Propagation and Optimisation

Assignment I

Lorenz Veithen

TU Delft

DELFT UNIVERSITY OF TECHNOLOGY

FACULTY OF AEROSPACE ENGINEERING

AE4866-1

PROPAGATION AND OPTIMISATION IN ASTRODYNAMICS

Lunar Ascent Trajectory Propagation and Optimisation

Integrator and Propagator Selection

Lecturers: Dr. D. Dirkx

Github repository link: https://github.com/LorenzVeithen/PropagationOptimisation_Lorenz_Veithen

May 14, 2023

Hours spent: $\approx 40 - 50$h

Lorenz Veithen 5075211

Cooperating Students: Srujan Vaidya (5072034) and Oliver Ross (5008042)

TU Delft Delft University of Technology

[1]Cover Image URL: https://www.wallpaperflare.com/search?wallpaper=Lunar+lander

1 Propagation Requirements

The following requirements are formulated for the choice of the integrator and propagator. The numerical error is chosen two orders of magnitude lower than the physical error, and the benchmark error three orders of magnitude lower than the numerical. This is done to ensure that the errors happening during the mission only arise due to the choice of the physical model. The ascent is assumed to aim to reach the initial position of the rendezvous approach, for which initial relative distances and velocities of the order of 1 km and 10 m/s respectively are common[1], and an error of $\approx 1\%$ is deemed permitted. A physical error of only 0.01 kg for the propellant mass ($\approx 0.0004\%$ of total), due to its mission criticality and to ensure compliance with the velocity accuracy (0.01 kg of propellant difference can lead to a change in ΔV of 0.01 m/s, which is smaller than PROP-2). Additionally, a requirement on the CPU time per run is formulated to keep the optimisation process feasible.

Table 1: Propagation requirements.

REQ ID	Requirement
PROP-1	The position accuracy of the physical modelling shall be 10m or smaller.
PROP-1.1	The numerical position accuracy shall be 0.1m or smaller.
PROP-1.2	The benchmark position accuracy shall be 1e-4m or smaller.
PROP-2	The velocity accuracy of the physical modelling shall be 0.1m/s or smaller.
PROP-2.1	The numerical velocity accuracy shall be 1e-3m/s or smaller.
PROP-2.2	The benchmark velocity accuracy shall be 1e-6m/s or smaller.
POP-3	The physical mass modelling accuracy shall be 0.01kg or smaller.
PROP-3.1	The numerical mass accuracy shall be 1e-4kg or smaller.
PROP-3.2	The numerical mass accuracy shall be 1e-7kg or smaller.
PROP-4	CPU time per run shall be 2s or smaller.

2 Benchmark

In this section, an appropriate time step size Δt is selected for the benchmark solution with default propagation parameters. A RKF7(8) integrator is used, and the error of the Δt propagation is estimated using the $\Delta t/2$ one, through Eq. (1). In the Local Truncation Error (LTE) regime, dividing the time step size by 2 should reduce the error by a factor 128 with a 7th order integrator.

$$\epsilon(t;\Delta t) \approx \bar{y}_1(t;\Delta t) - \bar{y}_2\left(t;\frac{\Delta t}{2}\right) \tag{1}$$

2.1 Benchmark Δt Selection

The Δt's considered are 2^i for i=4,3,...,-7,-8. The six first and last epochs of each propagation are excluded, where numerical noise arises due to the 8th order Lagrange interpolator used to evaluate $\bar{y}_1$ at the epochs of $\bar{y}_2$. Using Eq. (1), Fig. 1 is obtained. It is clear that for $\Delta t \leq$ 2e-2s, rounding errors are dominant, and larger Δt's are in the LTE region. The slope of the graph is expected to be 7 in a log-log plot, following Eqs. (2) and (3). However, this is not the case in Fig. 1, due to the definition of the error used in Eq. (1), which only gives an indication of the error's magnitude, and not its true value.

$$\boldsymbol{\epsilon}_{LTE}(t+\Delta t;\Delta t) = \sum_{i=8}^{\infty} \frac{(\Delta t)^i}{i!}\frac{d^i}{dt^i}(\boldsymbol{y}(t)) \approx \boldsymbol{K}_8(t)(\Delta t)^8 \qquad \boldsymbol{K}_8(t) = \frac{1}{8!}\frac{d^8}{dt^8}(\boldsymbol{y}(t)) \tag{2}$$

$$\boldsymbol{\epsilon}_{GTE}(t) = \sum_{i=0}^{i=n_t} \boldsymbol{\epsilon}_{LTE}(t_i+\Delta t) \sim \mathcal{O}(\Delta t^7) \qquad n_t\text{, number of time steps until time t} \tag{3}$$

To fit PROP-1.2, Fig. 1 indicates that $\Delta t \leq$ 0.2s is necessary; for PROP-2.2, Fig. 3 indicates that $\Delta t \leq$ 0.3s is necessary, and PROP-3.2 is always satisfied according to Tab. 2 (always in rounding error region). A time step size of $\Delta t = 0.1$s is then chosen for the benchmark, to allow for some margin with PROP-2.2, for when a broader acceleration environment will be included. Following, the error in the position and velocity as a function of time for this time step size, along others, is plotted in Fig. 2. It is clear that discontinuities and/or spikes arise at the nodes of the trajectory, and are more significant for larger Δt. This is due to the particular behaviour close to the nodes (where a change in the acceleration environment occurs) being missed by the coarser propagation $\bar{y}_1$, while it is better captured by the finer propagation $\bar{y}_2$.

[1] see Fundamentals of Astrodynamics, week 7.

A similar behaviour is observed for the velocity in Fig. 4. Furthermore, even when no spikes arise, the error sharply increases around the nodes due to the complex dynamics, showing that a variable step method adapting to a lower Δt around the critical region of the nodes could be beneficial.

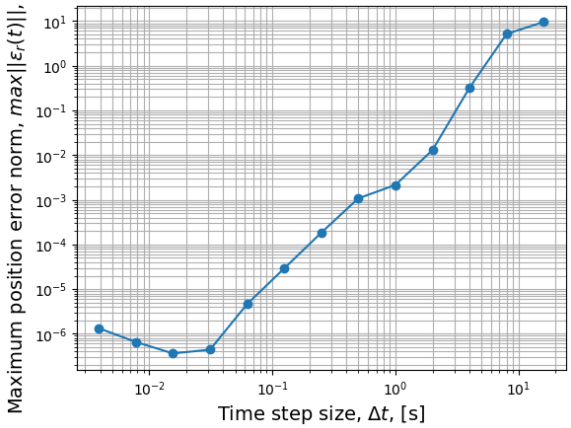


Figure 1: Norm of maximum position error as a function of the time step size.

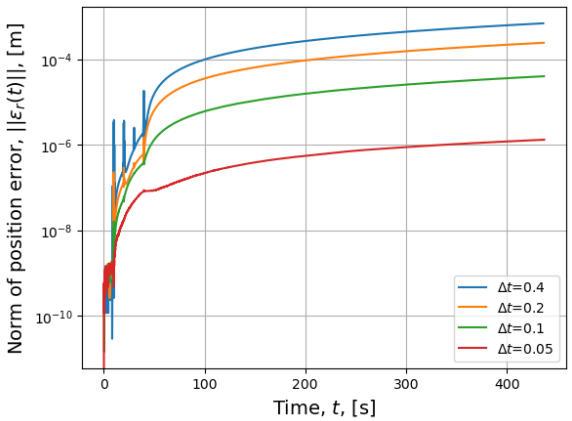


Figure 2: Norm of position error as a function of time for $\Delta t = 0.4, 0.2, 0.1, 0.05$s.

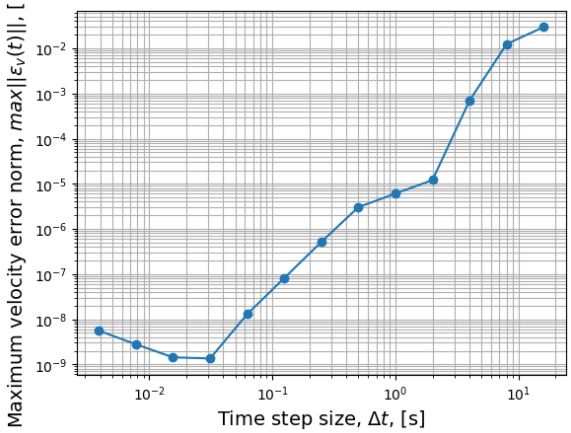


Figure 3: Norm of maximum position error as a function of the time step size. The spikes in error at the nodes are neglected as they are not representative of the general velocity error behaviour.

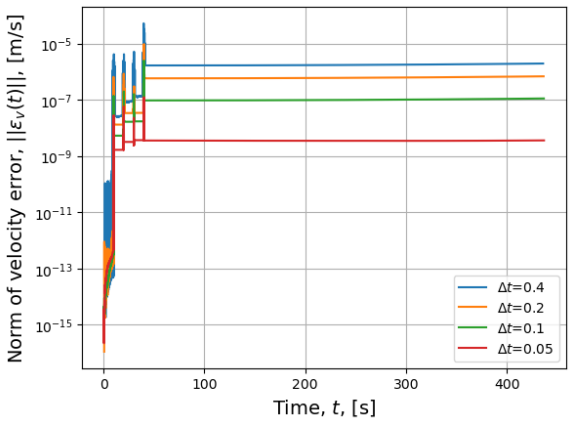


Figure 4: Norm of position error as a function of time for $\Delta t = 0.4, 0.2, 0.1, 0.05$s.

Table 2: Maximum mass error in [kg], for different timesteps sizes, using the RKF7(8) integrator.

Δt [s]	4.0	2.0	1.0	5.0e-1	2.5e-1	1.25e-1	6.25e-2	3.125e-2	1.5625e-2
Order of error	1e-12	1e-12	1e-10	1e-10	1e-12	1e-9	1e-10	1e-9	1e-9

The benchmark used for the following analysis will be generated using the RKF7(8) integrator and $\Delta t = 0.1$s. This benchmark fits the benchmark requirements, its error is dominated by LTE (desired for predictability), and can be run in the order of a few minutes.

2.2 Benchmark Trajectory Analysis

The acceleration environment on the vehicle, a select amount of Kepler elements (a, e, and i, the rest are described in the text below), and the trajectory in a range-altitude plane are plotted. The trajectory is plotted to make sure that it resembles an ascent (no obvious modelling errors). The Kepler elements are interesting to consider as they permit to have a first idea of whether element-based propagators could be suited to the problem. Finally, the acceleration environment could contain some discontinuities or large changes in total acceleration magnitude, which are important to consider for integrator choices.

Near-discontinuities (sharp spikes) in certain elements (ω, i, and Ω) are found, which indicates that element-based propagators likely are not suitable for the propagation (at a first glance). Additionally, even if they were to converge, the fast changing nature of some Keplerian elements (a, e, and ω) during the ascent will likely worsen the performance of such propagators (Keplerian Elements, MEE, USM), as their main advantage arises from having few fast-changing elements, which is not the case in this highly perturbed environment[2]. Furthermore, the acceleration norm close to the thrust nodes is non-linear and fast changing, meaning that the behaviour could be missed by fixed time-step integrators, and result in large increases of the error around the nodes (as already observed earlier). However, the latter is smooth (no discontinuities), and shall not present major problems for multi-step integrators, eg.

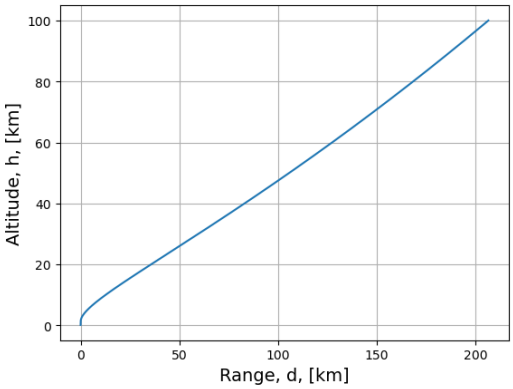


Figure 5: Trajectory of the vehicle. Range based on Moon radius of $R_M = 1737.4$km.

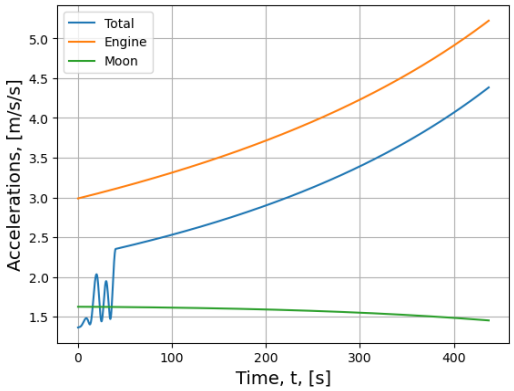


Figure 6: Acceleration environment on the vehicle.

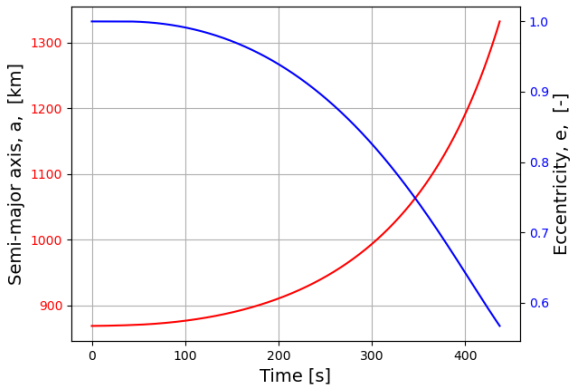


Figure 7: Semi-major axis and eccentricity as a function of time.

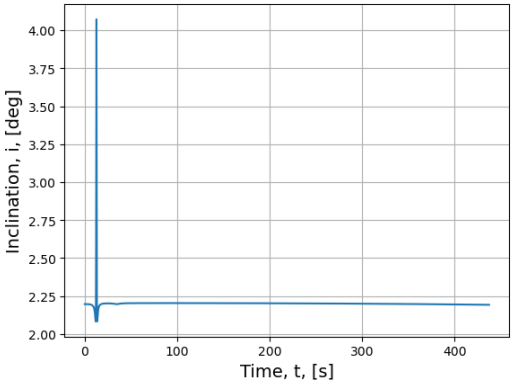


Figure 8: Inclination as a function of time, showing a severe discontinuity close to the thrust nodes.

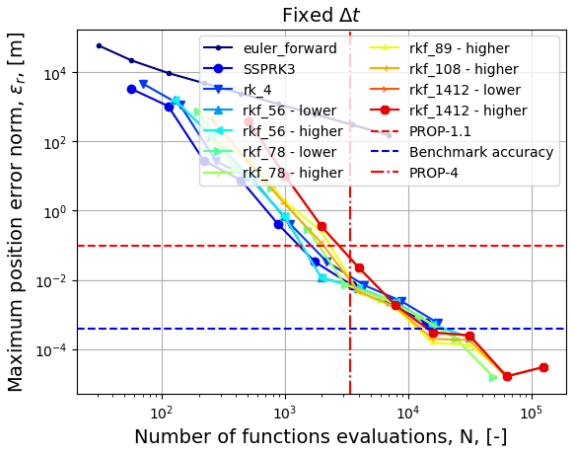
3 Integrator Analysis

Numerous integrator options are considered, analysed and compared in this section, resulting in a shortlist for further analysis. The following integrators are considered: (1) fixed-step multi-stage: forward Euler, SSPRK3, RK4, RK5, RK6, RK7, RK8, RK9, RK10, RK12, RK14; (2) variable-step multi-stage: RKF1(2), RKF4(5), RKF5(6), RKF7(8), RKDP8(7), RKF8(9), RKF10(8), RKF12(10), and RKF14(12); and (3-4) fixed and variable-step Extrapolation Methods with the Bulirsch-Stoer and Deufelhard sequences of orders 2, 4, 6, 8, and 10. This represents a large portion of the integrators available in TUDAT, covering both low and high orders of multi-stage and extrapolation methods[3], to decrease the probability of overlooking an optimal choice. Additionally, methods which are very likely to be inappropriate were still considered (such as RKF1(2)), to show the large differences in capabilities of integrators. Time step sizes of $\Delta t = 2^{-3+i}$s and tolerances of $ToL = 10^{-14+i}$ for i=0,1,..., 8; were considered as integration settings. In the latter, the relative and absolute tolerances were selected equal, with $\Delta t_{min} = 10^{-6}$s and $\Delta t_0 = 1$s. Following, the analysis is considered invalid if the predicted error falls lower than $10||\epsilon_{bench}||$ (for the variable under consideration). The required accuracy performance of the integrators is considered using PROP-1.1, 2.1, 3.1: the Cowell propagator is generally among the least accurate, meaning that the accuracy ensured in this section will likely improve if another more appropriate propagator is later selected. Furthermore, to ensure compliance with PROP-4, the requirement was translated to a maximum number of function evaluations through Tab. 3[4].

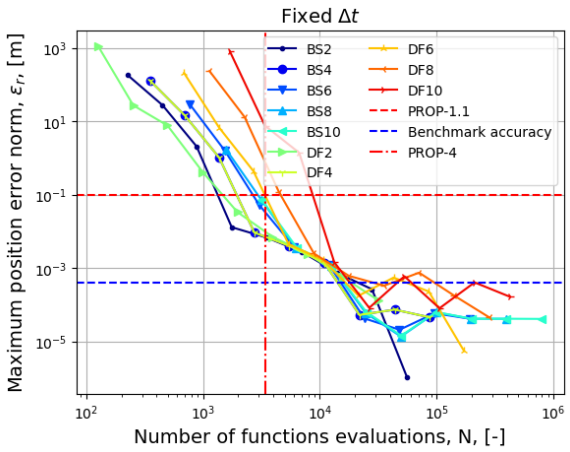
[2] Acceleration from engine thrust is twice as large as the Moon point mass.

[3] Multi-step methods were discarded following an announcement made in week 2.

[4] Results generated using a MacBook Pro M1 Max 32GB on MacOS Ventura, run natively.

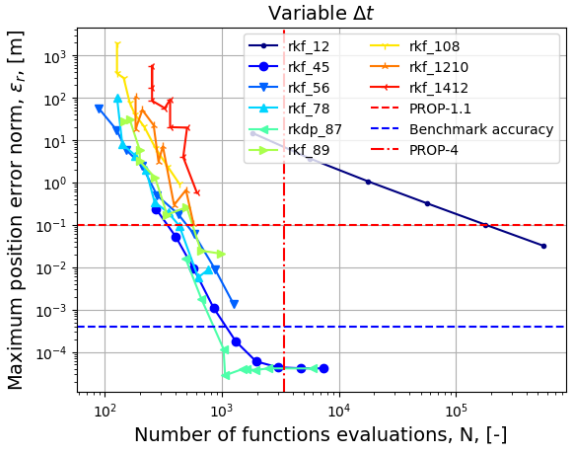


(a) Multi-stage integrators.

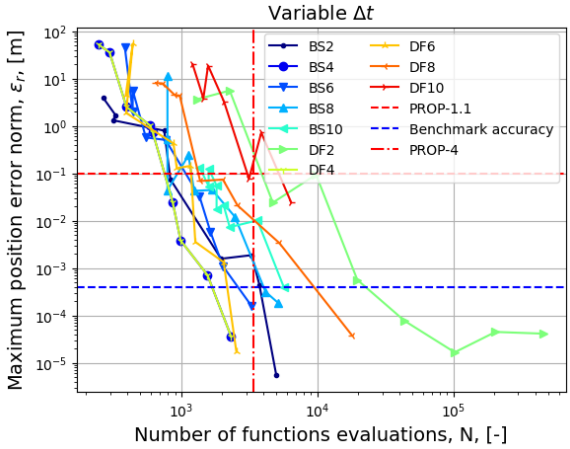


(b) Extrapolation-method integrators.

Figure 9: Maximum position error as a function of the number of evaluations, of fixed time step integrators, with $\Delta t = 0.125, 0.25, 0.5, 1, 2, 4, 8, 16, 32$s.



(a) Multi-stage integrators.



(b) Extrapolation-method integrators.

Figure 10: Maximum position error as a function of the number of evaluations, of variable time step integrators, with $ToL = 10^{-14}, 10^{-13}, 10^{-12}, 10^{-11}, 10^{-10}, 10^{-9}, 10^{-8}, 10^{-7}, 10^{-6}$.

Table 3: Statistics of the time per function evaluation. These statistics only aim to give a first idea of the number of function evaluations equivalent to 2s CPU time.

Statistics	**Value**
Total function evaluations	8057986
Mean, μ, [s]	3.5094e-4
Standard deviation, σ, [s]	7.8529e-05
$\mu + 3\sigma$, [s]	6.6231e-4
PROP-4, n_{max}, [-]	3410

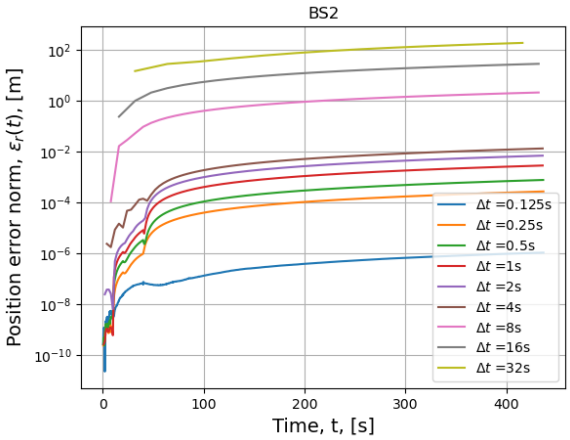


Figure 11: Maximum error as a function of time for the fixed Bulirsch-Stoer order 2 extrapolation method.

Note that it was found that both the mass and velocity requirements (PROP-2.1 & 3.1) are systematically satisfied if PROP-1.1 is met. Therefore, only the latter is discussed in details in the following. From Figs. 9 and 10, the following observations on the integrators are noted (all data points below the striped blue line are to be discarded due to the benchmark accuracy):

- From Fig. 9a, the fixed-step multi-stage integrators behave as expected, as with decreasing time steps (more functions evaluations), the maximum error becomes smaller. However, compared to the variable-step multi-stage integrators in Fig. 10a, their performance with respect to the requirements is much lower. That is the case because the

Table 4: Number of time steps taken for different variable time step integrators. The qualitative behaviour as discussed in text is given in the last column.

Method	1e-14	1e-13	1e-12	1e-11	1e-10	1e-9	1e-8	1e-7	1e-6
RKF4(5)	-	-	-	-	-	109	71	47	-
RKF5(6)	137	93	62	-	-	-	-	-	-
RKF7(8)	-	38	27	-	-	-	-	-	-
RKDP8(7)	-	-	-	-	-	-	-	30	22
BS2	-	-	-	72	35	-	-	-	-
BS4	-	44	29	22	-	-	-	-	-
BS6	-	20	17	14	-	-	-	-	-
DF4	-	44	29	22	-	-	-	-	-
DF6	-	-	-	-	13	9	-	-	-

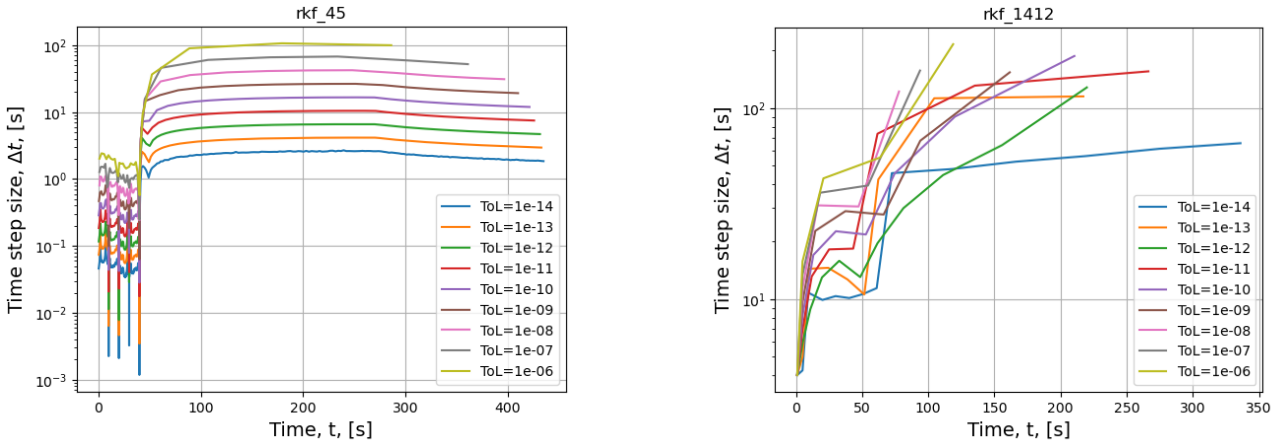
fixed step integrators do not adapt to the dynamics, meaning that both the non-linear behaviour close to the thrust nodes (trajectory critical), and the very predictable behaviour after the last node (see Fig. 6) are sampled with the same time step. This requires to impose a small Δt (large N) to obtain in a similar accuracy, resulting in an inefficient use of resources. Furthermore, as expected, higher order methods require more function evaluations for the same accuracy, but can use larger Δt (4s for RKF14 against 1s for SSPRK3). SSPRK3 (1s), RK4 (1s), RKF5 (2s), and RKF6 (2s) are the most promising integrators from this category. They are the further from the requirement boundaries, and would not require investigating different settings to fit within the requirements (like RKF14, eg.).

- From Fig. 9b, it appears that for similar maximum position errors, the Deufelhard sequence extrapolation methods are more computationally intensive than using the Bulirsch-Stoer sequence equivalent (DF6, 8, 10 are all in the right part of the graph). Within the region of interest, the behaviour is roughly linear, except for a kink for the BS2 and DF/BS4 methods (in the design space), considering Fig. 11, it is seen that this kink arises from the near-discontinuous error jump (where the error rises the most rapidly) being larger for the first node for $\Delta t = 8$s, while it is larger for the last node for the lower Δt. This change in behaviour gives rise to a different slope for $\Delta t \leq 4$s than for $\Delta t \geq 4$s. Nevertheless, a few good candidates for further analysis exist: the BS2 (4s), DF2 (2s), BS4 (4s), and DF4 (4s) versions, as they meet both requirements.
- From Fig. 10a, it is seen that all lower order variable step multi-stage methods until RKF8(9) (non-included), show the expected monotonously decreasing trend with increasing tolerance, until the benchmark error becomes significant. However, higher order methods show an erratic behaviour, with some not reaching the PROP-1.1 specification despite tolerances as low as 10^{-14}. This can be explained by comparing the time step behaviour of the RKF4(5) and RKF14(12) methods, as shown in Fig. 12. It is clear that the former is able to capture the fast dynamics of the thrust nodes (decreasing pikes in Δt), and as a result, all propagations follow similar dynamics, and the error as a function of N behaves well. The latter, however, shows that the nodes are not captured correctly as the first time steps are estimated too large, and the different curves are subject to significantly different dynamics, giving the random behaviour in Fig. 10a. The first time steps taken too large could be a result of higher order terms in Eq. (2) being non-negligible, and the Δt control algorithm breaking down in comparing the 14th and 12th order integrators. Good candidates for further analysis include RKF4(5) ($ToL = 10^{-8}, 10^{-7}$), RKF5(6) ($ToL = 10^{-13}, 10^{-12}$), and RKF7(8) ($ToL = 10^{-13}, 10^{-12}$) [5]. RKDP8(7) with a lower tolerance of 1e-5 could be an appropriate choice to be closer to the specification of PROP-1.1, but would result in too few time steps (dense output is desired, see Tab. 3).
- From Fig. 10b, a significant amount of integrators meet both PROP-1.1 and PROP-4. However, the higher order integrators (BS8, BS10, DF8, DF10), similarly to the multi-stage case, show a less monotonous behaviour. The reasons are similar to the multi-stage case: the higher order methods tend to take larger time steps due to the assumption of negligible higher order terms in Eq. (2) being invalid, and miss the specific behaviour at the nodes. Good candidates for further analysis include the BS4 ($ToL = 10^{-12}, 10^{-11}$), DF4 ($ToL = 10^{-12}, 10^{-11}$), and BS6 ($ToL = 10^{-12}, 10^{-11}$), as they fit both requirements, and are generally predictable in the vicinity of PROP-1.1. Particularly, the BS2 and DF6 methods show a number of kinks which are undesired.

For each of the preliminary selections shown above, the discussions were based on the data points lying above the benchmark validity line in the plots. No inherently random behaviour in the error as a function of time of any integrator was found (except at the very start of the propagation), meaning that the error of all considered propagations were in the LTE regions. The two main reasons for discarding an integrator was that: (1) it does not meet the accuracy and/or CPU time requirements; (2) showing an unpredictable behaviour (kinks / oscillations). Furthermore, a relatively dense output is desired to avoid losing accuracy by interpolating between the accurate data points. Hence, using Tab. 4 for variable time steps integrators, methods using less than 25[6] steps (for any tolerance) are discarded (BS6). The following integrators will then be considered in subsequent analysis: **fixed:** SPPRK3, RK4, RK5, RK6, BS2, DF2, BS4, DF4; and **variable:** RKF4(5), RKF5(6), RKF7(8), BS4, and DF4.

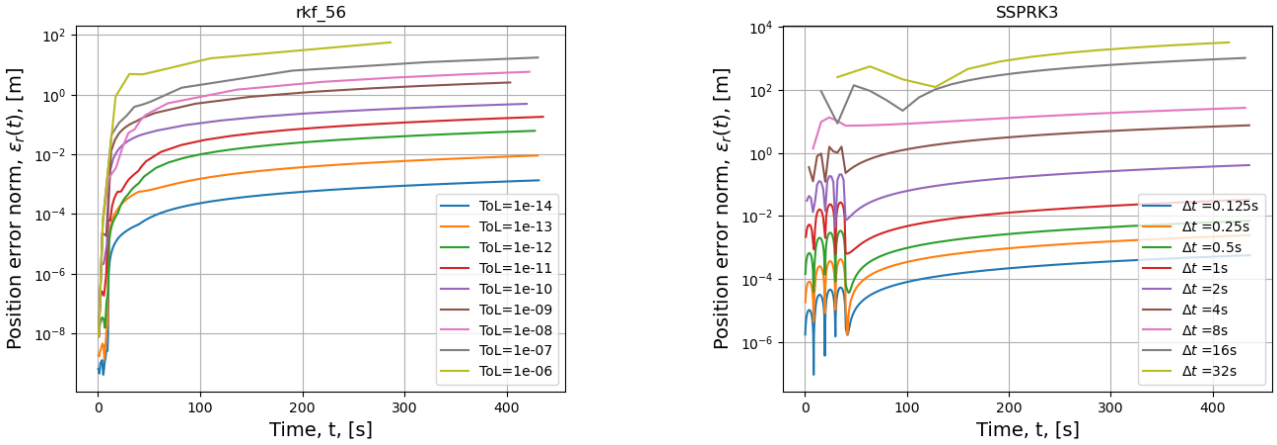
[5] RKF8(9) is also subject to the behaviour discussed earlier, and is less predictable

[6] Purely indicative value.



(a) RKF4(5). (b) RKF14(12).

Figure 12: Time step size as a function of time, with $ToL = 10^{-14}, 10^{-13}, 10^{-12}, 10^{-11}, 10^{-10}, 10^{-9}, 10^{-8}, 10^{-7}, 10^{-6}$.



(a) RKF5(6). (b) SSPRK3.

Figure 13: Error as a function of time for two selected integrators for the subsequent analysis.

4 Propagator Analysis

4.1 Fixed Time-Step Integrator Selection

As the multi-stage integrators generally behave more smoothly, and are better known to the author, the extrapolation methods will not be considered in order to analyse the different propagators available. This leaves only a few fixed-step integrators which meet both the CPU time and accuracy requirements (position, velocity, and mass): SSPRK3, RK4, and RKF5(6) (fixed, lower and higher orders). However, the RKF5(6)-based methods have a small kink in Fig. 9a, making them less predictable (which is important for the subsequent analysis). Both the SSPRK3 and RK4 have a similar behaviour and could be used for the propagator analysis. However, the SSPRK3 method shows an unusual error behaviour in Fig. 13b, while RK4 has a more conventional error behaviour. Furthermore, the latter is widely used across the industry for its simplicity and robustness, permitting to better attribute behaviours to the integrator or to the propagator. This leaves the RK4 integrator, with $\Delta t = 1$s, as the most suitable choice for a fixed-step integrator. Using a Cowell propagator, RK4 propagates the dynamics in $\approx$ 0.7s on average, with $max||\epsilon_r(t)|| \approx$ 3.326e-2 m, $max||\epsilon_v(t)|| \approx$ 9.314e-5 m/s and mass accuracy of the order of 1e-10 kg. The integrator does not rely on previous steps, and performs reasonably well at handling discontinuities in the dynamics (eg. acceleration norm seen earlier), as the goal of the method is to find the 'average' slope on the time step, which is permitted to be large. This makes the RK4 method a robust choice, if Δt is chosen small enough.

4.2 Performance of Different Propagators

The same propagation was run for the selected integrator type and settings. However, out of the seven propagators considered, only two completed the propagation: the Cowell and Encke propagators. The others either cross one of the altitude termination conditions around 12s (region close to the first thrust node), or result in an unsuccessful run. This was expected from the discussion in Subsection 2.2, based on the near-discontinuities / spikes in the inclination, argument of pericenter, and right ascension of the ascending node angles (although the reason presented was not accurate). This behaviour is analysed through Fig. 14 for $\Delta t = 0.125$s with RK4 (a smaller Δt is selected to better understand the

behaviour). The discontinuity arises in Ω, due to the ascent module having a positive thrust angle initially, and switching sign on each subsequent node. This causes the trajectory to first increase and then decrease in longitude, as seen in Fig. 15, and increase Ω beyond the 360°value. This causes a discontinuity in the plot, but the values themselves are continuous (370°=10°). Furthermore, the behaviours of the inclination and argument of pericenter, despite being fast-changing, are not discontinuities. Another characteristic of the trajectory is seen from the behaviour of the eccentricity, which reaches a maximum value of $e = 0.9999999997713495$ at $t = 12.625$s (retrieved numerically). As $e = 1$ is a singularity of the Gaussian Kepler equations, this is likely the reason for the Keplerian propagator to not converge successfully: with $\Delta t = 1$s, the eccentricity is wrongly predicted to cross the $e = 1$ value, and the propagation fails. However, those traits are only manifestations of the dynamics which generally cause the MEE and USM propagators to break down.

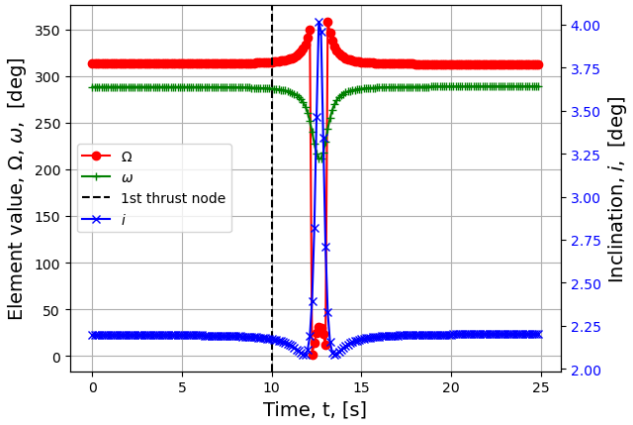


Figure 14: Behaviour of the inclination, argument of pericenter, and right ascension of the ascending node, Keplerian elements in the first 25s of the flight. Based on conversion from Cartesian coordinates.

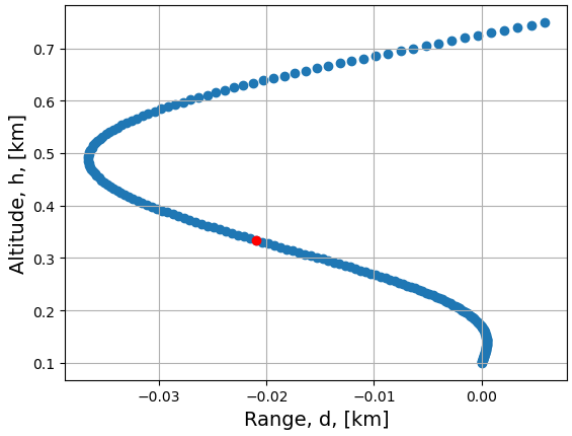


Figure 15: Altitude-range plot for the first 25s of the flight. The red dot corresponds to the point of maximum eccentricity.

Further analysis of Fig. 15 shows that the point of maximum eccentricity seems to be the inflection point (in range as a function of altitude) of the trajectory, which indicates that the velocity vector reaches a near vertical orientation (in local coordinates). This is further confirmed by Fig. 16, showing $\alpha = \arccos\left(\frac{\boldsymbol{r}\cdot\boldsymbol{V}}{||\boldsymbol{r}||||\boldsymbol{V}||}\right)$, which tends to zero where the element-based models break down. This is expected, as the shape of the orbit reduces to a single line when those vectors align, meaning that the definition of the orientation angles for the Keplerian, Mean Equinoctial, and Unified State Model elements break down. Furthermore, these vectors being aligned comes back to the angular momentum becoming zero, which is a known singularity of these types (element-based) of propagators. While α never exactly reaches zero, it comes close enough such that a large error arises due to the very fast changing shape of the orbit close to the discontinuity. This large error causes the USM propagators to reach one of the altitude termination conditions, which explains why their propagation stops after about 12s. This is seen in Fig. 18 and 19, where the unprocessed data for the USM-Exponential Map propagator is given (all other USM propagators behave similarly, with the exception that an additional parameter η replaces the shadow parameter omitted here, and shows a similar behaviour to σ_i's), showing a very sudden behaviour upon reaching 11-12s due to the angular momentum reaching values close to zero. Note that the Mean Equinoctial Elements are not given in this document due to the page limit constraint, but all elements (except p) rise quickly in the region of $t \approx 12$s, before the propagation stops for the reasons mentioned earlier.

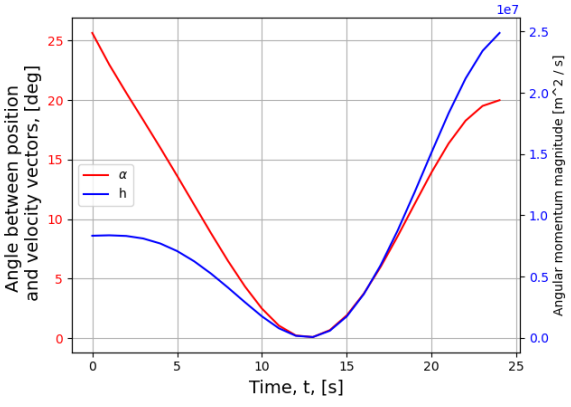


Figure 16: Angle between the 3D position and velocity vectors and angular momentum of the orbit, as a function of time.

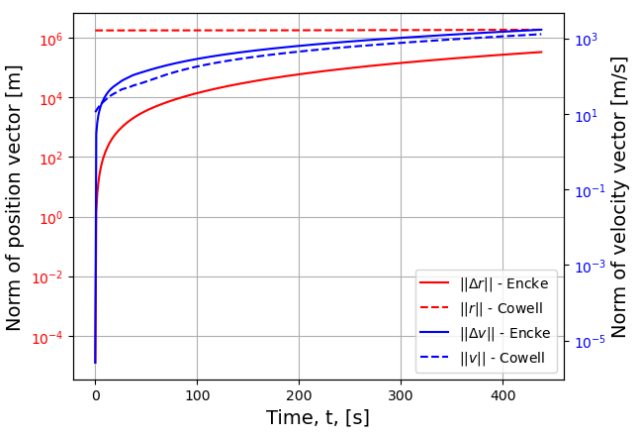


Figure 17: Comparison between the Cowell and Encke position and velocity component vectors norms.

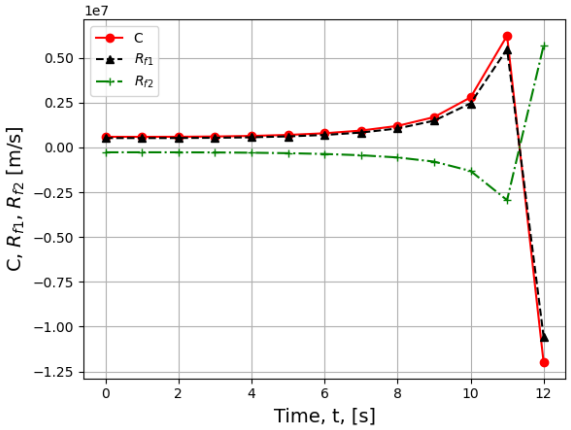


Figure 18: First three elements of the USM Exponential Map representation, as a function of time.

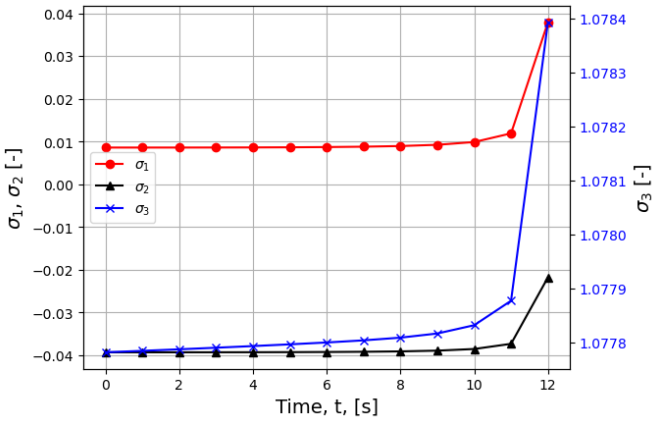


Figure 19: Last four elements of the USM Exponential Map representation, as a function of time.

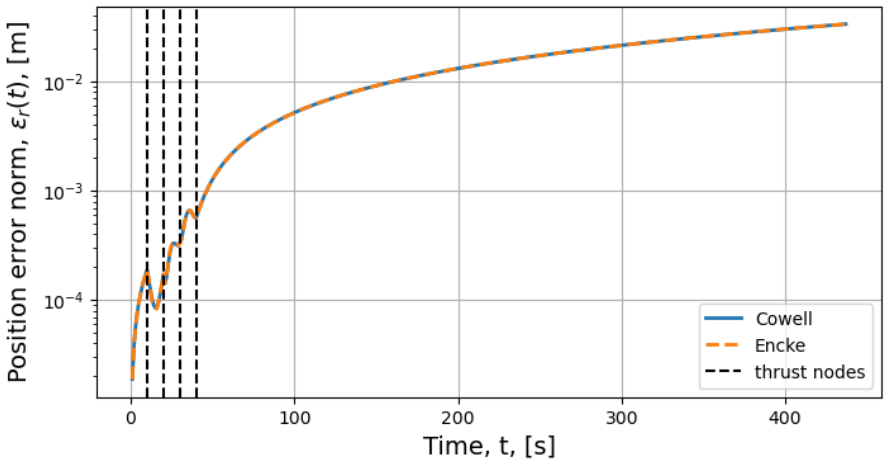


Figure 20: Norm of position error for the propagators which complete the full trajectory successfully, and reach the final epoch, with an RK4 integrator with $\Delta t = 1$s.

This therefore only leaves the Cowell and Encke propagators as viable options, as they do not show any risk of coming close to a singularity during the optimisation process. The main difference between those propagators, is that Encke computes the Cartesian position and velocity difference with respect to the Keplerian orbit with the same initial state at t_0. This permits, in lightly perturbed cases to have smaller state derivative (variations), reducing the errors due to the large variations in magnitude of the components (easier numbers to work with). However, the problem under consideration is highly perturbed (thrust magnitude is twice as large as the Moon point mass gravity field), and the benefits are quickly lost as the state magnitude becomes similar to the Cowell state magnitude, as seen in Fig. 17. This is confirmed by Fig. 20, showing that no significant difference in accuracy is found between the two propagators. In this plot, the sharp variations in norm of the position error are due to the non-linear dynamics close to the nodes. No significant difference in computational efficiency is observed in the CPU time per function evaluation, and the same number of function evaluations (2191) is observed for both (as expected for a given fixed step integrator). In the following section, both propagators will be considered. However, for essentially equal performance, the Cowell propagator would be preferred, as its formulation is more straightforward.

5 Final Propagator - Integrator Combination Selection

5.1 Selection of Combinations

In the previous two sections, a shortlist of potential integrators and propagators was generated. The list of integrators will not be revisited, as the analysis was performed based on a Cowell propagator, which is similar in formulation to Encke, being the only other feasible propagator. All possible combinations of the shortlisted integrators and propagators will be considered, as this only results in 26 combinations. The following integrators are considered with both a Encke and Cowell propagator; **fixed step:** SPPRK3 (Δt=1s), RK4 (Δt=1s), RK5 (Δt=2s), RK6 (Δt=2s), BS2 (Δt=4s), DF2 (Δt=2s), BS4 (Δt=4s), DF4 (Δt=4s); and **variable step:** RKF4(5) ($ToL = 10^{-8}, 10^{-7}$), RKF5(6) ($ToL = 10^{-13}, 10^{-12}$), RKF7(8)

($ToL = 10^{-13}, 10^{-12}$), BS4 ($ToL = 10^{-12}, 10^{-11}$), and DF4 ($ToL = 10^{-12}, 10^{-11}$). An extensive argumentation for the choice of those integrators (and not others) was provided in Section 3. Summarising, those options fit both PROP-1.1, 2.1, 3.1 and PROP-4, their accuracy does not come too close to the benchmark, their behaviour in terms of maximum position error as a function of the number of function evaluations is predictable (no kinks / oscillations in the vicinity of the requirement specifications), and their output is not too sparse. For all variable step methods, the relative and absolute tolerances are kept equal and the minimum time step is taken as $t_{min} = 10^{-6}$s.

5.2 Analysis of the Combinations

From the previous section, it became apparent that for the RK4 method with $\Delta t = 1$s, the Cowell and Encke propagators yielded very similar performance in terms of accuracy. It is therefore verified if this observation can be extended to variable step, and extrapolation methods. It was found that all fixed step integration methods (multi-stage and extrapolation methods) yielded very similar accuracy performance for both propagators. This is expected based on the discussion from the previous section, as the magnitude of the state components of the Encke propagator grow quickly and become similar to the state components used in the Cowell case (1 order of magnitude less in position, and same order of magnitude in velocity), and Encke loses its primary advantage. However, a significant difference in accuracy performance, for the same tolerance, is found for variable step integrators of both types (see Figs. 21 and 22). This is a result of the Δt-control algorithm (which is similar for both multi-stage and extrapolation methods): Eq. (4) is applied iteratively, until all state components satisfy the specified tolerance (ϵ^*_{req}), with one of the position components likely being the most constraining. As the position components magnitude for Encke are at largest one order of magnitude smaller than the Cowell ones (see Fig. 17), the control algorithms behave differently. However, this difference in performance also implies a difference in the number of functions evaluations, and computational load, as seen below. Furthermore, depending on the variable step integrator used, the maximum position error is either higher (eg. RKF5(6) and RKF7(8)), or lower (eg. BS4 and RKF4(5)) using Encke, meaning that no general rule on the relative performance of either propagator for variable step methods can be formulated for this problem.

$$\Delta t_{req} \approx \Delta t \left(\frac{\epsilon^*_{req}}{\epsilon^*} \right)^{\frac{1}{p-1}} \tag{4}$$

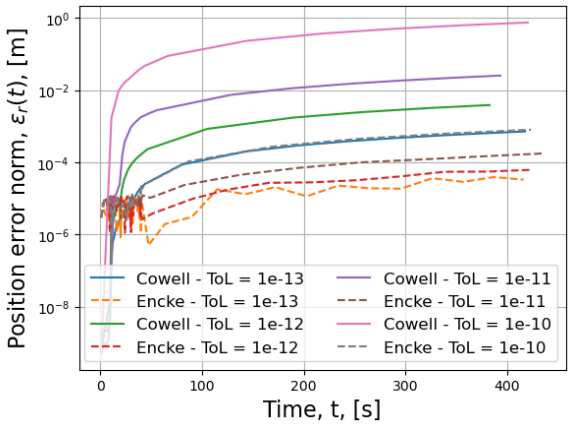


Figure 21: Error as a function of time of the variable time step Bulirsch-Stoer 4th order sequence, extrapolation method.

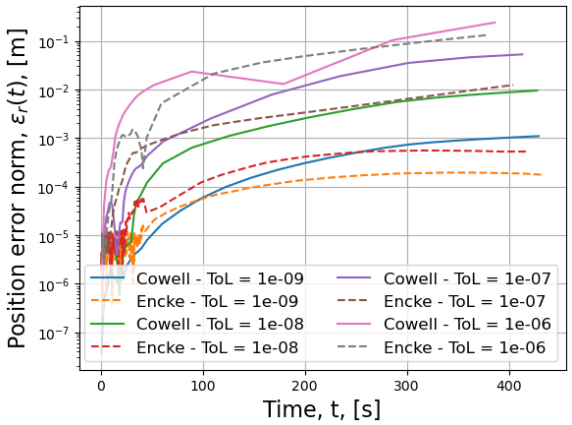


Figure 22: Error as a function of time of the variable time step RKF4(5), multi-stage method.

In the above discussion, it was shown that both propagators need to be considered further (for variable step methods at least). Following, the average time per function evaluation was found to be $t_{Cowell} = 3.468$e-4s (on 755886 evaluations) for Cowell, and $t_{Encke} = 3.474$e-4s (on 768238 evaluations) for Encke; showing that the difference in time per function evaluation of each propagator is minimal. This means that the computational efficiency of each propagator can be compared fairly using the number of functions evaluations per run. Similarly to the analysis done in Section 3, Figs. 23 and 24 present the maximum position error as a function of the number of function evaluations for the different integrator-propagator combinations. Considering the relative position of the curves in Fig. 23, it is clear that fixed step integrators do not provide the best trade-off of accuracy and computational efficiency, when compared to the general position of the curves for the variable step integrators with respect to the requirements[7]. This leaves the options shown in Fig. 24.

[7] Additionally, Fig. 23 confirms the observation on the very similar performance of Encke and Cowell for a given integrator.

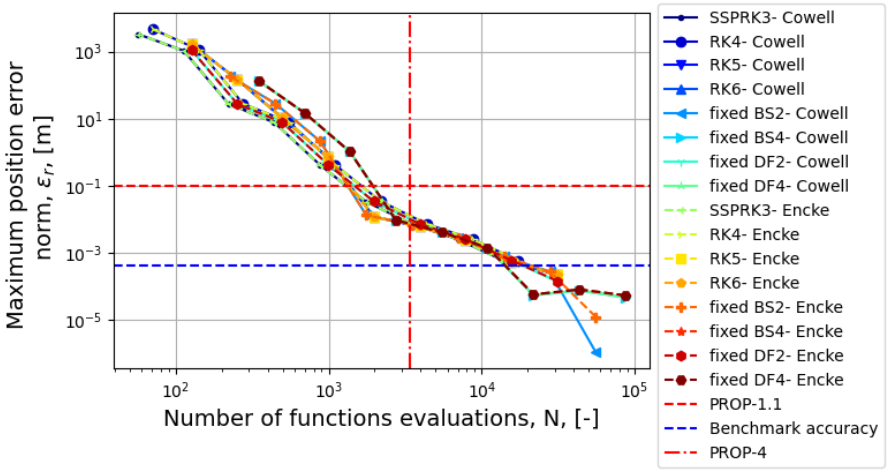


Figure 23: Maximum position error as a function of the number of function evaluations of the selected fixed time step integrator-propagator combinations. Generate with time step sizes of $\Delta t = 0.125, 0.25, 0.5, 1, 2, 4, 8, 16, 32$s.

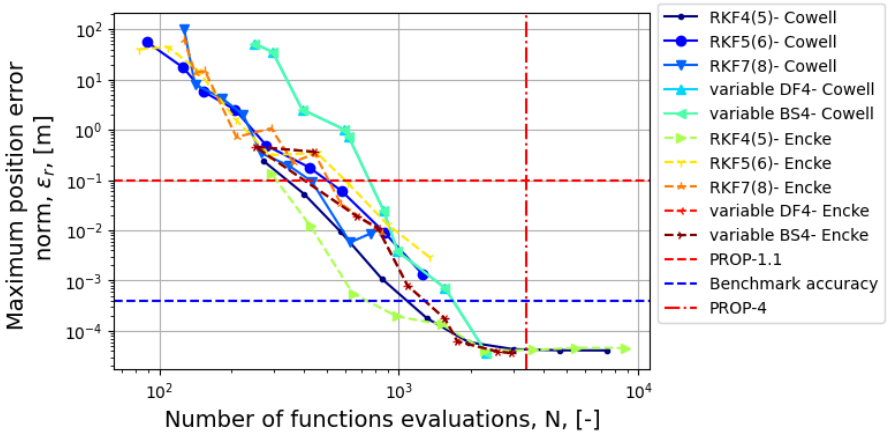


Figure 24: Maximum position error as a function of the number of function evaluations of the selected fixed time step integrator-propagator combinations. Generated with tolerances of $ToL = 10^{-14}, 10^{-13}, 10^{-12}, 10^{-11}, 10^{-10}, 10^{-9}, 10^{-8}, 10^{-7}, 10^{-6}$.

Table 5: RKF4(5) characteristics with $ToL = 10^{-7}$, for the Cowell and Encke propagators.

Propagator	$\underset{t}{max}\|\|\epsilon_r(t)\|\|$ [m]	$\underset{t}{max}\|\|\epsilon_v(t)\|\|$ [m/s]	$\underset{t}{max}\|\|\epsilon_m(t)\|\|$ [kg]	N	CPU time t_{prop} [s]
Cowell	5.26e-2	6.95e-05	2.92e-10	403.0	0.139
Encke	1.23e-2	3.12e-05	3.50e-10	427.0	0.149

Considering Fig. 24, two options come out of the rest: RKF4(5) with either Cowell or Encke as propagator (assuming no additional tolerance settings are considered). Both behave in a very predictable fashion in the region where it fits the requirements, and their key characteristics are given in Tab. 5. It is clear that both options fit the PROP-1.1, -2.1, and -3.1 requirements, and result in a propagation CPU time vastly below the specified value from PROP-4. While the Encke propagator offers a better overall accuracy (at the cost of a 0.01s longer run time), it was originally designed for lightly perturbed trajectories which do not deviate significantly from the analytical Keplerian orbit (which is not the case for a lunar ascent). This could result in varying performance depending on the thrust parameters input, as significantly different deviations from the initial orbit can arise. This can become an issue for the optimisation of the thrust parameters inputs to fine tune the lunar ascent trajectory, as a large range of different parameters will be considered ($\sim 10^5$ runs), with potentially of varying numerical performance. As the RKF4(5) integrator with a Cowell propagator and a tolerance of 1e-7 (relative and absolute) satisfies all requirements, requires less function evaluations, and is less prone to varying performance for different thrust parameters (due to the simplicity of the Cowell propagator and robustness of the multi-stage method); this propagation configuration will be selected for the upcoming analysis. The position error and the time step size as a function of time for this integrator-propagator combination were shown in Figs. 22 and 13a respectively.