Comparing Backsubstitution and MuJoCo-Based Spacecraft Dynamics in Basilisk

Juan Garcia-Bonilla[1,*], **Andrew Morell**[1], **and Hanspeter Schaub**[1]

[1]*Ann and H. J. Smead Department of Aerospace Engineering Sciences, University of Colorado Boulder, 3775 Discovery Drive, Boulder, CO 80303, USA*
[*]*Corresponding author: juan@garciabonilla.com*

ABSTRACT

Spacecraft guidance, navigation, and control verification depends on dynamics simulations that are faithful and computationally practical. Basilisk provides two multibody formulations: the analytical Backsubstitution Method and an engine based on MuJoCo's recursive solver. A controlled comparison uses matched integrators and simulation infrastructure across rigid, articulated, variable-mass, and flexible spacecraft models with up to 70 degrees of freedom. Analytic Kepler and principal-axis torque solutions provide external checks. The matched orbit, wheel-and-panel, and segmented-panel cases agree at integration-truncation or floating-point scales. Controlled variants trace the remaining differences to component idealizations, gravity-force application points, and a velocity-dependent numerical artifact in MuJoCo's bias-force implementation. Runtime depends on workload: the BSM is faster in two specialized fixed-topology cases, MuJoCo is faster in two others, and the remaining two are within 10% of parity. MuJoCo scales more computationally favorably across the segmented-panel sweep. The matched scenarios now provide repeatable cross-formulation validation cases, and the comparison also guided performance optimizations. These outcomes show how two dynamics formulations in one toolkit can support both verification and continued development.

1 Introduction

Spacecraft guidance, navigation, and control (GNC) software is qualified largely against simulation. Test campaigns exercise flight algorithms in closed loop with a simulated vehicle, often before flight hardware exists. An unnoticed modeling gap or numerical artifact can therefore affect many later verification results. Mission teams address this risk by comparing simulation tools with analytic references and, where possible, with independently implemented software [1].

Several mature open-source astrodynamics frameworks serve this community, including ESA's GODOT, Orekit, GMAT, Tudat, NASA's 42, and Basilisk [1–6]. Basilisk now contains two dynamics formulations behind one scripting and messaging layer. Its native Backsubstitution Method (BSM) couples a six-degree-of-freedom (DOF) hub to attached components through analytically derived matrices [7–10]. The second engine wraps MuJoCo, an open-source physics engine developed for robotics, [11] behind the same Basilisk interfaces.

The formulations place modeling effort in different parts of the software. In the BSM, each attached component contributes derived coupling terms to a hub-centered system. In the MuJoCo engine, the user describes a body tree in MJCF XML and the recursive solver assembles its equations of motion. Both engines now cover several of the same spacecraft models, making a controlled comparison possible. To the authors' knowledge, no published study has compared their numerical agreement and runtime on matched spacecraft models.

The comparison covers five scenarios and several controlled variants. Sharing integrators, initial conditions, and Basilisk infrastructure removes many confounding implementation differences, but it also limits the independence of the comparison. Cross-engine agreement is therefore interpreted as evidence of consistency between the two formulations. Analytic Kepler and principal-axis torque solutions provide independent references for the translational and rotational rigid-body cases. The remaining experiments use control variants to identify conditions under which the formulations cease to represent the same physical model.

The contributions are:

1) a controlled scenario suite that quantifies cross-formulation agreement and uses analytic references where they are available;
2) a runtime scaling study from 8 to 70 system DOF, together with work-precision sweeps for fixed-step and adaptive integrators;
3) an account of the gravity, variable-mass, and floating-point assumptions that bound agreement between the formulations; and
4) an executable scenario suite whose exact revision and outputs are archived with the paper.

2 Related work

Surveys of multibody dynamics engines exist in the broader simulation literature, but comparisons of end-to-end spacecraft simulation tools are scarce. Many spacecraft simulators are proprietary. The open-source tools available for comparison also differ in language, coordinate conventions, solvers, integrators, and parallelization, making it difficult to isolate the source of a discrepancy.

Prior cross-tool exercises in astrodynamics, such as the verification campaign behind GMAT, compare orbit-propagation accuracy across independent flight-dynamics tools [1]. They do not benchmark flexible-spacecraft dynamics or runtime. Simulation environments for GNC development, [12,13] DARTS/Dshell, [14] and scalable multi-satellite simulation studies [15] each document one dynamics formulation. Basilisk's two engines instead share an API and much of the surrounding simulation infrastructure, allowing this study to control those differences at the cost of a less independent comparison.

The robotics community has benchmarked its physics engines more extensively. Erez, Tassa, and Todorov compared Bullet, Havok, MuJoCo, ODE, and PhysX across model classes and integration settings, finding that each engine performs best in the domain for which it was designed [16]. An engine's numerical assumptions reflect that domain, and a new operating regime can expose previously inconsequential behavior. Gazebo-based manipulation benchmarking [17] and Simscape-based spacecraft simulator design [18] similarly evaluate one engine in one domain. Comparing a robotics engine with an astrodynamics-native formulation under a common simulation framework is, to the authors' knowledge, new.

Beyond quantitative comparisons, Ivaldi et al. surveyed user experience with dynamics simulation tools, MuJoCo among them [19]. Respondents weighted "simulation very close to reality" (26%), open-source availability (22%), using the same code for the real and simulated robot (17%), and being light and fast (17%) as the most important criteria, ahead of customization (11%) and non-interpenetration (4%); percentages as reported in the survey. Although that survey targets robotics, these priorities motivate the present focus on numerical agreement and runtime within an open-source simulation environment.

For verification anchors, the NASA Engineering and Safety Center published check cases for 6-DOF flight-vehicle simulations [**?**, 20]. The orbit and torque scenarios below are deliberately of that family: an analytic Kepler orbit and a constant-torque rigid body with an exact principal-axis control and translation-invariance checks.

3 Dynamics formulations

The two formulations arose from different needs within Basilisk. The BSM was developed for efficient, fully coupled simulation of the common spacecraft architecture in which a central bus carries appendages, actuators, and internal moving masses. The MuJoCo engine was added to represent articulated systems through a general body-tree solver while retaining Basilisk's environment models, task scheduling, messaging, and GNC modules. The formulations overlap on the models studied here, but they begin from different representations and place the work of extending a model in different parts of the software. Only the details needed to interpret the comparisons are summarized here; the cited references provide full derivations.

3.1 Backsubstitution Method

The BSM organizes the spacecraft around one privileged 6-DOF rigid hub. Its design preserves full coupling between the hub and attached components (referred to as "effectors" in the BSM) without solving one global system whose dimension contains every component coordinate. Each effector instead derives its acceleration in terms of the hub's translational and rotational accelerations. Substituting those relations into the hub equations leaves a constant-size hub solve, after which the component accelerations are recovered by substitution [7, 9, 10]. This structure allows established component models to be combined without deriving the equations of motion for the complete vehicle again.

Attached components contribute only the states needed by their specialized effectors. For example, the balanced-wheel effector advances wheel spin rate but omits spin angle because that angle is cyclic for an axisymmetric wheel. Hinges and slosh masses retain their minimal generalized coordinates and rates. Each effector supplies analytically derived matrices and vectors through a common interface. Published derivations for representative effectors provide a basis for conservation and limiting-case tests.

The hub-centered architecture also limits the articulated topologies that the BSM can represent. A specialized component may contain a serial chain, and load-generating hardware such as a thruster may be carried by a moving body in that chain. A body in the chain cannot, however, serve as the parent of another articulated component with independent generalized coordinates as formalized by the recent BSM survey [21]. The BSM therefore cannot represent a branched articulated mechanism below the hub; a multi-finger gripper at the end of a robotic arm is one example. This is a structural limitation of the formulation rather than a missing component model. Contact dynamics are also absent from the current BSM implementation.

The cost structure follows from the same design. The hub solve is constant-size, but the coupling work grows with the number of effector coordinates, and for chained appendages (an N-segment panel, where each segment couples to its neighbors through the chain) the per-step work grows faster than linearly in N. The scaling measurement in the results section quantifies this.

3.2 MuJoCo-based engine

The MuJoCo engine addresses a broader class of articulated configurations. A modeler can branch the body tree at any body and add joint constraints and contact geometry without deriving new spacecraft coupling equations for each topology. MuJoCo was developed as a general-purpose engine for model-based control and uses recursive Newton-Euler and composite rigid-body algorithms [11, 22]. Its model is a tree of bodies connected by joints in minimal coordinates, and it computes forward dynamics recursively over that tree. There is no privileged hub; a spacecraft is represented as a free-floating body tree described in MJCF XML.

Basilisk uses MuJoCo as a multibody backend rather than as a separate simulation environment. The `MJScene` class invokes MuJoCo's forward dynamics, while Basilisk retains ownership of the numerical integrator, task scheduling, messaging, environment models, and GNC modules. Existing flight-software and environment components can therefore interact with either dynamics engine through the same framework interfaces. For the orbital comparisons, gravity is evaluated through the `NBodyGravity` module, once at each MuJoCo body center, instead of MuJoCo's uniform-gravity option. This captures differential forces between lumped body centers (but not gravity variation within an individual rigid body).

3.3 Consequences for model construction

The hub-and-effector and body-tree formulations differ in both development effort and model scope. When a BSM effector already exists, the modeler instantiates it and supplies its physical parameters; its coupling equations and implementation are part of the library. General effectors for sequential rotating or translating joints reduce how often a new derivation is needed, but a component outside those formulations still requires analytically deriving BSM contributions, a C++ implementation, and tests of conservation and limiting cases [9, 23]. The BSM's articulated coordinates remain limited to components coupled at the hub and serial chains defined within those components; a moving body in a chain cannot parent another articulated subsystem.

MuJoCo assembles the equations from bodies, joints, actuators, and constraints described in MJCF. A new topology built from these elements generally changes the model description rather than the dynamics solver. This shifts work from deriving component-specific coupling equations to constructing and checking the body tree; it does not remove the need to specify the physical assumptions. The segmented-panel scenario later illustrates the two construction paths on the same spacecraft model. MuJoCo also provides contact and constraint machinery beyond the scope of the present comparison; contact-rich spacecraft applications are studied elsewhere [24].

4 Comparison methodology

Numerical consistency is assessed against analytic results where they exist and against the other engine elsewhere. Each accuracy comparison uses the same integrator, step size, simulated duration, and recording cadence. Timing runs use separately built simulations with logging disabled, and model construction and initialization occur outside the timed interval. Each configuration receives one discarded warm-up followed by five propagation trials, with the implementation order reversed on alternate rounds. The tables report the median for each implementation.

Three controls make cross-engine differences meaningful. Both engines are stepped by the same Basilisk integrator at the same step size; initial conditions are computed once and handed to both engines; and both use the same central gravity parameter and inverse-square point-gravity law. The extended-body variant deliberately changes the points at which that law is applied. Because the BSM represents attitude with Modified Rodrigues Parameters (MRPs) and MuJoCo with a quaternion, attitude histories are compared through the principal rotation angle Φ of the relative direction-cosine matrix, evaluated as:

$$\Phi = 4\arctan\|\boldsymbol{\sigma}_{\text{rel}}\| = 2\arctan\frac{\|\boldsymbol{\beta}_{1:3}\|}{\beta_0} \tag{1}$$

where $\boldsymbol{\sigma}_{\text{rel}}$ is the relative MRP set, β_0 the scalar quaternion parameter, and $\boldsymbol{\beta}_{1:3}$ the vector part. This metric is parameterization-invariant and remains well conditioned down to machine precision.

The suite combines five scenarios, one extended-body variant, and two integrator studies.* Some cases introduce more than one change, so attribution relies on the controlled variants described with each result. Every script emits machine-readable JSON and its figures. The results were regenerated on a MacBook Pro (Mac16,7) with an Apple M4 Pro processor (14 cores) and 48 GB of memory, running macOS 26.6.2, Python 3.11.15, and MuJoCo 3.11.0. The Basilisk binaries were compiled in Release mode with AppleClang 17.0.0 and C++17. The completed scenarios are available in the Basilisk repository at commit `1e9a4dfa9767`. †

5 Experimental setup and scenarios

The benchmark suite begins with isolated translational and rotational motion, then adds coupled internal motion, variable mass, and larger articulated structures. Controlled variants separate numerical behavior from differences in physical assumptions.

Translational orbit. A 750 kg point mass on a Keplerian orbit ($a = 7000$ km, $e = 0.01$) about point-mass Earth, propagated for 11,640 s (1.997 orbits) with RK4 at $\Delta t = 10$ s. The BSM uses its translational-only point-mass mode, while the MuJoCo body has three orthogonal slide joints and no rotational coordinates. The analytic Kepler solution exists, so this scenario anchors both engines to truth while exercising only the translational equations and the gravity plumbing.

Constant body torque. A six-degree-of-freedom rigid body with principal inertias 900, 800, and 600 kg m^2 is driven for 600 s by the body-frame torque $(0.2, -0.3, 0.4)$ N m. Each engine runs once in orbit and once at rest. Translation cannot torque a body about its center of mass, so this pair isolates velocity-dependent attitude behavior. A separate at-rest control aligns the initial rate and torque with the body z axis, for which the angular rate and attitude have closed-form solutions.

Hub with three reaction wheels and two hinged panels. This 11-degree-of-freedom model has a free-floating 600 kg hub, three orthogonal wheels under constant motor torques, and two spring-damper hinged panels released from a 12° deflection. It tests internal momentum exchange and flexible-appendage coupling.

Extended-body variant. The same multi-body spacecraft is placed in orbit and run in compact (hub and wheels only) and extended (with offset panels) configurations. The variant isolates system-center-of-mass gravity in the BSM from per-body gravity in MuJoCo.

*`https://avslab.github.io/basilisk/develop/examples/index.html#comparing-the-back-substitution-and-mujoco-dynamics-engines`

†`https://github.com/AVSLab/basilisk/commit/1e9a4dfa97672a5369fdeb922c23064fcd97391d`

Fuel slosh and variable mass. A 15-minute orbit-raising burn of a MOOG Monarc-445 thruster (445 N, $I_{sp} = 234$ s) drawing from a spherical tank holding 1500 kg of propellant in a 3000 kg wet vehicle. The slosh model comprises one spherical pendulum parameterized from Dodge [25] and three spring-mass-damper particles; all four masses deplete continuously. Both engines apply the same thrust and prescribed depletion history while updating the retained masses and inertias. Forces and torques associated with propellant transport through the tank and feed system are omitted; the nozzle thrust is applied explicitly.

***N*-segment lumped-mass panels.** Each of two solar arrays is discretized into a serial chain of N rigid segments joined by torsional spring-damper hinges. The scenario evaluates the same chain with two BSM effectors and an equivalent MuJoCo body tree. The specialized `nHingedRigidBody` formulation follows the lumped rigid-body approximation developed for first-order solar-array flexing [26]. The general `spinningBodyNDOF` formulation admits arbitrary chain length, hinge axes, and link mass properties [23]. The general formulation is restricted to the same parallel-axis panel geometry as the specialized model. The sweep from $N = 1$ to 32 spans 8 to 70 physical system degrees of freedom; the 16-segment case has 38.

Work–precision studies. A work–precision curve compares computational work, measured here by propagation time, with numerical error relative to an engine-specific reference. The wheel-and-panel study sweeps fixed-step Euler, RK2, and RK4 and adaptive RKF45 and RKF78. The stiff 16-segment array omits Euler, which is unstable over the useful step-size range, and compares RK2, RK4, RKF45, and RKF78.

6 Numerical results

The orbit and wheel-and-panel cases establish the error levels obtained when both engines represent the same physical model. The torque, extended-gravity, and variable-mass controls then distinguish numerical artifacts from differences in physical assumptions. Runtime is considered only after these agreement checks.

6.1 Point-mass orbit comparison

The point-mass orbit provides two distinct checks. Comparison with the analytic Kepler solution measures integration error. Comparison between the engines measures the residual introduced by their different force and acceleration calculations. If both engines implement the same translational equations, the first error should decrease at fourth order under RK4 refinement, while the second need not follow the step size once it reaches floating-point scale.

At the baseline step of $\Delta t = 10$ s, both engines remain within 3.05 cm of the analytic position over 11,640 s. Their relative specific-energy drift stays below 5.1×10^{-11}. In Fig. 1a, decreasing the step from 40 to 2.5 s over the common 11,520 s sweep horizon reduces the maximum position error from 10.22 m to 1.09×10^{-4} m. The fitted orders between adjacent steps are 4.25, 4.14, 4.08, and 4.05, which is the expected RK4 convergence.

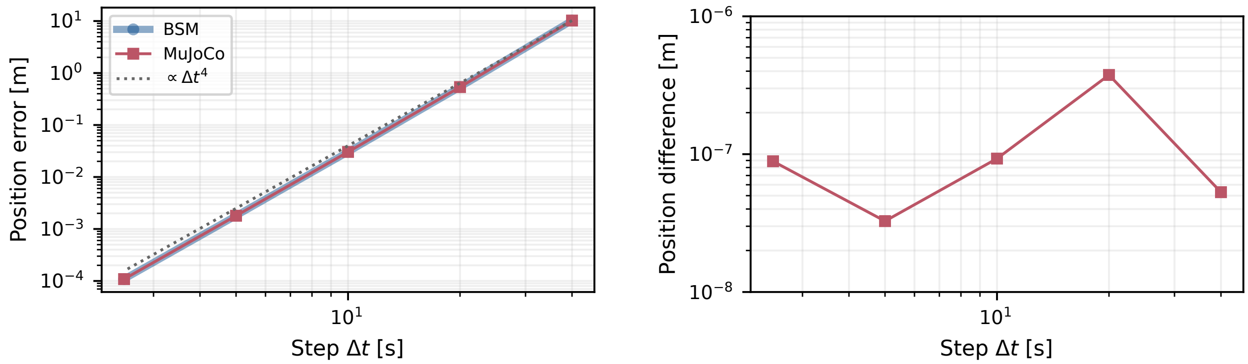
The cross-engine residual follows a different pattern. It is 9.8×10^{-8} m at the baseline step and remains below 4×10^{-7} m throughout the sweep, without a monotonic dependence on Δt (Fig. 1b). Thus step refinement improves both trajectories together but does not systematically reduce the distance between them. The observed residual is consistent with differences at floating-point scale in the force and acceleration evaluations; it does not indicate different continuous orbit equations.

6.2 Torqued rigid-body controls

The torque scenario adds rotational dynamics to the orbit comparison. A single rigid body begins on a Keplerian orbit and rotates under a constant body-frame torque. Both engines use the same mass properties, initial attitude, body rate, and applied torque. Over the 600 s baseline propagation, their attitude histories differ by at most 1.38×10^{-4} degrees, and their body rates differ by at most 4.7×10^{-8} rad/s.

The torque response is first tested against an independent solution. In a separate at-rest control, the initial rate and applied torque lie along the body z principal axis: $\omega_z(0) = 0.03$ rad/s and $L_z = 0.4$ N m. The gyroscopic cross term then vanishes, and the exact motion is

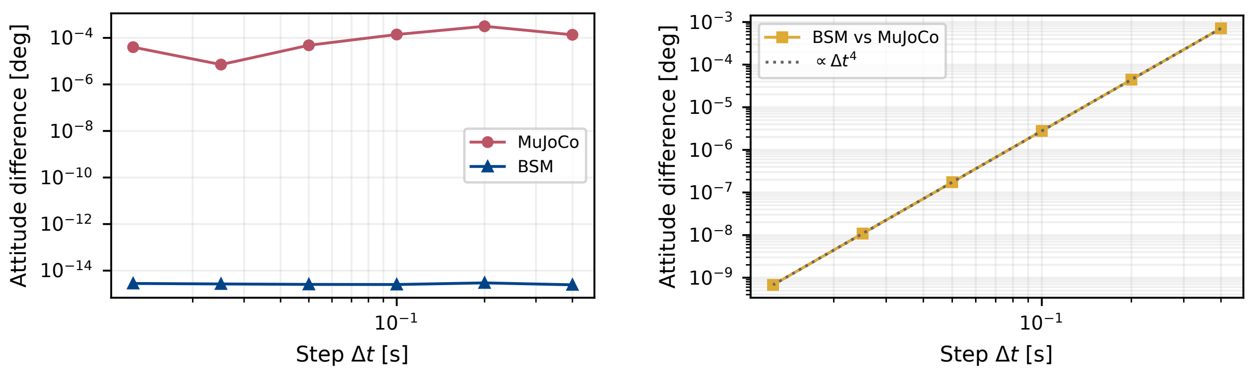
$$\omega_z(t) = \omega_z(0) + \frac{L_z}{I_z} t, \qquad \theta_z(t) = \omega_z(0) t + \frac{L_z}{2 I_z} t^2 \tag{2}$$



(a) *Error against the analytic Kepler orbit. The BSM and MuJoCo curves overlap and follow the fourth-order reference slope.*

(b) *Cross-engine position difference. The residual stays below* 4×10^{-7} *m and does not decrease systematically with the step.*

Figure 1 *The orbit sweep separates common RK4 truncation from the much smaller cross-engine residual. Each panel uses a separate vertical scale.*



(a) *Orbiting-versus-rest difference within each engine. The MuJoCo residual does not decrease systematically with* Δt*.*

(b) *Cross-engine difference at rest. The residual follows the fourth-order reference slope.*

Figure 2 *Step refinement of the torque controls. The orbiting-versus-rest discrepancy persists only in MuJoCo, whereas the at-rest cross-engine residual decreases at fourth order.*

with $I_z = 600\,\mathrm{kg\,m^2}$. Over 600 s, both engines remain within 2.77×10^{-14} rad/s of the exact rate. The maximum attitude errors are 2.94×10^{-7} degrees for the BSM and 1.77×10^{-6} degrees for MuJoCo. Both formulations therefore reproduce the analytically solvable torque response at the baseline step.

The direct comparison leaves two broad possibilities: ordinary integration error or an effect associated with the orbiting formulation. A force through the center of mass changes the body's translation but cannot change its rotation about the center of mass. Repeating each propagation at rest, with the same rotational initial conditions and body-frame torque, tests this invariance. The orbiting and at-rest attitude histories within each engine should be identical.

At the baseline step, the BSM orbiting and at-rest attitudes differ by at most 2.5×10^{-15} degrees. The corresponding MuJoCo difference reaches 1.38×10^{-4} degrees. Refining Δt from 0.4 to 0.0125 s does not reduce the MuJoCo value monotonically; it ranges from 6.9×10^{-6} to 3.2×10^{-4} degrees (Fig. 2a). The absence of systematic convergence distinguishes this result from ordinary integration truncation.

In the at-rest case, the cross-engine attitude difference decreases from 6.9×10^{-4} to 6.9×10^{-10} degrees and follows the fourth-order reference slope (Fig. 2b). The two engines therefore approach the same rotational solution when the large translational state is absent.

The orbiting and at-rest cases differ in both gravity and translational velocity. To separate those changes, a second control removes gravity and varies only the inertial drift speed. The exact rotational history remains the at-rest history. The BSM difference stays at 2.5×10^{-15} degrees for every tested speed. The MuJoCo difference grows from 3.0×10^{-15} degrees at rest to 6.9×10^{-5} degrees at 0.75 km/s and 3.6×10^{-3} degrees at 7.5 km/s (Fig. 3).

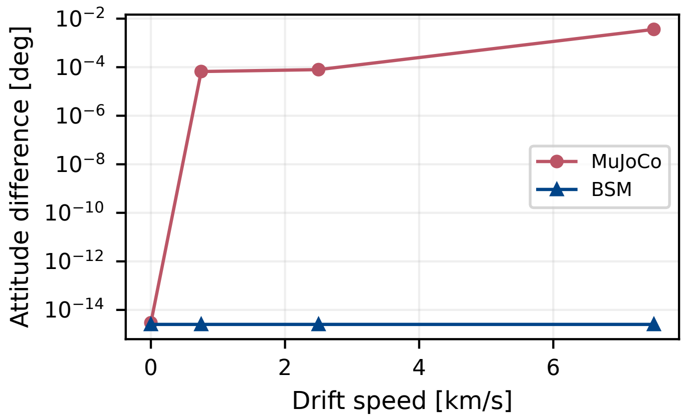


Figure 3 *Attitude difference from an otherwise irrelevant inertial drift velocity, with gravity removed. The BSM stays at floating-point scale. The MuJoCo difference grows with speed and reaches* 3.6×10^{-3} *degrees at 7.5 km/s.*

This dependence on speed identifies a velocity-dependent numerical effect rather than a difference in the gravity model.

The speed dependence points to MuJoCo's velocity-dependent bias-force calculation. In MuJoCo 3.11.0, the recursive Newton-Euler calculation forms the spatial bias force as $\boldsymbol{v} \times^* ([I]\boldsymbol{v})$. For pure translation about the body center of mass, the rotational part contains $\boldsymbol{v}_{\text{lin}} \times (m\boldsymbol{v}_{\text{lin}})$. This term is zero in exact arithmetic, but its implementation subtracts products of order $m\|\boldsymbol{v}_{\text{lin}}\|^2$, making floating-point cancellation a plausible source of the residual.

The generalized bias force was recorded directly from the linked binary for a 7.5 km/s drift. With direction $[2, -3, 6]/7$, the initial rotational bias is

$$[-3.11 \times 10^{-7},\ 1.56 \times 10^{-6},\ 7.82 \times 10^{-7}]\,\text{N m},$$

and its translational entries are zero. The same test along a coordinate axis gives exactly zero bias and zero attitude drift over 600 s. These measurements connect the propagation error to the nominally zero translational self-term.

Two final propagation controls test the expected cancellation-error scale without the applied torque or initial body rate. Doubling the drift speed successively from 0.9375 to 7.5 km/s multiplies the 600 s attitude error by 4.0000, 4.0001, and 4.0003. Doubling the body mass from 750 to 1500 kg, with inertia held fixed, multiplies the error by 2.0002. The direct bias measurement, exact axis cancellation, and $m\|\boldsymbol{v}_{\text{lin}}\|^2$ scaling strongly support floating-point cancellation as the source of the isolated single-body drift.

6.3 Reaction-wheel and hinged-panel comparison

This scenario compares coupled internal motion in a spacecraft with three driven reaction wheels and two hinged panels. Gravity and variable mass are omitted, which isolates the hub, wheel, and panel dynamics.

The hub attitude difference reaches 6.3×10^{-10} degrees over 120 s (Fig. 4). The hub and wheel-rate differences remain below 5.3×10^{-13} rad/s, and the panel-angle difference remains below 4.6×10^{-8} degrees. Agreement in these outputs tests momentum exchange with the wheels and spring-damper coupling through the panels, rather than attitude alone.

Most of the attitude difference comes from the BSM's massless-wheel idealization. Each MuJoCo wheel is assigned 10^{-6} kg. Reducing that mass to 10^{-8} kg lowers the maximum attitude difference from 6.3×10^{-10} to 7.5×10^{-12} degrees. At 10^{-10} kg, the difference is 2.4×10^{-12} degrees. This mass difference therefore sets the scale of the baseline residual.

6.4 Extended multibody orbit comparison

The orbiting multibody case compares the engines' point-gravity implementations. The BSM applies one force at the total system center of mass, whereas the MuJoCo wrapper applies a separate force at each lumped body center.

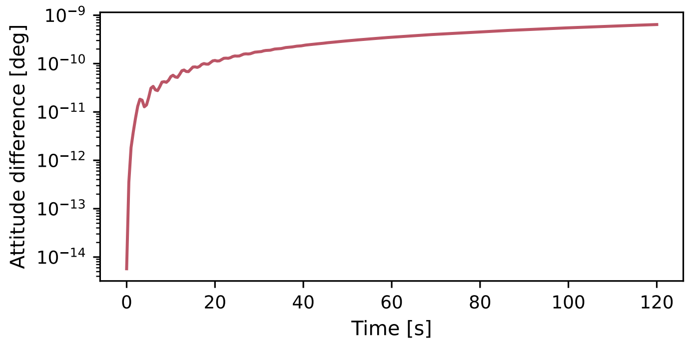
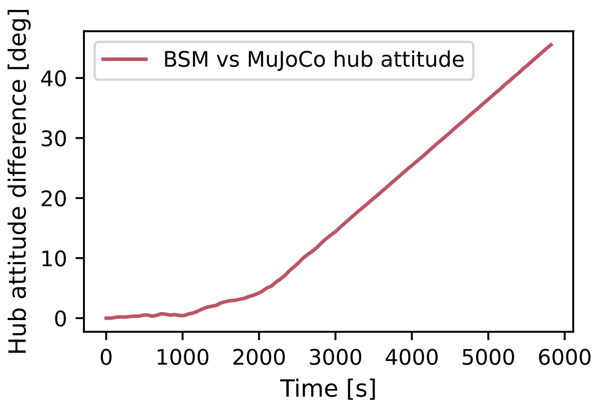
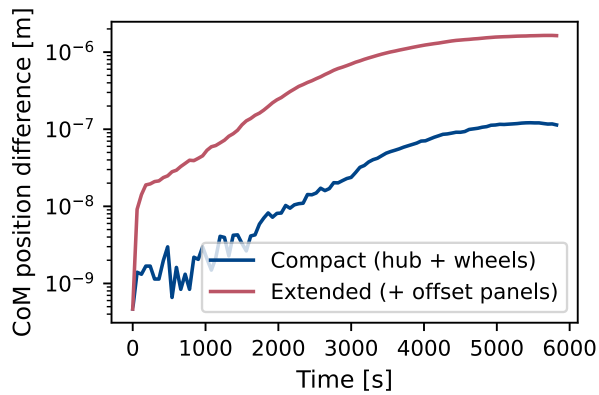


Figure 4 *Hub attitude difference for the matched wheel-and-panel model.*



(a) *Extended-model hub attitude difference. Separated point forces produce a 45.5° rotational response.*



(b) *Total-CoM position difference. Both compact and extended models remain below 2×10^{-6} m.*

Figure 5 *Per-body point gravity affects the extended vehicle's attitude, while the total center-of-mass orbit remains closely matched.*

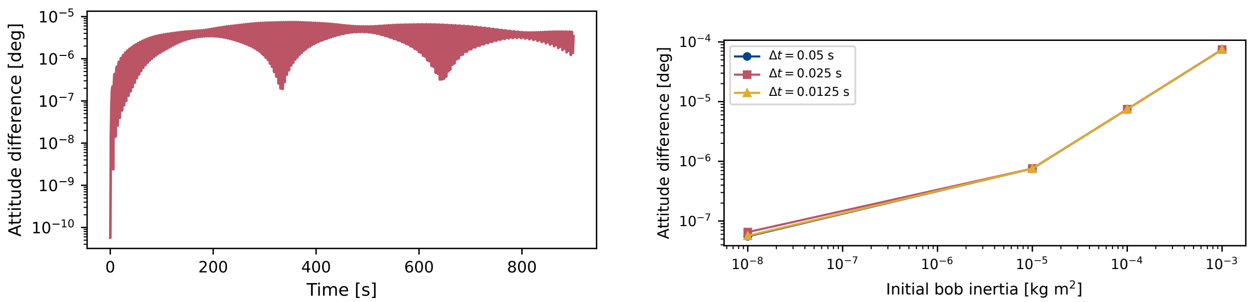
The compact configuration contains only the hub and co-located wheels, so these application points coincide. Its hub attitudes differ by 5.3×10^{-3} degrees after one orbit, which is of the same order as the velocity-dependent MuJoCo residual at orbital speed.

Adding offset panels separates the gravity application points. In this extended configuration, the attitude difference grows to 45.5° (Fig. 5a). The forces applied at the individual body centers can exert a net torque about the total center of mass. MuJoCo's per-body application captures this gravity-gradient torque implicitly, whereas the BSM's single-point application requires it as a separate dynamic effector, which this scenario did not add. An estimate based on the initial separation of the body centers gives an angular-acceleration scale of 2.0×10^{-6} rad/s^2. The flexible, tumbling vehicle does not follow a constant-acceleration trajectory, so this estimate characterizes the forcing scale rather than fitting the attitude history.

Despite the large attitude difference, the total center-of-mass trajectories remain close. Over one orbit, the compact models differ by at most 1.3×10^{-7} m and the extended models by 1.7×10^{-6} m (Fig. 5b). The individual MuJoCo forces have different moment arms and produce a net torque, but their vector sum remains close to the single BSM force at the total center of mass. The rotational motion therefore diverges while the translational motion remains closely matched. MuJoCo resolves gravity variation between body centers; neither model resolves variation across the finite extent of an individual rigid body.

6.5 Variable-mass and slosh comparison

The variable-mass scenario compares the engines during a 900 s burn by a 445 N thruster. A spherical tank supplies the propellant, which is divided among a bulk component, one pendulum mode, and three translational slosh masses. All components follow the same prescribed fractional depletion rate, changing the vehicle mass



(a) *Principal-angle attitude difference at* $\Delta t = 0.05$ s *and an initial bob inertia of* 10^{-4} kg m^2.

(b) *Maximum hub-attitude difference as the initial MuJoCo bob inertia and integration step vary.*

Figure 6 *Deep-space variable-mass and slosh comparison. The plotted inertias are initial values and decrease with the bob mass during depletion. The maximum hub-attitude difference depends strongly on bob inertia and shows little sensitivity to the integration step.*

and inertia throughout the burn. The comparison is performed in deep space so that the difference in gravity application does not enter the result. Both models apply the same thrust and depletion history. This leaves the variable mass and slosh representations as the quantities under comparison.

Both engines represent the pendulum with two angular coordinates and equivalent viscous damping. The BSM idealizes the bob as a point mass, whereas MuJoCo represents it as a finite rigid body with centroidal inertia. The baseline MuJoCo model assigns an initial inertia of 10^{-4} kg m^2, which decreases with the bob mass.

The full deep-space models differ by at most 7.46×10^{-6} degrees in attitude and 9.94×10^{-8} rad/s in body rate. The translational slosh coordinates differ by 1.14 μm, and the pendulum coordinates by 1.82×10^{-4} degrees. The prescribed mass histories agree within 9.1×10^{-13} kg, and the total center-of-mass trajectories remain within 19 μm after approximately 134 m/s of accumulated burn velocity.

Fig. 6b varies the initial MuJoCo bob inertia and integration step. At $\Delta t = 0.05$ s, decreasing the inertia from 10^{-3} to 10^{-4}, 10^{-5}, and 10^{-8} kg m^2 lowers the maximum attitude difference from 7.45×10^{-5} to 7.46×10^{-6}, 7.57×10^{-7}, and 5.54×10^{-8} degrees. Half and quarter steps give essentially the same values. The attitude difference decreases strongly with bob inertia and changes little under the tested step refinements; at the baseline inertia, bob inertia dominates the observed rotational difference. A near-rigid control suppresses the pendulum motion and reduces the maximum attitude difference to 3.70×10^{-9} degrees, checking the depletion treatment independently of the pendulum response.

6.6 Propagation cost

The preceding agreement analysis establishes which timing cases compare identical physical models and which retain the documented gravity or inertial-property differences. Table 1 compares propagation cost for five fixed-topology cases and the variable-mass workload.

The timing simulations disable logging. Model construction and initialization are excluded, and each value is the median of five interleaved trials after one discarded warm-up. Within each row, both engines use the same integrator, step size, and simulated duration. The fixed-topology results do not give either engine a uniform advantage. The BSM takes 0.45× the MuJoCo time in the translational orbit, but this result uses a specialized translational-only path and is not a general rigid-body comparison.

The rigid and articulated cases show that DOF count alone does not determine runtime. The orbiting torqued rigid-body timings are within 5%, with a BSM/MuJoCo ratio of 0.97, so this case is near parity on the test machine. The compact orbital model adds three reaction-wheel DOF and takes 0.71× the MuJoCo time. Both engines represent the same nine mechanical DOF: six for the free hub and one spin DOF per wheel. Their numerical representations differ, however. The specialized BSM effector advances only the three wheel rates because the spin angles are cyclic. MuJoCo represents each wheel as a hinged child body, advances both joint angle and rate, and includes the wheel in its body-tree calculations. This additional work may explain the compact result, although the present timings do not isolate its contribution.

Adding two panel hinges raises the BSM time by 77% and the MuJoCo time by 35%, changing the BSM/MuJoCo

Scenario	BSM [s]	MuJoCo [s]	BSM/MuJoCo
Two-body orbit (11,640 s, $\Delta t = 10$ s)	0.0032	0.0073	0.45
Orbiting torqued rigid body (600 s, $\Delta t = 0.1$ s)	0.0441	0.0455	0.97
Hub + 3 RWs + 2 panels, no gravity (120 s, $\Delta t = 0.02$ s)	0.1018	0.0595	1.71
Orbiting hub + 3 RWs, compact (1 orbit, $\Delta t = 0.1$ s)	0.5391	0.7574	0.71
Orbiting hub + 3 RWs + 2 panels, extended (1 orbit, $\Delta t = 0.1$ s)	0.9532	1.0206	0.93
Variable mass + slosh, orbit (900 s, $\Delta t = 0.05$ s)	0.6069	0.5233	1.16

Table 1 *Propagation time with logging disabled. Values are medians of five interleaved trials after a discarded warm-up; model setup is excluded. A BSM/MuJoCo ratio below one indicates that the BSM is faster.*

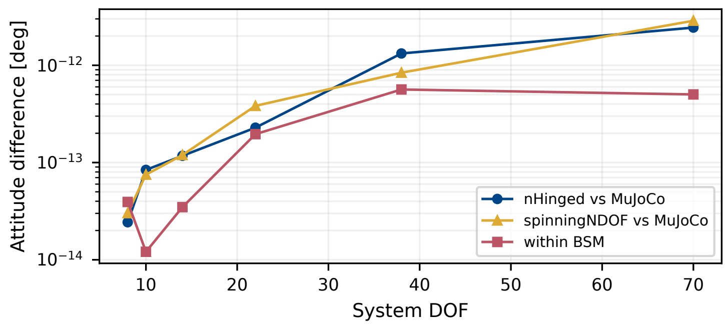


Figure 7 *Maximum attitude difference over the segmented-panel sweep. All three pairwise comparisons remain at or below 2.9×10^{-12} degrees from 8- to 70-DOF.*

ratio from 0.71 to 0.93. The extended orbital case is therefore also near parity, although MuJoCo adds the panel bodies at lower marginal cost. MuJoCo is faster in the no-gravity wheel-and-panel case and the orbital variable-mass case, with BSM/MuJoCo ratios of 1.71 and 1.16.

6.7 Segmented-panel comparison and scaling

The segmented-panel sweep varies model size while holding the total array mass and length and the per-hinge laws fixed. Each solar array contains N rigid segments. Two BSM effectors are compared with the equivalent MuJoCo body tree, from $N = 1$ (8-DOF) to $N = 32$ (70-DOF).

The specialized hinged-panel BSM series uses `nHingedRigidBody`; the general sequential-rotation series uses `spinningBodyNDOF`. Both remain numerically consistent with MuJoCo throughout the sweep. The maximum cross-engine attitude difference is 2.9×10^{-12} degrees, and the two BSM effectors differ by no more than 5.7×10^{-13} degrees (Fig. 7). This agreement lets the timing study be interpreted as a cost comparison for the same dynamics.

Each scaling point is the median propagation time for a fixed budget of 600 integrator steps. MuJoCo is cheaper at every measured size. The `nHingedRigidBody`/MuJoCo ratio is 2.38 at 8-DOF, decreases to 1.88 at 22-DOF, and rises to 4.43 at 70-DOF. The advantage is therefore largest for the largest measured model, although it does not grow monotonically across the sweep (Fig. 8).

The two BSM effectors remain close through $N = 4$. Beyond that point, `spinningBodyNDOF` grows more quickly and costs 2.17 times as much as `nHingedRigidBody` at $N = 32$. Over the largest measured interval, from 38- to 70-DOF, the local cost exponents are 2.32 for `nHingedRigidBody`, 2.70 for `spinningBodyNDOF`, and 1.35 for MuJoCo. Writing the local relation as $T \propto D^p$, the exponent p is the slope of $\log T$ against $\log D$. The BSM costs therefore rise faster than quadratically over this interval, whereas MuJoCo remains between linear and quadratic growth.

The difference is consistent with the work performed during each dynamics evaluation. Both BSM effectors assemble and explicitly invert a dense mass matrix in the panel coordinates. Forming this matrix accounts for the coupling of each joint to its downstream segments. MuJoCo instead forms the joint-space inertia through a composite rigid-body recursion and uses a tree-structured sparse factorization to solve for the accelerations [11]. The measured slopes therefore indicate a lower marginal cost for MuJoCo over this interval. They describe the shape of the measured curves, not asymptotic complexity: each uses two timing points, and the horizontal axis includes the six fixed hub DOF.

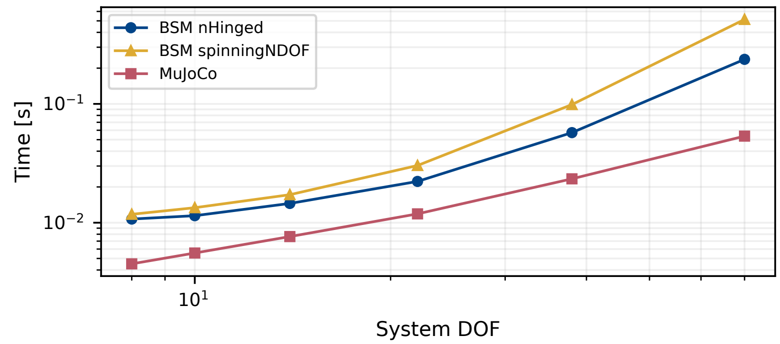
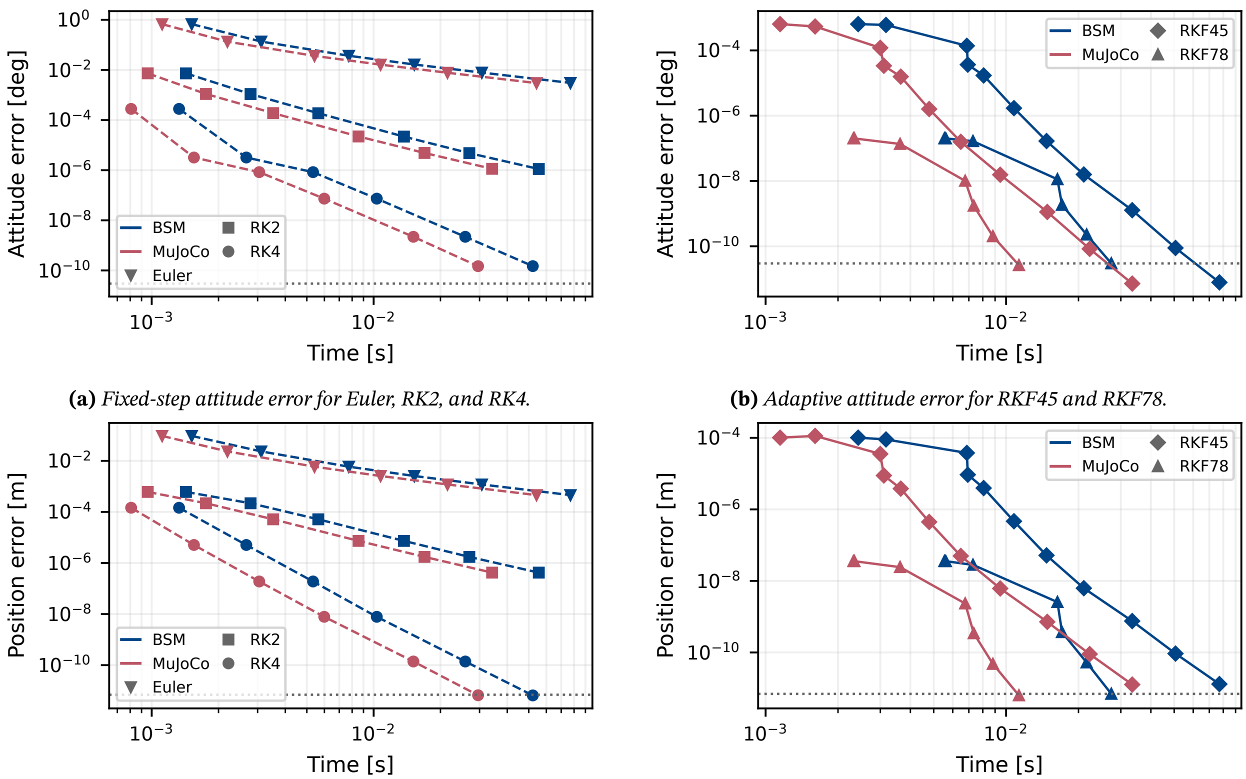


Figure 8 *Median propagation time for 600 steps versus system DOF. MuJoCo remains below both BSM curves. Its advantage is largest for the 70-DOF model but is not monotonic over the full sweep.*



(a) *Fixed-step attitude error for Euler, RK2, and RK4.*

(b) *Adaptive attitude error for RKF45 and RKF78.*

(c) *Fixed-step position error for the same runs as panel (a).*

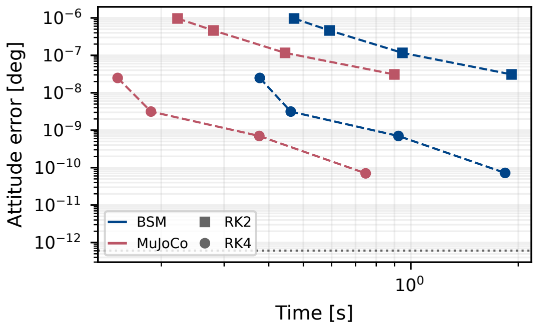
(d) *Adaptive position error for the same runs as panel (b).*

Figure 9 *Work-precision results for the wheel-and-panel spacecraft. Within each integrator family, the two engines follow similar error trends, while the MuJoCo points generally lie farther left.*

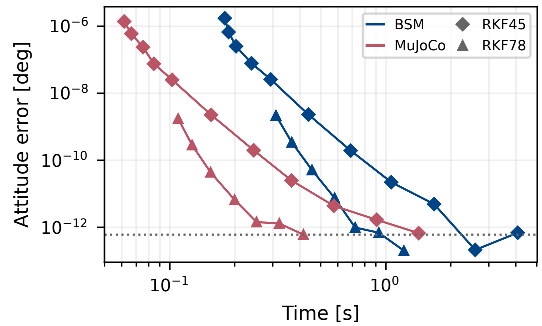
6.8 Work-precision analysis

The baseline table uses RK4, so it cannot establish whether the runtime ranking depends on that integrator. The work-precision studies repeat the wheel-and-panel and 16-segment comparisons with fixed-step and adaptive Runge–Kutta methods. Error is measured against a tight reference generated separately by each engine.

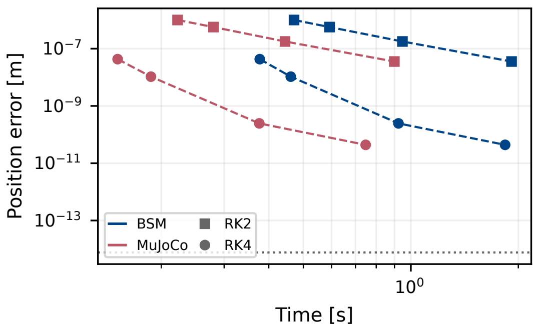
Figures 9 and 10 separate fixed-step from adaptive results. Points farther down and to the left have lower error and cost. Color identifies the engine, marker shape identifies the integrator, and the dotted line marks the resolution of the engine-specific references.



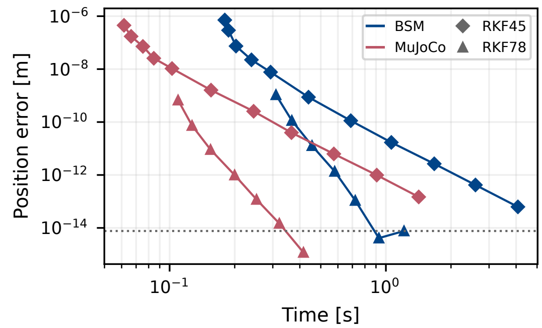
(a) *Fixed-step attitude error for RK2 and RK4.*



(b) *Adaptive attitude error for RKF45 and RKF78.*



(c) *Fixed-step position error for the same runs as panel (a).*



(d) *Adaptive position error for the same runs as panel (b).*

Figure 10 *Work-precision results for the stiff 16-segment flexible array. Euler is omitted because it is unstable over the useful step-size range. The dotted levels bound the resolution of comparisons against the engine-specific references.*

MuJoCo is faster in all 36 matched wheel-and-panel configurations. Within each integrator family, the BSM and MuJoCo curves have nearly the same error dependence but different runtime. On the logarithmic axis, this multiplicative cost difference appears as an approximately horizontal translation. The translation is specific to each integrator family rather than one constant shift.

The 16-segment study gives the same ranking in all 26 matched configurations (Figures 10a–10d). Changing the integrator or spacecraft changes the curve shape; changing the dynamics engine primarily changes its horizontal position. Euler produces useful points for the wheel-and-panel model but none for the more stability-limited segmented array. Values below the dotted thresholds cannot support further accuracy comparisons, but they still show the propagation cost required to reach the resolution of the study.

7 Discussion

The numerical results separate three sources of disagreement. The BSM idealizes wheel mass and bob centroidal inertia as zero, while the corresponding MuJoCo bodies retain small finite values. Applying gravity at the individual body centers produces a torque absent when the same point-gravity law is applied at the total center of mass. The remaining torque-case discrepancy is a numerical artifact in the MuJoCo 3.11.0 bias-force calculation. With these causes identified, the two formulations can be compared in terms of model construction, runtime, and their value as independent implementations within one toolkit.

7.1 Model construction and scope

The formulations assign model-development work to different parts of the software. For a spacecraft component already represented in the BSM, the coupling equations and numerical implementation are part of the Basilisk library. A scenario author selects that component and supplies its physical parameters through a spacecraft-specific interface. Common spacecraft models are therefore concise to assemble, but that concision depends on the existence of a suitable component model.

```python
effector = NHingedRigidBodyStateEffector()
effector.r_HB_B = [[sign*PANEL_HINGE_X], [0.0], [0.0]]
effector.dcm_HB = panelRootDcm(sign)

for i in range(nSegments):
    panel = HingedPanel()
    panel.mass = segmentMass(nSegments)
    panel.d = segmentHalfLength(nSegments)
    panel.k, panel.c = PANEL_K, PANEL_C
    panel.IPntS_S = inertia
    panel.thetaInit = ROOT_THETA0 if i == 0 else 0.0
    effector.addHingedPanel(panel)

scObject.addStateEffector(effector)
```

(a) *BSM construction. The effector defines the serial-chain topology.*

```xml
<body name="pP_0" pos="0.8 0 0" quat="0 0 0 1">
  <joint name="pP_0" axis="0 1 0" stiffness="600" damping="5"/>
  <inertial pos="-0.75 0 0" mass="30" diaginertia="0.28125 5.625 5.625"/>

  <body name="pP_1" pos="-1.5 0 0">
    <joint name="pP_1" axis="0 1 0" stiffness="600" damping="5"/>
    <inertial pos="-0.75 0 0" mass="30" diaginertia="0.28125 5.625 5.625"/>
  </body>
</body>
```

(b) *Equivalent two-segment MJCF subtree. Longer chains repeat the nested body block.*

Listing 1 *Equivalent segmented-panel construction.*

A component outside the BSM library requires a new derivation of its Backsubstitution Method terms, a C++ implementation, and separate verification. The hub-centered formulation also constrains articulated topology. A moving link can carry a force- or torque-generating model, so a thruster may be mounted on an articulated arm. However, the link cannot serve as the base of another mechanism with independent generalized coordinates. Branched articulated systems below the hub, such as a multi-finger gripper at the end of an arm, are therefore outside the scope of BSM.

In turn, MuJoCo shifts this work into the model description. The scenario defines each body's inertial properties and parent joint in MJCF, while the general solver assembles the equations of motion. A new articulated configuration usually requires a new body tree rather than a new dynamics derivation. The tree can branch from any body, and MuJoCo provides contact and constraint dynamics that are not implemented in the BSM. The modeler must describe and check the mechanism explicitly, but does not have to add a component-specific multibody formulation to Basilisk.

The segmented-panel scenario illustrates the two construction paths. Listing 1 places the generic BSM construction in panel (a) beside the corresponding two-segment MJCF subtree in panel (b). Both use the same masses, inertias, hinge laws, and geometry. Simulation setup and module prefixes are omitted; the complete constructors are archived with the scenario.[‡]

The excerpts expose this division of work; they do not measure usability or development time. The BSM panel is shorter because the serial-chain derivation is already in the library, whereas the MuJoCo panel states the body tree explicitly. For a supported spacecraft architecture, the BSM requires less topology description in the scenario. For a new articulated architecture, MuJoCo avoids the component-specific dynamics development required by the BSM.

7.2 Runtime across the benchmark suite

Neither engine is uniformly cheaper. The BSM/MuJoCo ratios are 0.45 for point-mass translation and 0.71 for the compact orbiting reaction-wheel model. The torqued rigid body and extended orbital model are within 10% of parity, with ratios of 0.97 and 0.93. MuJoCo is faster in the no-gravity wheel-and-panel and variable-mass workloads, with ratios of 1.71 and 1.16, and at every size in the segmented-panel sweep. At 70-DOF, the ratios reach 4.43 for `nHingedRigidBody` and 9.60 for `spinningBodyNDOF`.

The runtime differences follow partly from the engines' model primitives. MuJoCo builds mechanisms from a fixed set of multibody primitives, principally bodies and joints, and advances the coordinates associated with those primitives. Within this representation, a scenario cannot replace one component with custom reduced equations. The BSM instead permits each specialized component to supply its own equations and retain only the states required by that model. Such reductions may define the ideal system of interest in a controlled study. They may also reduce cost by omitting cyclic or dynamically irrelevant states.

The compact reaction-wheel case illustrates the computational use of specialization. The BSM balanced-wheel model advances wheel rate but omits spin angle because the angle is cyclic. MuJoCo represents each wheel as

[‡] `scenarioCompareFlexPanels.py` at 9581a59f8f7f

a hinged child body and advances both joint position and velocity. This difference helps explain why the BSM is faster for the compact orbiting wheel model. Adding the two hinged panels narrows the BSM advantage to near parity, consistent with a lower marginal cost for extending MuJoCo's recursive body tree. The no-gravity wheel-and-panel case favors MuJoCo. An analogous modeling distinction appears in the slosh case: the BSM can represent the bob as a point mass, whereas MuJoCo realizes it as a finite rigid body. The complete variable-mass workload also favors MuJoCo, so the choice of primitives does not determine the runtime ranking by itself.

The work-precision studies test whether the relative costs arise from the selected integrator. Within each integrator family, the two engines have nearly the same curve shape and differ mainly by a horizontal translation. Since time is plotted on the horizontal axis, this translation means that both engines approach the reference at nearly the same rate as the step or tolerance is tightened, but one pays a different propagation cost to reach that error. The wheel-and-panel and segmented-panel rankings persist across every tested fixed-step and adaptive integrator.

These measurements cover dynamics propagation rather than the cost of a complete mission simulation. Logging, model construction, and initialization are excluded, and the scenarios contain little of the GNC and environment workload present in an operational simulation. The runtime difference between the engines will therefore be smaller in a complete mission simulation unless dynamics dominates the total workload.

7.3 Value of two dynamics formulations in one toolkit

Basilisk exposes both dynamics formulations through the same simulation interfaces. A scenario can exchange the BSM and MuJoCo engines while retaining its integrator, scheduler, environment models, message connections, and GNC logic. This removes most of the integration work that usually separates two simulation tools and narrows an investigation to the dynamics model itself. The engines also provide complementary scope. The BSM supplies specialized spacecraft components for its supported hub-centered topologies, while MuJoCo accepts general body trees together with contact and constraint dynamics. Both remain available without dividing the surrounding mission-simulation software into separate toolchains.

Implementing the same physical system through both engines also forces the model specification to become more precise. Quantities that appear interchangeable at the scenario level may encode different assumptions about mass properties, force application points, damping, or retained states. A cross-engine residual exposes the consequence of those choices, but attribution still requires analytic references and controlled variants. The shared Basilisk infrastructure reduces unrelated differences between runs, although it also means that the comparison is not fully independent: an error in common infrastructure can affect both engines. The distinct equations-of-motion implementations provide a strong internal check, not a substitute for external validation.

The matched cases also provide regression tests for later development. Each disagreement must be explained by an analytic reference, a controlled variant, or an explicit difference in the physical model. The executable scenarios preserve these relationships, so later changes can be checked against expected cross-engine behavior as well as engine-specific unit tests.

The timing comparisons had a similar development role. Scaling measurements motivated matrix storage and operations better suited to the BSM multibody effectors. Profiling also removed repeated state-map copies from the shared Runge–Kutta path, reduced repeated model and state access in the MuJoCo wrapper, and identified contact and constraint work that could be disabled in models that do not use those features. The value of the benchmark was therefore not limited to ranking two final runtimes. Matched workloads made performance costs visible and provided repeatable tests for the resulting optimizations.

Maintaining a second dynamics formulation carries a real software and verification cost. In return, it provides an independently implemented dynamics path that a larger collection of single-engine unit tests cannot replace. Within a common toolkit, that path supports model verification, regression testing, performance analysis, and informed selection between specialized and general multibody representations. The comparison infrastructure is consequently most useful as a permanent part of the simulation capability, rather than as a one-time study performed after an engine is complete.

8 Conclusion

The engines agree closely when they encode the same physical model. Both translational orbits match the analytic solution at the expected integration-truncation level, while the wheel-and-panel and segmented-panel cases agree near floating-point precision. The variable-mass and slosh histories also remain closely matched. The small rotational residuals in the wheel and slosh cases are governed largely by the massless-wheel and point-mass bob idealizations in the BSM. Larger controlled differences arise from applying gravity at separate body centers and from velocity-dependent cancellation in the MuJoCo 3.11.0 bias-force implementation.

Runtime depends on the spacecraft representation. The BSM takes 0.45× and 0.71× the MuJoCo time in the specialized translational and compact reaction-wheel cases. The torqued rigid body and extended orbital model remain within 10% of parity. MuJoCo takes less time in the no-gravity wheel-and-panel and variable-mass workloads, where the BSM/MuJoCo ratios are 1.71 and 1.16. MuJoCo also remains faster throughout the segmented-panel sweep, where the measured growth is less steep than for either BSM effector over the largest models. The work-precision studies show that the wheel-and-panel and segmented-panel rankings are not peculiar to RK4: across the tested fixed-step and adaptive integrators, switching engines primarily translates the curves horizontally in propagation time.

The BSM offers concise and efficient construction when a suitable specialized effector already exists, but extending its component library requires new dynamics derivations and its hub-centered topology excludes branched articulated mechanisms below the hub. MuJoCo represents general body trees and includes contact and constraint dynamics. Placing both formulations in the same simulation framework made those differences measurable and turned the comparison into a practical method for validating dynamics behavior and improving propagation performance.

Acknowledgment

This work is supported by a NASA Space Technology Graduate Research Opportunity (NSTGRO) Grant 80NSSC24K1394.

References

[1] Steven P. Hughes, Rizwan H. Qureshi, Steven D. Cooley, and Joel J. Parker. Verification and validation of the general mission analysis tool (gmat). In *AIAA/AAS Astrodynamics Specialist Conference*, 2014.

[2] Francesco Castellini, Stephan Kranz, Steffen Weber, Bernard Godard, Ruaraidh Mackenzie, Paul Steele, and Jan Siminski. Development and usage of the GODOT astrodynamics software at Telespazio Germany. In *SpaceOps 2023: 17th International Conference on Space Operations*, Dubai, United Arab Emirates, March 2023. Paper ID 655.

[3] Luc Maisonobe, Véronique Pommier, and Pascal Parraud. Orekit: An open source library for operational flight dynamics applications. In *4th International Conference on Astrodynamics Tools and Techniques (ICATT)*, ESAC, Madrid, Spain, May 2010.

[4] Kartik Kumar, P. W. L. van Barneveld, Dominic Dirkx, J. C. P. Melman, Erwin Mooij, and Ron Noomen. Tudat: A modular and robust astrodynamics toolbox. In A. Benoit, editor, *Proceedings of the 5th International Conference on Astrodynamics Tools and Techniques (ICATT)*, pages 1–8, Noordwijk, The Netherlands, 2012. ESA.

[5] Eric Stoneking. 42: An open-source simulation tool for study and design of spacecraft attitude control systems. NASA Goddard Space Flight Center, NTRS Document 20180000954, Report No. GSFC-E-DAA-TN52069, February 2018. Software available at `https://github.com/ericstoneking/42`.

[6] Patrick W. Kenneally, Scott Piggott, and Hanspeter Schaub. Basilisk: A flexible, scalable and modular astrodynamics simulation framework. *Journal of Aerospace Information Systems*, 17(9):496–507, 2020.

[7] Cody Allard, Manuel Diaz Ramos, and Hanspeter Schaub. Computational performance of complex spacecraft simulations using back-substitution. *Journal of Aerospace Information Systems*, 16(10):427–436, 2019.

[8] John Alcorn, Cody Allard, and Hanspeter Schaub. Fully coupled reaction wheel static and dynamic imbalance for spacecraft jitter modeling. *Journal of Guidance, Control, and Dynamics*, 41(6):1380–1388, 2018.

[9] João Vaz Carneiro, Cody Allard, and Hanspeter Schaub. General dynamics for single- and dual-axis rotating rigid spacecraft components. *Journal of Spacecraft and Rockets*, 61(4):1099–1113, 2024.

[10] João Vaz Carneiro and Hanspeter Schaub. Spacecraft dynamics with the backsubstitution method: Survey and capabilities. In *AAS Spaceflight Mechanics Meeting*, Kauai, Hawaii, Jan. 19–23 2025. Paper No. AAS 25-232.

[11] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.

[12] Erwin Mooij and Marcel Ellenbroek. Multi-functional guidance, navigation, and control simulation environment. *Journal of Physics: Conference Series*, 08 2007.

[13] Greg Dell, Mark Hametz, Paul Noonan, Lauri Newman, David Folta, and John Bristow. Eos am-1 and eo-1 support using freeflyer and autocon. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 1998.

[14] Juan Garcia-Bonilla, Carl Leake, Asher Elmquist, Tristan D. Hasseler, Vivian Steyert, Aaron Gaut, and Abhinandan Jain. Dshell-darts: A reusability-focused multi-mission aerospace and robotics simulation toolkit. In *2025 IEEE Aerospace Conference*, pages 1–13, 2025.

[15] João Vaz Carneiro and Hanspeter Schaub. Scalable architecture for rapid setup and execution of multi-satellite simulations. *Advances in Space Research*, 73(11):5416–5425, 2024. Recent Advances in Satellite Constellations and Formation Flying.

[16] Tom Erez, Yuval Tassa, and Emanuel Todorov. Simulation tools for model-based robotics: Comparison of Bullet, Havok, MuJoCo, ODE and PhysX. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4397–4404, Seattle, WA, 2015.

[17] Wei Qian, Zeyang Xia, Jing Xiong, Yangzhou Gan, Yangchao Guo, Shaokui Weng, Hao Deng, Ying Hu, and Jianwei Zhang. Manipulation task simulation using ros and gazebo. In *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, pages 2594–2598, 2014.

[18] Giovanni Boschetti and Teresa Sinico. Designing digital twins of robots using simscape multibody. *Robotics*, 13(4), 2024.

[19] Serena Ivaldi, Vincent Padois, and Francesco Nori. Tools for dynamics simulation of robots: A survey based on user feedback, February 2014.

[20] Daniel G. Murri, E. Bruce Jackson, and Robert O. Shelton. Check-cases for verification of 6-degree-of-freedom flight vehicle simulations. Technical Report NASA/TM-2015-218675/Volume I, NESC-RP-12-00770, NASA Langley Research Center, Hampton, VA, January 2015.

[21] João Vaz Carneiro and Hanspeter Schaub. Spacecraft dynamics with the backsubstitution method: Survey and capabilities. In *AAS Spaceflight Mechanics Meeting*, Kauai, Hawaii, Jan. 19–23 2025. Paper No. AAS 25-232.

[22] Roy Featherstone. *Rigid Body Dynamics Algorithms*. Springer, New York, 2008.

[23] João Vaz Carneiro, Cody Allard, and Hanspeter Schaub. Software architecture of sequentially rotating rigid spacecraft components using the backsubstitution method. *Journal of Aerospace Information Systems*, 23(7):597–613, 2026.

[24] Andrew M. Morell, João Vaz Carneiro, Leah Kiner, and Hanspeter Schaub. Multi-arm post-docking spacecraft dynamics using penalty methods. In *AIAA SCITECH 2024 Forum*, 2024.

[25] F. Dodge. The new dynamic behavior of liquids in moving containers. Technical report, Southwest Research Institute, San Antonio, TX, 2000.

[26] Cody Allard, Hanspeter Schaub, and Scott Piggott. General hinged rigid-body dynamics approximating first-order spacecraft solar panel flexing. *Journal of Spacecraft and Rockets*, 55(5):1291–1299, 2018.