geometry, such as articulating solar arrays.

The SRP force calculation for each facet k is performed in the inertial frame as given by Vallado [20], with the net SRP force obtained by summing contributions from all illuminated facets:

$$P = \frac{1361.0}{c\, n_{AU}^2}, \tag{6}$$

$$\begin{aligned} \boldsymbol{f}_{\text{SRP}}^{(R)} = &- P\,\sigma \sum_{k \in \cup_\ell \mathcal{F}_\ell} A_k \cos\alpha_k \\ &\Big[(1-\delta_k)\,\hat{\boldsymbol{s}} + 2\big(\tfrac{\rho_k}{3} + \delta_k \cos\alpha_k\big)\,\hat{\boldsymbol{n}}_k^{(R)}\Big] \end{aligned} \tag{7}$$

where P is the solar radiation pressure magnitude at spacecraft distance n_{AU} (in AU), σ is computed at each epoch using Vallado's conical umbra model [20] and applied uniformly to all facets, 1361.0 $\text{W}\cdot\text{m}^2$ is the solar radiation flux at a distance of 1 AU, c is the speed of light in $\text{m}\cdot\text{s}^{-1}$.

The complete SRP calculation is summarized in Alg. (1), where α_k is the incident angle cosine and $\boldsymbol{r}_{S/SC}^{(R)}$ is the vector pointing from the spacecraft to the Sun.

This calculation is repeated for each facet; forces and torques are summed to obtain total SRP contributions in inertial and body frames, respectively.

Aerodynamics—The aerodynamic model in SpaceAGORA.jl models all links as rectangular prisms of varying

Algorithm 1 Solar radiation pressure calculation

$\hat{\boldsymbol{s}} \leftarrow \boldsymbol{r}_{S/SC}^{(R)} / \|\boldsymbol{r}_{S/SC}^{(R)}\|$ (Spacecraft → Sun)
$\cos\_\alpha_k \leftarrow \max\big(0, \hat{\boldsymbol{n}}_k{}^{(R)} \cdot \hat{\boldsymbol{s}}\big)$
if $\cos\_\alpha_k > 0$ **then**
  $\boldsymbol{f}_{SRP,k}^{(R)} \leftarrow$ Force calculated in (7)
  $\boldsymbol{\tau}_{SRP,k}^{(B)} \leftarrow \boldsymbol{r}_k^{(B)} \times (\boldsymbol{\mathcal{O}}_{B/R} \boldsymbol{f}_{SRP,k}^{(R)})$
end if

dimensions. To model a flat plate, such as the solar panel, one dimension of the link can be set to a small number. Future development may include expanding to more shapes such as spheres and cylinders or adopting a facet model similar to the solar radiation pressure model.

Under the rectangular prism assumption, the aerodynamic force and moment coefficients are derived from analytic free-molecular solutions provided by Hart [17]. The body-axial, sideflow, and normal force coefficients are C_A, C_{Sf}, C_N, respectively, and the corresponding rolling, pitching, and yawing moments are C_l, C_m, and C_n.

These body-frame coefficients are converted to conventional aerodynamic coefficients (drag, lift, and side force coefficients, C_D, C_L, C_S) as defined in (8), where α is the angle of attack and β is the sideslip angle. C_{Sf} is the body sideflow coefficient.

$$\begin{aligned} C_D &= -\sin(\alpha) C_A + \cos(\alpha) C_N \\ C_L &= \cos(\alpha)\cos(\beta) C_A + \sin(\beta) C_{Sf} + \sin(\alpha)\cos(\beta) C_N \\ C_S &= -\cos(\alpha)\sin(\beta) C_A + \cos(\beta) C_{Sf} - \sin(\alpha)\sin(\beta) C_N \end{aligned} \tag{8}$$

The aerodynamic force on link ℓ, expressed in the spacecraft body frame, is calculated as:

$$\boldsymbol{f}_{\text{aero},\ell}^{(B)} = \tfrac{1}{2}\rho V_{\text{rel}}^2 A_\ell \big(C_{D,\ell} \hat{\boldsymbol{d}}^{(B)} + C_{L,\ell} \hat{\boldsymbol{l}}^{(B)} + C_{S,\ell} \hat{\boldsymbol{s}}^{(B)} \big) \tag{9}$$

where ρ is the atmospheric density, V_{rel} is the wind-relative velocity magnitude, A_ℓ is the reference area of the ℓ-th link. The aerodynamic unit vectors $\hat{\boldsymbol{d}}, \hat{\boldsymbol{l}}, \hat{\boldsymbol{s}}$ describing the drag, lift, and side force coefficients are computed as:

$$\hat{\boldsymbol{d}}^{(B)} \triangleq -\frac{\boldsymbol{v}_{rel}^{(B)}}{\|\boldsymbol{v}_{rel}^{(B)}\|};\ \hat{\boldsymbol{l}}^{(B)} \triangleq \frac{\hat{\boldsymbol{n}}_\ell^{(B)} - \left(\hat{\boldsymbol{n}}_\ell^{(B)} \cdot \hat{\boldsymbol{d}}^{(B)}\right) \hat{\boldsymbol{d}}^{(B)}}{\|\hat{\boldsymbol{n}}_\ell^{(B)} - \left(\hat{\boldsymbol{n}}_\ell^{(B)} \cdot \hat{\boldsymbol{d}}^{(B)}\right) \hat{\boldsymbol{d}}^{(B)}\|};$$
$$\hat{\boldsymbol{s}}^{(B)} = \hat{\boldsymbol{d}}^{(B)} \times \hat{\boldsymbol{l}}^{(B)} \tag{10}$$

Here, $\hat{\boldsymbol{n}}_\ell^{(B)}$ is a link-fixed reference normal used to define the lift direction (e.g., the outward normal of the primary reference face of the rectangular prism).

The aerodynamic torque contribution from each link ℓ is calculated from the force, link position, and moment coefficients and combines the torque due to the aerodynamic force offset from the spacecraft center of mass and the pure aerodynamic moments:

$$\begin{aligned} \boldsymbol{\tau}_{\text{aero},\ell}^{(B)} =& \boldsymbol{r}_\ell^{(B)} \times \boldsymbol{f}_{\text{aero},\ell}^{(B)} + \\ & \tfrac{1}{2}\rho V_{\text{rel}}^2 A_\ell \big(C_{l,\ell} l_1 \hat{\boldsymbol{e}}_1 + C_{m,\ell} l_2 \hat{\boldsymbol{e}}_2 + C_{n,\ell} l_3 \hat{\boldsymbol{e}}_3 \big) \end{aligned} \tag{11}$$

where $\boldsymbol{r}_\ell$ is the position of the link center-of-pressure relative to the spacecraft center-of-mass, l_1, l_2, l_3 are the spacecraft reference lengths, and $\hat{\boldsymbol{e}}_1, \hat{\boldsymbol{e}}_2, \hat{\boldsymbol{e}}_3$ are the elementary basis vectors in $\mathbb{R}^3$.

Gravity Gradient—The gravity gradient torque model in SpaceAGORA.jl follows the approximation presented in [19], and depends on the gravitational parameter of the central body, μ; the spacecraft inertia matrix, $\boldsymbol{J}$; and the spacecraft position vector expressed in the body-fixed frame. The torque is computed as:

$$\boldsymbol{\tau}_{GG}^{(B)} = \frac{3\mu}{||\boldsymbol{r}||^5} \boldsymbol{r}^{(B)} \times \boldsymbol{J}^{(B)} \boldsymbol{r}^{(B)} \tag{12}$$

Magnetic Torques—Spacecraft equipped with magnetorquers or those possessing a significant residual magnetic dipole experience magnetic torques due to interaction with the geomagnetic field. The magnetic torque, expressed in the spacecraft body-fixed frame, is given by:

$$\boldsymbol{\tau}_{\text{mag}}^{(B)} = \boldsymbol{m}^{(B)} \times \boldsymbol{B}^{(B)}, \qquad \boldsymbol{B}^{(B)} = \boldsymbol{\mathcal{O}}_{\boldsymbol{B/R}} \boldsymbol{B}^{(R)}. \tag{13}$$

where $\boldsymbol{m}^{(B)} \in \mathbb{R}^3$ is the spacecraft magnetic dipole moment ($\text{A}\cdot\text{m}^2$) defined in the body frame, and $\boldsymbol{B}^{(R)}$ (T) is the geomagnetic field vector obtained from an appropriate geomagnetic model and expressed in the inertial frame. The geomagnetic vector is transformed to the body frame to compute the torque.

Actuator Models

The primary attitude actuators currently modeled in SpaceAGORA.jl are reaction wheels and reaction control thrusters. This actuator combination is representative of many common spacecraft configurations.

Reaction Wheels—Reaction wheels accept motor torque magnitudes as control inputs, producing a net angular momentum change $\dot{\boldsymbol{h}}_{rw}$. Mathematically, the wheel set is represented by a reaction wheel Jacobian matrix $\boldsymbol{B} \in \mathbb{R}^{3\times n}$, where each column corresponds to the unit vector direction of an individual wheel's angular momentum in the spacecraft body-fixed frame. The commanded motor torques applied to the wheels are denoted by $\boldsymbol{u}_{rw} \in \mathbb{R}^n$. The net angular momentum change due to reaction wheels is thus computed by:

$$\dot{\boldsymbol{h}}_{rw} = \boldsymbol{B} \boldsymbol{u}_{rw} \tag{14}$$

This angular momentum exchange is incorporated directly into Euler's rotational equations (4). Each reaction wheel is modeled with a maximum allowable angular momentum, beyond which the wheel is considered saturated. A saturated wheel can no longer transfer additional angular momentum along its spin axis until a suitable desaturation command is issued.

Thrusters—The reaction control thruster (RCS) model in SpaceAGORA.jl accepts control torques as input and maps them to individual thruster force requests. Because RCS thrusters are typically binary actuators, meaning they are either on or off with no ability to modulate the thrust, the requested force is translated into an "on-time" over each control period. These steps follow the Basilisk thruster model [7]. The total impulse applied over the control period is then analytically integrated using a first-order filter model [7] for the ramp-up and ramp-down phases of the thrust profile. The resulting thrust force used in the numerical propagation is the average thrust required to achieve the total impulse in one control period.

The first step, mapping the commanded control torques to individual thruster forces, is accomplished using the Moore-Penrose pseudoinverse of the thruster torque Jacobian. The columns of the Jacobian matrix specify the torque produced by each thruster if it fires with a thrust of 1N. The vector $\boldsymbol{f}$ contains the thrust produced by each thruster. The torque produced by the i-th thruster about the body-fixed center-of-mass is:

$$\boldsymbol{\tau}_i^{(B)} = (\boldsymbol{r}_{t_i}^{(B)} \times \hat{\boldsymbol{g}}_{t_i}^{(B)}) f_i = \boldsymbol{d}_i f_i; \qquad i = 1, \ldots, n \quad (15)$$

where $f_i \in \mathbb{R}$ is the thrust force from the i-th thruster, that is, the i-th element of $\boldsymbol{f}$, $\hat{\boldsymbol{g}}_{t_i}$ is the unit thrust direction in the body-frame, and $\boldsymbol{r}_{ti}$ is the thruster location relative to the center-of-mass. Each $\boldsymbol{d}_i \in \mathbb{R}^3$ is stacked column-wise into the thruster Jacobian $\boldsymbol{D} \in \mathbb{R}^{3\times n}$:

$$\boldsymbol{D} = [\boldsymbol{d_1} \quad ... \quad \boldsymbol{d_n}] \quad (16)$$

The total torque requested from all thrusters is then:

$$\boldsymbol{\tau}_{RCS}^{(B)} = \boldsymbol{D}\boldsymbol{f} \quad (17)$$

where $\boldsymbol{f} \in \mathbb{R}^n$ is the vector of all individual thrust forces. The corresponding translational force produced by the RCS thrusters is:

$$\boldsymbol{f}_{RCS} = \sum_{i=1}^{n} f_i \hat{\boldsymbol{g}}_{t_i} \quad (18)$$

Because $\boldsymbol{D}$ typically has more columns than rows, (17) is underdetermined. Because of this, the Moore-Penrose pseudoinverse yields the minimum-norm least squares solution for $\boldsymbol{f}$, minimizing $||\boldsymbol{\tau}_{RCS} - \boldsymbol{D}\boldsymbol{f}||$:

$$\boldsymbol{f} = \boldsymbol{D}^{\dagger}\boldsymbol{\tau}_{RCS}^{(B)} \quad (19)$$

where † denotes the Moore-Penrose pseudoinverse.

If negative thrust values appear, the most negative value is subtracted from all thruster forces to ensure physically valid commands. The total RCS force in the inertial frame is obtained by transforming to the inertial frame:

$$\boldsymbol{F}_{\boldsymbol{RCS}}^{(\boldsymbol{R})} = \boldsymbol{\mathcal{O}}_{R/B} \sum_{i=1}^{n} F_i \hat{\boldsymbol{g}}_{ti}^{(B)} \quad (20)$$

The on-off nature requires thruster pulse modulation to achieve average thrusts over a control period not equal to the maximum thrust [21]. In addition, thrust profiles are not, in general, perfect square waves, instead exhibiting a ramp-up period upon activation and a ramp-down period upon shut-off.

To calculate the length of time per control period that the thruster must fire, the model assumes a perfect square-wave thrust profile. The on-time request, t_{on}, per control period, Δt, is calculated from the requested thrust force, F_i, and maximum thrust, F_{max}, using (21).

$$t_{on} = \begin{cases} \frac{F_i}{F_{max}} \Delta t & 0 \le F_i \le F_{max} \\ \Delta t & F_i > F_{max} \end{cases} \quad (21)$$

Once the nominal on-time has been determined, the total impulse over the control period is determined by analytically integrating the thruster ramp-up and ramp-down. The thrust factor, κ, describes the instantaneous fraction of maximum thrust output, and it is governed by:

$$\dot{\kappa} = \begin{cases} \omega(1-\kappa) & \text{Thrusting} \\ -\omega\kappa & \text{Not thrusting} \end{cases} \quad (22)$$

where ω is the cutoff frequency of the first-order filter.

To save computational effort, the closed-form solutions for these equations are used in the analytical integration of the thrust profile. The true thrust, T, at any given time is given by the equations in (23), where T_{max} is the maximum thrust of the thruster and κ_0 is the initial thrust factor.

$$T(t) = \begin{cases} (1 + (\kappa_0 - 1)e^{-\omega t})T_{max} & \text{Thrusting} \\ \kappa_0 e^{-\omega t} T_{max} & \text{Not thrusting} \end{cases} \quad (23)$$

The thruster is integrated using the ramp-up equations for $0 \le t \le t_{on}$ and the ramp-down equations for $t_{on} \le t \le \Delta t$. Thus, the analytical solution for the total impulse over the control period is given by (24), where κ_0 is again the initial thrust factor and κ_1 is the thrust factor at t_{on}.

$$\int_0^{\Delta t} T(t)dt = T_{max}\left(t_{on} + \frac{\kappa_0 - 1}{\omega}\left(1 - e^{-\omega t_{on}}\right)\right) + \frac{T_{max}\kappa_1}{\omega}\left(1 - e^{-\omega(\Delta t - t_{on})}\right) \tag{24}$$

$$\kappa_1 = 1 + (\kappa_0 - 1)e^{-\omega t_{on}} \tag{25}$$

The integrated thrust is then divided by Δt to determine the true average thrust over the control period, which is then used in the dynamics propagation.

Net Force and Torque

At each timestep, the total spacecraft force and torque are computed as the sum of all modeled environmental and actuator contributions. First, forces and torques from individual facets are summed per link, and then these link-level contributions are combined to obtain the spacecraft-level totals. Finally, gravity gradient torque and magnetic torque (if applicable) are included in the torque summation. The net external force and torque acting on the spacecraft are computed as:

$$\boldsymbol{F}_{\text{net}}^{(R)} = \underbrace{\boldsymbol{F}_{\text{grav}}^{(R)} + \boldsymbol{F}_{\text{third}}^{(R)}}_{\text{gravity}} + \underbrace{\sum_{\ell\in\mathcal{L}} \boldsymbol{\mathcal{O}}_{R/B}\boldsymbol{f}_{\text{aero},\ell}^{(B)}}_{\text{environment (aero)}} + \underbrace{\sum_{\ell\in\mathcal{L}}\sum_{k\in\mathcal{F}_\ell} \boldsymbol{f}_{\text{SRP},k}^{(R)}}_{\text{environment (SRP)}} + \underbrace{\boldsymbol{F}_{\text{RCS}}^{(R)}}_{\text{actuators}}$$

$$\boldsymbol{\tau}_{\text{ext}}^{(B)} = \underbrace{\boldsymbol{\tau}_{\text{GG}}^{(B)}}_{\text{gravity gradient}} + \underbrace{\sum_{\ell\in\mathcal{L}} \boldsymbol{\tau}_{\text{aero},\ell}^{(B)}}_{\text{environment (aero)}} + \underbrace{\sum_{\ell\in\mathcal{L}}\sum_{k\in\mathcal{F}_\ell} \boldsymbol{\tau}_{\text{SRP},k}^{(B)}}_{\text{environment (SRP)}} + \underbrace{\boldsymbol{\tau}_{\text{RCS}}^{(B)} + \boldsymbol{\tau}_{\text{mag}}^{(B)}}_{\text{actuators/other}}$$

5. Validation Methodology

Previous validations of SpaceAGORA.jl have focused on translational motion using GMAT comparisons and on trajectory reconstruction of aerobraking missions, such as Mars Odyssey [22] and Venus Express [23], as detailed in [8]. Those validations emphasized apoapsis and periapsis evolutions. In contrast, this paper focuses on detailed, high-fidelity trajectory and attitude validation through comparisons with validated simulators and available flight telemetry data.

To facilitate accurate comparisons between trajectories generated by SpaceAGORA.jl, other validated simulators, and spacecraft telemetry, all datasets are first synchronized using a common start epoch. To manage differing sampling frequencies, linear interpolation is applied to all translational states, while quaternion states utilize spherical linear interpolation (Slerp) [24] to preserve the unit-norm constraint.

Basilisk Comparison

Preliminary validation of orbital and attitude dynamics is performed through comparison with Basilisk [7]. Validation scenarios included gravity gradient torque, solar radiation pressure torque, reaction wheel detumbling, and reaction control thruster (RCS) detumbling. For cases involving environmental perturbations (gravity gradient and solar radiation pressure), no spacecraft actuators were active. Conversely, actuator validation cases (reaction wheels and RCS thrusters) excluded environmental disturbance torques. To isolate modeling differences, Basilisk-generated actuator command histories were replayed in SpaceAGORA.jl; thus, these cases primarily validate actuator and rigid-body dynamics consistency rather than control-law performance.

SpaceAGORA.jl and Basilisk share identical models for gravitational harmonics, employing Pines' uniform gravitational harmonics formulation [25] with matching degree and order. In all cases, the atmosphere is neglected. Additionally, both simulators use identical solar constants, optical coefficients, facet-based SRP geometry, spacecraft mass, inertia matrices, and actuator performance limits (reaction wheel maximum torque and angular momentum, thruster maximum thrust, and thruster impulse filtering parameters). Integration tolerances are selected as 1e-9 relative tolerance and 1e-11 absolute tolerance.

Uncontrolled Motion Comparison— Initial conditions for the uncontrolled test cases, matching standard Basilisk test cases, are provided in Table 1. For the torque-free (TF) scenario, an initial angular velocity of $[0.001, -0.01, 0.03]$ rad/s was applied, consistent with subsequent actuator test scenarios. The gravity gradient and solar radiation pressure test cases began with zero initial angular velocity to isolate and highlight environmental torque effects. Each uncontrolled scenario was propagated for 1 million seconds (approximately 11.6 days). Basilisk simulations employed the built-in adaptive timestep RKF45 integrator, a 4th-order Runge-Kutta method.

Controlled Motion Comparison— Controlled attitude simulations involved reaction wheel and RCS thruster detumbling scenarios. Initial conditions for these tests, summarized in Table 2, matched those of the uncontrolled scenarios, except for the angular velocity, which was identical to the torque-free initial condition. Each controlled scenario was propagated for 600 seconds, sufficient time for the spacecraft to fully detumble and reach a stable attitude.

For the reaction wheel detumbling comparison, control commands are generated at 10 Hz in Basilisk. However, the reaction wheel torque data sampled at 100 Hz in Basilisk were directly used as control input to SpaceAGORA.jl, applied via a zero-order hold at 100 Hz to ensure accurate capture of wheel torque transient behavior. For the RCS detumbling scenario, thruster force magnitudes computed in Basilisk were directly input into SpaceAGORA.jl. SpaceAGORA.jl then employed the filtered-impulse thruster model described in Sec. 4 to calculate mean thrust levels during each control period.

Telemetry Data

Telemetry data used in this validation study are sourced from two University of Michigan SmallSat missions: CYGNSS [9] and GRIFEX [10].

The CYGNSS spacecraft have an average mass of ap-

Table 1. Uncontrolled Comparison Initial Conditions

State	Value
Semimajor Axis (km)	10,000
Eccentricity	0.1
Inclination (deg)	33.3
RAAN (deg)	48.2
AOP (deg)	347.8
True Anomaly (deg)	85.3
Quaternion	[0.175, 0.351, -0.526, 0.754]
Ang. Vel. (rad/s)	$\begin{cases}[0.001, -0.01, 0.03] & \text{TF}\\ [0.0, 0.0, 0.0] & \text{else}\end{cases}$
Date	06 November 2001
Time	19:00:00 UTC
Duration	1e6 seconds

Table 2. Controlled Comparison Initial Conditions

State	Value
Semimajor Axis (km)	10,000
Eccentricity	0.1
Inclination (deg)	33.3
RAAN (deg)	48.2
AOP (deg)	347.8
True Anomaly (deg)	85.3
Quaternion	[0.175, 0.351, -0.526, 0.754]
Angular Velocity (rad/s)	[0.001, -0.01, 0.03]
Date	06 November 2001
Time	19:00:00 UTC
Duration	600 seconds

proximately 28.93 kg. Due to export restrictions, the inertia matrix is omitted. Mission-specific surface optical coefficients are not publicly available; therefore, both the CYGNSS and GRIFEX simulations use the nominal SRP optical coefficients from the Basilisk examples listed in Table 3.

Table 3. SRP optical coefficients used for CYGNSS and GRIFEX.

Surface	Specular δ	Diffuse ρ
Bus	0.336	0.139
Solar panel (front)	0.16	0.16
Solar panel (back)	0.00	0.56

The CYGNSS dataset spans a 48-hour interval sampled at 1 Hz. Additionally, higher-frequency (4 Hz) telemetry is used specifically to validate SpaceAGORA.jl performance during spacecraft slew maneuvers.

The GRIFEX mission (GEO-CAPE ROIC In-Flight Performance Experiment) is a 3U CubeSat developed by the Michigan eXploration Lab (MXL) at the University of Michigan. GRIFEX has a nominal mass of 3.0 kg and a spacecraft body-fixed inertia matrix:

$$\boldsymbol{J}_{GRIF} = \begin{bmatrix} 1.720\times10^{-2} & 7.100\times10^{-5} & 6.210\times10^{-4} \\ 7.100\times10^{-5} & 1.874\times10^{-2} & 9.930\times10^{-4} \\ 6.210\times10^{-4} & 9.930\times10^{-4} & 1.697\times10^{-3} \end{bmatrix} \text{kg m}^2$$

For GRIFEX, translational reference states are reconstructed via SGP4 propagation of publicly available TLEs rather than onboard navigation telemetry. Its telemetry provides approximately one data point every six seconds, and the validation time frame is restricted to a matching 48-hour period for consistency with CYGNSS.

The initial conditions used in this validation are summarized in Tables 4 and 5.

The telemetry data inherently contain uncertainties from measurement error and state estimation. For CYGNSS, the reported attitude estimation uncertainty is approximately 2.1° (3 standard deviations). GRIFEX translational states, derived from publicly available TLEs, carry uncertainties on the order of 1 km at epoch, increasing over time. Furthermore, GRIFEX attitude estimates are reconstructed from relatively low-frequency and noisy magnetometer measurements. Position uncertainty also propagates errors into the magnetic field model, further affecting attitude accuracy. Thus, the telemetry data serve primarily as a consistency check within the known uncertainty bounds, rather than as absolute truth states.

Error Metrics

Validation comparisons of SpaceAGORA.jl with both Basilisk and spacecraft telemetry utilize absolute numerical error and normalized relative error metrics for translational states. Additionally, force and torque values are directly compared in validation scenarios involving Basilisk. Relative errors for translational position (ε_r) and velocity (ε_v) are computed by normalizing the magnitude of their absolute differences by the magnitude of the corresponding reference vector. Similarly, force (ε_F) and torque (ε_τ) errors are computed by normalizing the magnitude of absolute differences by the maximum magnitude of the reference values observed throughout each simulation case. This normalization avoids artificially large relative error spikes near zero-crossing events. The translational and force/torque relative errors are defined mathematically as:

$$\varepsilon_r(t) = \frac{\left\|\mathbf{r}(t) - \mathbf{r}_{\text{ref}}(t)\right\|}{\left\|\mathbf{r}_{\text{ref}}(t)\right\|},\ \varepsilon_v(t) = \frac{\left\|\mathbf{v}(t) - \mathbf{v}_{\text{ref}}(t)\right\|}{\left\|\mathbf{v}_{\text{ref}}(t)\right\|} \tag{26}$$

$$\varepsilon_F(t) = \frac{\|\boldsymbol{F}(t) - \boldsymbol{F}_{\text{ref}}(t)\|}{\max_t \|\boldsymbol{F}_{\text{ref}}(t)\|},\ \varepsilon_\tau(t) = \frac{\|\boldsymbol{\tau}(t) - \boldsymbol{\tau}_{\text{ref}}(t)\|}{\max_t \|\boldsymbol{\tau}_{\text{ref}}(t)\|} \tag{27}$$

where $\boldsymbol{x}_{ref}(t)$ denotes the reference ("truth") state at time t and $\boldsymbol{x}$ denotes the state simulated by SpaceAGORA.jl.

Attitude errors are represented using the principal rotation angle $\delta\theta(t)$ associated with the error quaternion $\boldsymbol{\delta q}(t)$. Following [19], the error quaternion is

$$\boldsymbol{\delta q}(t) = \boldsymbol{q}_A(t) \otimes \boldsymbol{q}_B^{-1}(t), \qquad \delta\theta(t) = 2\arccos\big(|\delta q_4(t)|\big), \tag{28}$$

Table 4. GRIFEX Comparison Initial Conditions

State	Value
Semimajor Axis (km)	6909.697477
Eccentricity	0.016026
Inclination (deg)	99.040943
RAAN (deg)	295.949024
AOP (deg)	3.706235
True Anomaly (deg)	256.941770
Quaternion	[0.9819, 0.1605, 0.06988, 0.07158]
Ang. Vel. (rad/s)	[-4.494e-3, -1.486e-2, 4.013e-4]
Date	05 September 2015
Time	01:18:42 UTC
Duration	48 hours

Table 5. CYGNSS Comparison Initial Conditions

State	Value
Semimajor Axis (km)	6818.8611
Eccentricity	4.7900e-4
Inclination (deg)	34.9357
RAAN (deg)	177.3709
AOP (deg)	140.6318
True Anomaly (deg)	276.6553
Quaternion	[0.3891, -0.09973, 0.8489, 0.3435]
Ang. Vel. (rad/s)	[5.8214e-5, -9.3680e-4, -6.8727e-4]
Date	06 June 2025
Time	00:00:00 UTC
Duration	48 hours

where $\otimes$ denotes quaternion multiplication, $\boldsymbol{q}^{-1}$ is the quaternion inverse (equal to the conjugate for unit quaternions), and δq_4 is the scalar component of $\boldsymbol{\delta q}$ (scalar-last convention).

6. Results

This section presents the results of comparisons between SpaceAGORA.jl, Basilisk, and telemetry data. Comparison with Basilisk provides a rigorous, high-fidelity baseline for model correctness and numerical propagation. Comparison with telemetry extends this to real-world applicability by testing SpaceAGORA.jl against flight data with realistic uncertainties, disturbances, and incomplete state information.

Basilisk Comparison

Tables 6 and 7 report maximum, 95th-percentile, and RMSE differences between SpaceAGORA.jl and Basilisk for uncontrolled and controlled validation scenarios. Uncontrolled cases include torque-free motion, gravity-gradient-only dynamics, and solar-radiation-pressure torque. The gravity-gradient-only case exhibits long-horizon phase sensitivity in attitude; therefore, that case is reported over a shorter window (1e5 s) to demonstrate torque-model agreement before numerical phase divergence dominates state differences.

Across the uncontrolled scenarios, RMS position differences are on the order of meters to tens of meters (RMSE ranging from 2.6–22.8 m and maximum difference $\leq$ 65.9 m). Attitude errors range from $\mathcal{O}(10^{-4})$ rad for torque-free and gravity-gradient cases to $\mathcal{O}(10^{-2})$ rad under SRP torque. Angular-rate differences remain below 10^{-5} rad/s. Normalized torque errors remain below 5×10^{-3} for gravity-gradient torque and below 10^{-1} for SRP torque.

For the controlled detumbling scenarios, Basilisk-generated actuator histories were replayed in SpaceAGORA.jl, so these cases validate rigid-body and actuator model consistency rather than controller performance. SpaceAGORA.jl reproduces Basilisk detumbling behavior with RMSE attitude errors of 1.09×10^{-2} rad for the reaction wheels case, and 2.46×10^{-2} rad for the thrusters case, and RMSE normalized torque errors of 1.66×10^{-3} and 7.36×10^{-3}, respectively. Translational differences remain sub-meter or smaller over the 600 s detumbling window.

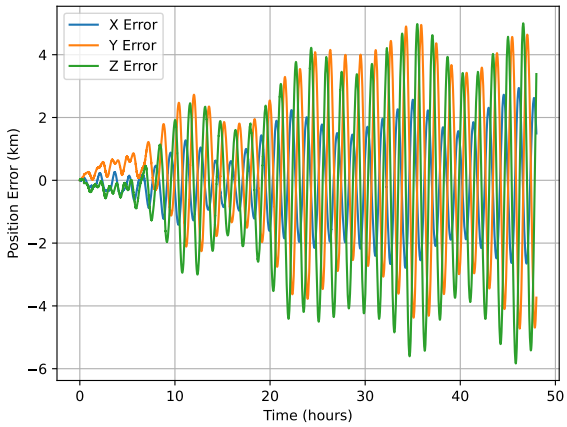


Figure 2. Position error of SpaceAGORA.jl-simulated trajectory versus GRIFEX telemetry.

Telemetry Comparison

GRIFEX—Position and velocity errors for GRIFEX over a 48-hour window (approximately 31 orbits) are shown in Figures 2 and 3. The position difference remains below approximately 5 km, corresponding to a relative position error below 0.07%. The velocity error remains within approximately 6 m/s. SpaceAGORA.jl was configured with gravitational harmonics to degree and order 50, third-body gravity from the Sun and Moon, solar radiation pressure, and GRAM-based atmospheric drag. Because the GRIFEX reference trajectory is reconstructed from publicly available TLEs via SGP4 rather than onboard navigation telemetry, these errors reflect a combination of model mismatch and the intrinsic uncertainty of TLE-based orbit reconstruction over multi-day horizons.

GRIFEX does not provide onboard quaternion and body-rate telemetry—only raw magnetometer and photodiode measurements are available. Because of this, attitude validation is deferred in this work.

Table 6. Basilisk Uncontrolled Comparison Results

Scenario	Torque-Free			Gravity Gradient			Solar Radiation		
Error Metric	Max	95%	RMSE	Max	95%	RMSE	Max	95%	RMSE
Attitude Error (rad)	6.820e-4	2.679e-4	1.822e-4	2.047e-3	1.684e-3	3.595e-4	7.563e-2	5.578e-2	2.861e-2
Ang. Vel. Error (rad/s)	5.511e-6	2.301e-6	1.601e-6	1.766e-10	1.198e-10	6.938e-11	1.126e-5	6.737e-6	3.136e-6
Torque Error (nd)	N/A	N/A	N/A	4.506e-3	2.296e-3	9.225e-4	9.019e-2	5.851e-2	2.816e-2
Pos. Error (m)	3.982e1	3.157e1	2.064e1	5.189	4.239	2.555	6.586e1	4.824e1	2.282e1
Rel. Pos. Error (nd)	4.424e-6	3.352e-6	2.048e-6	3.502e-5	1.121e-6	7.019e-6	7.478e-6	4.876e-6	2.511e-6

Table 7. Basilisk Controlled Comparison Results

Scenario	Reaction Wheel Detumbling			Thruster Detumbling		
Error Metric	Max	95%	RMSE	Max	95%	RMSE
Attitude Error (rad)	1.723e-2	1.649e-2	1.089e-2	2.645e-2	2.645e-2	2.463e-2
Angular Velocity Error (rad/s)	7.943e-5	7.716e-5	6.817e-5	2.781e-3	5.952e-4	3.244e-4
Torque Error (nd)	1.064e-2	3.061e-3	1.655e-3	1.216e-1	1.003e-2	7.362e-3
Position Error (m)	3.178e-1	1.209e-1	9.212e-2	2.410e-6	2.256e-6	1.231e-6
Relative Position Error (nd)	3.169e-8	1.205e-8	9.203e-9	2.403e-13	2.250e-13	1.230e-13

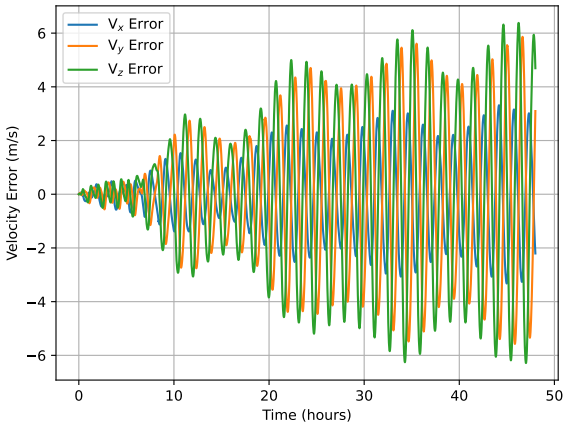


Figure 3. Velocity error of SpaceAGORA.jl-simulated trajectory versus GRIFEX telemetry.

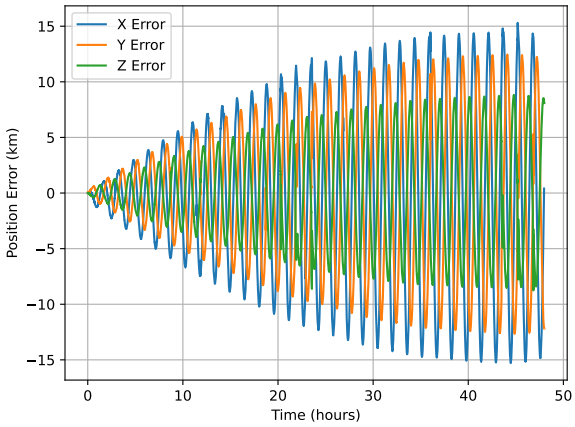


Figure 4. Position error of SpaceAGORA.jl-simulated trajectory versus CYGNSS telemetry.

CYGNSS— Position and velocity errors between SpaceAGORA and CYGNSS telemetry over a 48-hour interval (approximately 31 orbits) are presented in Figures 4 and 5. Throughout this window, the inertial position discrepancies remain bounded within approximately 15 km, equating to a relative positional error of approximately 0.2%. The velocity error remains within approximately 17 m/s. Due to GPS blackouts, the position and velocity data has been filtered to remove significant outliers in the error. Given the use of nominal environmental modeling, simplified spacecraft geometry, and intermittent gaps in telemetry data, this comparison serves primarily as a propagation-level consistency check rather than a precise orbit determination benchmark.

To validate high-rate attitude dynamics and actuator state propagation, a 3600-second representative slew maneuver from CYGNSS telemetry, sampled at 4 Hz, was analyzed. SpaceAGORA.jl was initialized with telemetry-derived initial conditions (as reported in Table 8), and all subsequent data were synchronized to a common $t = 0$ epoch.

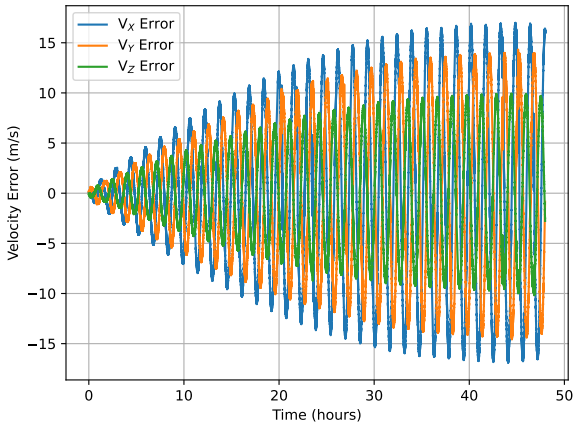


Figure 5. Velocity error of SpaceAGORA.jl-simulated trajectory versus CYGNSS telemetry.

Table 8. CYGNSS Slew Comparison Initial Conditions

State	Value
Semimajor Axis (km)	6815.5199
Eccentricity	1.1602e-3
Inclination (deg)	35.0063
RAAN (deg)	143.5194
AOP (deg)	175.8049
True Anomaly (deg)	276.6553
Quaternion	[-0.7693, -0.0287, 0.3684, 0.5211]
Ang. Vel. (rad/s)	[9.1679e-5, -1.1931e-3, 1.2954e-4]
Date	04 October 2025
Time	00:00:00 UTC
Duration	3600 seconds

Direct open-loop replay of the flight reaction wheel telemetry was not anticipated to exactly replicate the observed closed-loop attitude trajectory due to the absence of the on-board controller and detailed actuator dynamics. Therefore, we applied an inverse-dynamics approach, reconstructing the net body torque required to reproduce the observed attitude and angular velocity profiles using Euler's rigid-body equation (given in (4)) with external torques disabled. This reconstructed torque was distributed across individual reaction wheels using the Moore–Penrose pseudoinverse of the reaction wheel Jacobian, analogous to the method shown in (19), and subsequently replayed within SpaceAGORA.jl.

The results of the inverse-dynamics replay are illustrated in Figures 6 and 7, which compare quaternion and body-rate component trajectories between telemetry and SpaceAGORA.jl. The error magnitudes are instead presented in Table 9. Over the analyzed slew segment, the attitude error magnitude was characterized by an RMSE of 3.56×10^{-2} rad and a 95th-percentile error of 4.24×10^{-2} rad. The corresponding body-rate error magnitudes were smaller, with an RMSE of 4.61×10^{-5} rad/s and a 95th-percentile error of 7.20×10^{-5} rad/s.

Table 9. CYGNSS Slew Replay Error Statistics

Metric	RMSE	95%	Max
Attitude error $\delta\theta$ (rad), inverse-dynamics replay	3.427×10^{-2}	4.237×10^{-2}	4.245×10^{-2}
Body-rate error $\|\Delta\boldsymbol{\omega}\|$ (rad/s), inverse-dynamics replay	4.605×10^{-5}	7.201×10^{-5}	2.810×10^{-4}
Wheel speed error $\|\Delta\boldsymbol{\Omega}_{rw}\|$ (rpm), $\dot{\boldsymbol{h}}_{rw}$ replay	7.905	13.997	124.502
Wheel momentum error $\|\Delta\boldsymbol{h}_{rw}\|$ (Nms), $\dot{\boldsymbol{h}}_{rw}$ replay	2.371×10^{-5}	4.199×10^{-5}	3.735×10^{-4}

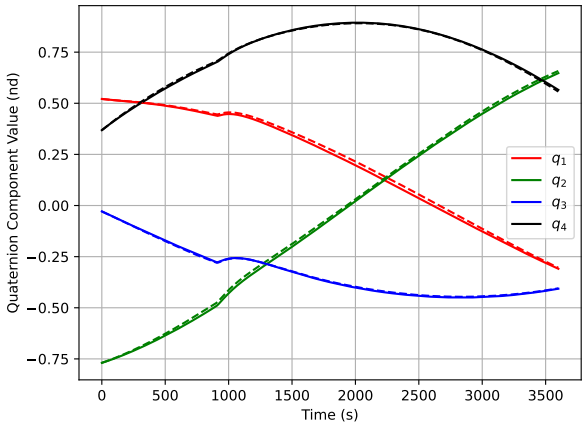


Figure 6. CYGNSS slew quaternion comparison with SpaceAGORA.jl results.

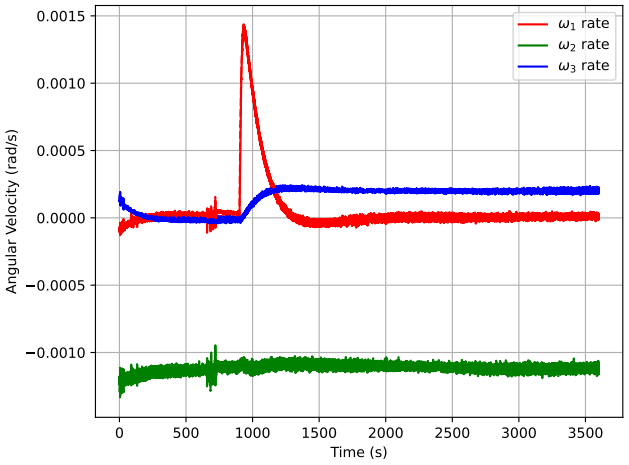


Figure 7. CYGNSS slew angular velocity comparison with SpaceAGORA.jl

As a complementary validation of reaction-wheel state propagation and timeline alignment, a separate replay was conducted using telemetry-derived reaction wheel momentum rates, $\dot{\mathbf{h}}_{rw}$. This replay was intended specifically for validation of wheel-state integration rather than closed-loop attitude matching. Figure 8 demonstrates comparisons of reaction-wheel speeds between telemetry and SpaceAGORA.jl; reaction-wheel momentum over-

lays are omitted for brevity. Wheel speed errors were bounded with an RMSE of 7.905 rpm and a 95th-percentile error of 13.997 rpm, while wheel momentum errors had an RMSE of 2.99×10^{-5} Nms and a 95th-percentile error of 4.97×10^{-5} Nms, as reported in Table 9. These results validate the wheel-state propagation accuracy of SpaceAGORA.jl under realistic telemetry-driven inputs.

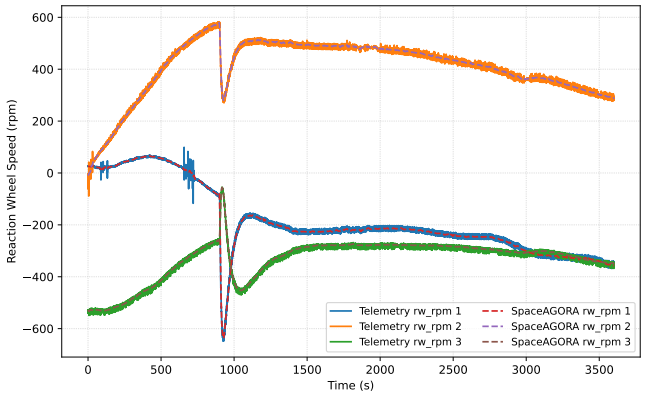


Figure 8. CYGNSS reaction-wheel speed replay comparison with SpaceAGORA.jl results.

7. Conclusion

This paper validates SpaceAGORA.jl as an open-source 6-DOF simulation environment for coupled orbit and rigid-body attitude propagation in autonomy and GNC workflows. Cross-validation against Basilisk confirms consistent numerical propagation and model implementation across diverse scenarios, including passive spacecraft dynamics and active detumbling maneuvers driven by reaction wheels and reaction control thrusters. Position errors relative to Basilisk remained within tens of meters, corresponding to relative errors less than 0.01%, while attitude RMSE ranges from 1e-4 to 3e-2 radians, depending on the scenario.

Propagation-level consistency was additionally assessed using University of Michigan SmallSat datasets from CYGNSS and GRIFEX. Over a 48-hour window, SpaceAGORA.jl remains within approximately kilometers (0.07% relative error) of a GRIFEX orbit reconstructed via SGP4 from public TLEs and within approximately kilometers (0.2% relative error) of CYGNSS navigation telemetry. A telemetry-driven inverse-dynamics replay of a representative CYGNSS slew reproduces the measured attitude evolution with RMSE 3.43×10^{-2} rad, demonstrating the suitability of SpaceAGORA.jl for telemetry replay and actuator-state propagation studies.

Future developments will extend SpaceAGORA.jl to higher-fidelity geometry and multibody configurations, multi-spacecraft scenarios, and additional validation cases (including atmospheric-flight applications), while continuing validation against flight and operations datasets.

Acknowledgments

The authors would like to thank Will Wells of the Southwest Research Institute and Dr. Jon Van Noord of the Space Physics Research Lab at the University of Michigan for providing invaluable data from the CYGNSS mission.

References

[1] G. Falcone and Z. R. Putnam, "Autonomous Decision-Making for Aerobraking via Parallel Randomized Deep Reinforcement Learning," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 3, pp. 3055–3070, Jun. 2023. [Online]. Available: https://ieeexplore.ieee.org/document/9946445/

[2] ——, "Energy Depletion Guidance for Aerobraking Atmospheric Passes," *Journal of Guidance, Control, and Dynamics*, vol. 45, no. 4, pp. 651–668, Apr. 2022. [Online]. Available: https://arc.aiaa.org/doi/10.2514/1.G006171

[3] K. S. Tracy, G. Falcone, and Z. Manchester, "Robust Entry Guidance with Atmospheric Adaptation," in *AIAA SCITECH 2023 Forum*. National Harbor, MD & Online: American Institute of Aeronautics and Astronautics, Jan. 2023. [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.2023-0301

[4] S. P. Hughes, "General Mission Analysis Tool (GMAT)," Aug. 2007, nTRS Author Affiliations: NASA Goddard Space Flight Center NTRS Document ID: 20080045879 NTRS Research Center: Goddard Space Flight Center (GSFC). [Online]. Available: https://ntrs.nasa.gov/citations/20080045879

[5] Analytic Graphics, "Systems Tool Kit," Exton, PA, 2025. [Online]. Available: https://www.ansys.com/products/missions/ansys-stk

[6] a.i. Solutions, "FreeFlyer," Lanham, MD, Sep. 2024. [Online]. Available: https://ai-solutions.com/freeflyer/

[7] P. W. Kenneally, S. Piggott, and H. Schaub, "Basilisk: A Flexible, Scalable and Modular Astrodynamics Simulation Framework," *Journal of Aerospace Information Systems*, vol. 17, no. 9, pp. 496–507, Sep. 2020. [Online]. Available: https://arc.aiaa.org/doi/10.2514/1.I010762

[8] E. D. Yu, N. Simha, and G. Falcone, "A Comparative Analysis of Aerobraking Dynamics Across Celestial Bodies Using SpaceAGORA.jl," Orlando, FL, Jan. 2026.

[9] C. S. Ruf, S. Gleason, Z. Jelenak, S. Katzberg, A. Ridley, R. Rose, J. Scherrer, and V. Zavorotny, "The CYGNSS nanosatellite constellation hurricane mission," in *2012 IEEE International Geoscience and Remote Sensing Symposium*. Munich, Germany: IEEE, Jul. 2012, pp. 214–216. [Online]. Available: http://ieeexplore.ieee.org/document/6351600/

[10] J. W. Cutler, C. Lacy, T. Rose, D. Rider, and C. Norton, "An update on the grifex mission."

[11] D. Sternberg and S. Mohan, "Validation of small satellite dynamics simulation modules using ASTERIA flight data," in *2021 IEEE aerospace conference (50100)*, 2021, pp. 1–14.

[12] D. C. Sternberg, C. Pong, N. Filipe, S. Mohan, S. Johnson, and L. Jones-Wilson, "Jet Propulsion Laboratory Small Satellite Dynamics Testbed Simulation: On-Orbit Performance Model Validation," *Journal of Spacecraft and Rockets*, vol. 55, no. 2, pp. 322–334, 2018, publisher: American Institute of Aeronautics and Astronautics _eprint: https://doi.org/10.2514/1.A33806. [Online]. Available: https://doi.org/10.2514/1.A33806

[13] P. Bangert, S. Busch, and K. Schilling, "Performance characteristics of the uwe-3 miniature attitude determination and control system," Mar. 2014.

[14] A. Slavinskis, H. Ehrpais, H. Kuuste, I. Sünter, J. Viru, J. Kütt, E. Kulu, and M. Noorma, "Flight results of estcube-1 attitude determination system," *Journal of Aerospace Engineering*, vol. 29, p. 04015014, Jan. 2016.

[15] C. Menegaldo, "Simulador de controle de atitude e propagação de órbita aplicado a nanossatélites em órbita baixa terrestre: desenvolvimento e validação com dados de voo do nanossatélite PicSat," *Master's thesis*, Jan. 2020. [Online]. Available: https://www.academia.edu/45570947/Attitude_control_simulator_and_orbit_propagation_applied_to_nanosatellites_in_low_Earth_orbit_development_and_validation_with_nanossatellite_PicSat_fligh_data_in_Portuguese_

[16] C. Tsitouras, "Runge–Kutta pairs of order 5(4) satisfying only the first column simplifying assumption," *Computers & Mathematics with Applications*, vol. 62, no. 2, pp. 770–775, Jul. 2011. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0898122111004706

[17] K. Hart, K. Simonis, B. Steinfeldt, and R. Braun, "Analytic Free-Molecular Aerodynamics for Rapid Propagation of Resident Space Objects," *Journal of Spacecraft and Rockets*, vol. 55, pp. 27–36, Sep. 2017.

[18] "An Overview of Reference Frames and Coordinate Systems," Apr. 2023. [Online]. Available: https://spiftp.esac.esa.int/workshops/2023_04_ESAC_BASIC_TRAINING/Tutorials/PDF/Individual_tutorials/15_frames_and_coordinate_systems.pdf

[19] F. L. Markley and J. L. Crassidis, *Fundamentals of Spacecraft Attitude Determination and Control*. New York, NY: Springer New York, 2014. [Online]. Available: http://link.springer.com/10.1007/978-1-4939-0802-8

[20] D. Vallado and W. McClain, *Fundamentals of Astrodynamics and Applications*, ser. Fundamentals of Astrodynamics and Applications. Microcosm Press, 2001. [Online]. Available: https://books.google.com/books?id=OCkGmwEACAAJ

[21] R. A. Hall, S. Hough, C. Orphee, and K. Clements, "Design and Stability of an On-Orbit Attitude Control System Using Reaction Control Thrusters," in *AIAA Guidance, Navigation, and Control Conference*. San Diego, California, USA: American Institute of Aeronautics and Astronautics, Jan. 2016. [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.2016-0087

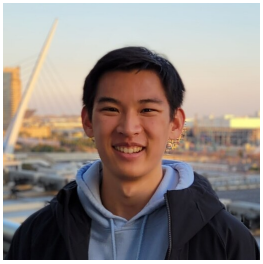
[22] J. C. Smith and J. L. Bell, "2001 Mars Odyssey Aerobraking," *Journal of Spacecraft and Rockets*, vol. 42, no. 3, pp. 406–415, May 2005. [Online]. Available: https://arc.aiaa.org/doi/10.2514/1.15213

[23] S. Damiani, J. M. Garcia, R. Guilanyà, P. Muñoz, and M. Müller, "FLIGHT DYNAMICS OPERATIONS FOR VENUS EXPRESS AEROBRAKING CAMPAIGN: A SUCCESSFUL END OF LIFE EXPERIMENT," Munich, Germany, Oct. 2015. [Online]. Available: https://issfd.org/2015/files/downloads/papers/097_Damiani.pdf

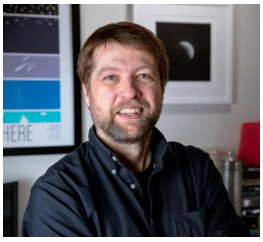
[24] K. Shoemake, "Animating rotation with quaternion curves," *SIGGRAPH Comput. Graph.*, vol. 19, no. 3, pp. 245–254, Jul. 1985. [Online]. Available: https://dl.acm.org/doi/10.1145/325165.325242

[25] S. Pines, "Uniform Representation of the Gravitational Potential and its Derivatives," *AIAA Journal*, vol. 11, no. 11, pp. 1508–1511, Nov. 1973. [Online]. Available: https://arc.aiaa.org/doi/10.2514/3.50619

BIOGRAPHY



Evan Yu received the B.S. degree in Aerospace Engineering from the University of Illinois Urbana-Champaign, Urbana, IL, USA, in 2024. He is currently pursuing the Ph.D. degree in Aerospace Engineering at the University of Michigan, Ann Arbor, MI, USA, in the Space-Flight Autonomous Leading CONcepts (Space-FALCON) Lab where he conducts research on atmospheric flight mechanics and guidance and control of hypersonic vehicles.



James Cutler received the B.S. degree in Computer and Electrical Engineering from Purdue University, West Lafayette, IN, USA, in 1996 and the M.S. and Ph.D. degrees from Stanford University in 1998 and 2005, respectively. He is currently a Professor in the Department of Aerospace Engineering at the University of Michigan and serves as the Principal Investigator of the Michigan eXploration Lab (MXL). His research focuses on space systems, communication, and robust computing.

Giusy Falcone received the B.S. and M.S. degrees in Aerospace Engineering from the University of Pisa, Italy, in 2014 and 2017, respectively, and the Ph.D. degree in Aerospace Engineering from the University of Illinois at Urbana-Champaign in 2022. She is currently an Assistant Professor in the Department of Aerospace Engineering at the University of Michigan and serves as the Principal Investigator of the Space-Flight Autonomous Leading CONcepts

(Space-FALCON) Lab. Her research focuses on guidance, control, and autonomous decision-making for hypersonic and space vehicles.