(Preprint) AAS 23-318

A HYBRID STOCHASTIC-DETERMINISTIC INTEGRATOR FOR SPACECRAFT DYNAMICS WITH UNCERTAINTY

Carmine Giordano* **and Francesco Topputo†**

High-fidelity spacecraft trajectory propagation can become a cumbersome task when dealing with uncertainties modeled as random processes. The stochastic differential equations describing the uncertain dynamics can be numerically integrated, but they are challenging from the computational point of view. Traditional methods usually require either the storage of a relevant amount of data or small integration steps. In this work, an hybrid method, embedding a stochastic integration method in a deterministic higher-order scheme, is conceived to obtain fast and stochastically correct results. Results show a reduction of at least one order of magnitude for both computational time and memory usage with respect to state-of-the-art techniques, while it is able to provide statistically correct results.

INTRODUCTION

In a real-life scenario, a spacecraft is unlikely to follow the prescribed nominal path, due to uncertainty in the *dynamics* (e.g., gravitational parameters or radiation pressure noisy profiles), *navigation* (i.e. imperfect state knowledge or approximations in measurement model), and *command actuation* (i.e., thrust magnitude and pointing angles error).[1] The correct quantification of these uncertainties and their impact on a spacecraft trajectory are required tasks in space operations, e.g., to evaluate the observability of a spacecraft trajectory or to assess the collision probability risk with another object.[2] In mission analysis, uncertainty quantification is of paramount importance to determine the flyability of trajectories and as consequence the feasibility of a spacecraft mission. In fact, it is not uncommon that, even if they could be nominally exploited, some trajectories are not usable after an uncertainty assessment is performed due to the costs[3] or the risks.[4]

Uncertainties related to spacecraft trajectory analysis are both random variables (e.g., measurement errors, drag coefficient, mass parameters, usually modeled as Gaussian variables), and random processes (e.g., solar radiation pressure, low thrust acceleration). While random variables usually do not pose any additional complexity from the computational point of view, stochastic processes could be difficult to handle, due to their noisy dynamics. However, their impact on the propagation under uncertainty of space trajectory is relevant and an improper modelization can lead to under- or over-estimate the stochastic characteristics of the given trajectory. For traditional spacecraft, errors in the estimation of the dispersion, i.e., the distance between the real and the nominal trajectory, are usually a minor problem, since they have the control authority and the propellant to compensate them in flight. However, an increasing number of future space exploration mission is foreseen to

*PostDoc Fellow, Department of Aerospace Science and Technology, Politecnico di Milano, via La Masa, 34, Milan, 20156, Italy.

†Full Professor, Department of Aerospace Science and Technology, Politecnico di Milano, via La Masa, 34, Milan, 20156, Italy.

exploit CubeSats,[5,6] characterized by limited-control authority, due to low thrust levels and reduced propellant budget. In this case, dispersion misestimate can lead to discard feasible trajectories (in case of over-estimation) or to select unflyable trajectories, requiring a large amount of propellant when in flight (in case of under-estimation). For this reason, it is critical to be able to propagate the trajectories in a high-fidelity setting considering properly their stochastic background.
To this aim, the integration of a system of stochastic differential equations (SDE) is required. In recent times, numerical methods to obtain both weak and strong approximations for the solution of stochastic differential equations have been developed,[7,8] with a special interest in derivative free Runge-Kutta schemes.[9–13] In the last years, the work by Rossler[14] introduced some efficient stochastic Runge-Kutta (SRK) schemes, that exploit the colored rooted tree analysis[15] to reduce significantly the number of function evaluation required for any single step. The possibility to create natural embedded pairs, able to both perform the step and estimate the errors with no extra function evaluation,[16] increased the flexibility of the SRK and allowed to create efficient adaptative methods. However, the complexity of the SRK methods grows with the approximation order, since more terms should be considered in the generating Itô–Taylor expansions.[17] Moreover, the extra terms involve the computation of high-order iterated and cross-term Itô integrals, for which a simple approximation, like the one by Wiktorsson,[18] is still missing. For this reason, higher-order SRK schemes could be impractical and, even though strong order 2.0 SRK methods have been introduced,[7] existing work mainly focused on order 1.5 schemes.
This limitation in the convergence order strongly reduces the use of pure SDEs solvers in astrodynamics applications, since they usualy have great time horizons and stringent tolerances. Indeed, in this case, the number of steps required to provide an accurate solution can be considerable and massive Monte Carlo simulations can be unfeasible due to the unbearable amount of required computational resources.
In this work, a hybrid integration methodology, combining a high-order deterministic method with a lower-order stochastic scheme, is presented, in order to solve in a fast and efficient way the SDEs associated to the orbital dynamics. The paper is organized as follows: in Section , the random ordinary differential equations are illustrated and the approach for a general hybrid integrator is presented in Section . Then it is specialized for a given combination of stochastic and deterministic schemes in Section . The novel strategy is tested in Section 63 and Section 63 concludes the work.

THE RANDOM ORDINARY DIFFERENTIAL EQUATIONS

Generally speaking, a stochastic differential equation can be written in the Itô formulation as[19]

$$\mathrm{d}\mathbf{x}_t = \mathbf{f}\left(t, \mathbf{x}_t\right)\mathrm{d}t + H\left(t, \mathbf{x}_t\right)\mathrm{d}\mathbf{W}_t \tag{1}$$

where $\mathbf{x}_t$ represent the state at a given time t, $\mathbf{f}$ describes the deterministic dynamics, usually defined as the *drift function* in the SDE framework, H measures the stochastic dynamics, i.e., it is the *diffusion function*, and $\mathbf{W}$ is a Brownian motion.
Dynamical systems under uncertain external forces, as the spacecraft dynamics, however, can be modeled as a combination of two sub-domains: 1) a deterministic state, whose dynamics is described by an ordinary differential equation (ODE), that is perturbed by 2) a stochastic state, whose dynamics is described by a stochastic differential equation, which in turn does not depend on the deterministic part. This kind of dynamical systems are described by the so-called random ordinary

differential equations (RODE), and they can be formulated as

$$\dot{\mathbf{x}} = \mathbf{f}\left(t, \mathbf{x}, \boldsymbol{\omega}_t\right) \tag{2a}$$

$$\mathrm{d}\boldsymbol{\omega}_t = \mathbf{g}\left(t, \boldsymbol{\omega}_t\right)\mathrm{d}t + H\left(t, \boldsymbol{\omega}_t\right)\mathrm{d}\mathbf{W}_t \tag{2b}$$

where $\mathbf{x}$ is the state related to the ODE, $\boldsymbol{\omega}$ is the stochastic state, and $\mathbf{f}$ represents the deterministic dynamics. From Eqs. (2), it is clear that the noise is independent from the deterministic state, while the noise acts as an external force for the deterministic dynamics. Ideally, Eq. (2b) can be integrated a-priori and then the result can be fed to Eq. (2a). This procedure, even if effective, could require to store a relevant amount of data, that must be later accessed during Eq. (2a) integration. From the computational point of view, this slows down the code execution and can prevent the use of massive Monte Carlo simulations due to memory saturation. Alternatively, the noise can be generated on the fly, exploiting fixed-step scheme,[20] thus potentially reducing the solution accuracy.
Exploiting the loose coupling between the spacecraft stochastic dynamics, an hybrid scheme, combining a high-order ODE solver with an SDE scheme, can be conceived. It integrates the deterministic part of the dynamics (Eq. (2a)) with a high-order Runge-Kutta scheme, while propagating contemporary the stochastic part (Eq. (2b)) with a lower-order stochastic method. This class of integrators will be able to correctly integrate the process noise, feeding it to the spacecraft dynamics at each integration step. Thus, it can exploits the accuracy of a high-order variable-step scheme, without the need to store data or to perform small steps.

A GENERAL CLASS OF HYBRID INTEGRATORS

For the strong approximation of the solution of an SDE, with reference to Eq. (2b), an s-stage SRK method takes the following form[12]

$$\boldsymbol{\omega}_{n+1} = \boldsymbol{\omega}_n + \sum_{i=1}^{s} \alpha_i h\, \mathbf{g}\left(t_n + c_i^{(0)} h, \boldsymbol{\Omega}_i^{(0)}\right) + \sum_{i=1}^{s} \sum_{\nu \in \mathcal{M}} z_i^{(\nu)} H\left(t_n + c_i^{(\nu)} h, \boldsymbol{\Omega}_i^{(\nu)}\right) \tag{3}$$

with $\boldsymbol{\omega}_n = \boldsymbol{\omega}\left(t_n\right)$, $h = t_{n+1} - t_n$, and

$$\boldsymbol{\Omega}_i^{(\nu)} = \boldsymbol{\omega}_n + \sum_{j=1}^{s} A_{ij}^{(\nu)} h\, \mathbf{g}\left(t_n + c_j^{(0)} h, \boldsymbol{\Omega}_j^{(0)}\right) + \sum_{j=1}^{s} \sum_{\mu \in \mathcal{M}} Z_{ij}^{(\nu,\mu)} H\left(t_n + c_j^{(\mu)} h, \boldsymbol{\Omega}_j^{(\mu)}\right) \tag{4}$$

where

$$z_i^{(\nu)} = \sum_{\lambda \in \mathcal{M}} \beta_i^{(\lambda),(\nu)} \theta_\lambda(h)$$

$$Z_{ij}^{(\nu,\mu)} = \sum_{\lambda \in \mathcal{M}} B_{ij}^{(\lambda),(\nu,\mu)} \theta_\lambda(h)$$

with $\mathcal{M}$ a set of multi-indices with $|\mathcal{M}| = \kappa$, and θ_ν some random variables satisfying

$$E\left[\theta_{\nu_1}^{p_1} \cdot \ldots \cdot \theta_{\nu_\kappa}^{p_\kappa}\right] = \mathcal{O}\left(h^{(p_1 + \ldots + p_\kappa)/2}\right)$$

for all $p_i \in \mathbb{N}_0$ and $\nu_i \in \mathcal{M}$, $1 \leq i \leq \kappa$.
Values of A, B, α, β, and c, are the coefficients of the SRK methods, defining the different schemes. They can be organized in some extended Butcher tableau for their easy representation.

On the other hand, for the approximation of the solution of an ODE, with reference to Eq. (2a), an s-stage Runge-Kutta (RK) method takes the following form

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \sum_{i=1}^{s} \alpha_i h\, \mathbf{f}\left(t_n + c_i h, \mathbf{X}_i, \boldsymbol{\omega}_{n+c_i}\right) \tag{5}$$

with $\mathbf{x}_n = \mathbf{x}\left(t_n\right)$, $\boldsymbol{\omega}_{n+c_i} = \boldsymbol{\omega}\left(t_n + c_i h\right)$, and

$$\mathbf{X}_i = \mathbf{x}_n + \sum_{j=1}^{s} A_{ij} h\, \mathbf{f}\left(t_n + c_j h, \mathbf{X}_j, \boldsymbol{\omega}_{n+c_j}\right) \tag{6}$$

with A, α, and c being the coefficient of the ordinary RK methods. If $A_{ij} = 0$ (= B_{ij} in the SRK case) for $j \geq i$, than Eq. (5) (or Eq. (3)) represent an explicit RK (or SRK) method, that are the focus of this work.
Additionally, both ordinary and stochastic Runge-Kutta methods can be written exploiting naturally embedded higher-order and lower-order pair of temporal integrators, performing the same stage evaluations. This means that the lower-order components are estimated, respectively, as

$$\tilde{\mathbf{x}}_{n+1} = \mathbf{x}_n + \sum_{i=1}^{s} \tilde{\alpha}_i h\, \mathbf{f}\left(t_n + c_i h, \mathbf{X}_i, \boldsymbol{\omega}_{n+c_i}\right) \tag{7}$$

and

$$\tilde{\boldsymbol{\omega}}_{n+1} = \boldsymbol{\omega}_n + \sum_{i=1}^{s} \tilde{\alpha}_i h\, \mathbf{g}\left(t_n + c_i^{(0)} h, \boldsymbol{\Omega}_i^{(0)}\right) + \sum_{i=1}^{s} \sum_{\nu \in \mathcal{M}} \tilde{z}_i^{(\nu)} H\left(t_n + c_i^{(\nu)} h, \boldsymbol{\Omega}_i^{(\nu)}\right) \tag{8}$$

with $\tilde{z}_i^{(\nu)} = \sum_{\lambda \in \mathcal{M}} \tilde{\beta}_i^{(\lambda),(\nu)} \theta_\lambda(h)$, and $\mathbf{X}_i$ and $\boldsymbol{\Omega}_i$ evaluated using the same expressions in Eq. (6) and (4), respectively. In this case, the higher-order scheme is used to advance the solution, while a measure of its difference with the lower-order scheme (e.g., $\epsilon_{\mathbf{x}} = \|\mathbf{x}_{n+1} - \tilde{\mathbf{x}}_{n+1}\|_\infty$ and $\epsilon_{\boldsymbol{\omega}} = \|\boldsymbol{\omega}_{n+1} - \tilde{\boldsymbol{\omega}}_{n+1}\|_\infty$, respectively) is employed to estimate the local integration error, that is later used to adapt the time-step if it exceeds a threshold, following an *adaptive time-stepping algorithm*.
Thus, given the RODE system in Eqs. (2) and its local solution given by Eqs. (3)–(5), a general hybrid scheme is devised. At a given time $t_n \in [t_0, t_f]$, where t_0 and t_f are the time boundaries of the simulation, a forward integration step of length $h = t_{n+1} - t_n$ of the hybrid integrator can be summarized as:

1) the stochastic dynamics is propagated forward from t_n to t_{n+1} using Eq. (3) with a time step $h_{SRK} \leq h$ to obtain $\boldsymbol{\omega}_{n+1}$;

2) the error $\epsilon_{\boldsymbol{\omega}}$ is estimated. If it exceeds a prescribed tolerance, h_{SRK} is reduced exploiting an appropriate adaptive time-stepping algorithm and the procedure restart from Step 1);

3) the stochastic state is interpolated on the time grid $t_n + c_i h,\ i \in \{0, s\}$, identified from the ordinary Runge-Kutta scheme, to obtain $\boldsymbol{\omega}_{n+c_i}$;

4) the deterministic state is propagated forward from t_n to t_{n+1} using Eq. (5) to obtain $\mathbf{x}_{n+1}$;

5) the error $\epsilon_{\mathbf{x}}$ is estimated. If it exceeds a prescribed tolerance, h is reduced exploiting an appropriate adaptive time-stepping algorithm and the procedure restart from Step 3).

An outline of this hybrid integration scheme is provided in Figure 1. From this general outline, different hybrid integrators can be constructed by changing the method associated to each step, specifically:

1) the *SRK method* from Eq. (3), varying the number of stages s and the coefficients,

2) the *SRK adaptive time-stepping algorithm*,

3) the *interpolation method*, projecting the noise into the ordinary differential equation scheme,

4) the *RK scheme* from Eq. (5), and

5) the *RK time-stepping algorithm*, associated to the deterministic step.

The choice of each of these items is bound to the characteristics of the differential equations system to be solved, the required accuracy, and the available computational resources.

EULER-MARUYAMA-RUNGE-KUTTA SCHEME

The Euler-Maruyama-Runge-Kutta (EMRK) algorithm is an integration scheme, based on the general stochastic-deterministic method, conceived to solve problem with a simple stochastic dynamics, but requiring stringent tolerances, such as the RODE system associated to the uncertain spacecraft dynamics. In this case, high-order stochastic Runge-Kutta schemes and stochastic step-varying algorithm are not needed. The stochastic dynamics is then integrated by exploiting a *fixed-step Euler-Maruyama scheme*, i.e.,

$$\boldsymbol{\omega}_{n+1} = \boldsymbol{\omega}_n + h_{EM}\, \mathbf{g}(t_n, \boldsymbol{\omega}_n) + H(t_n, \boldsymbol{\omega}_n)\, \Delta W_n \tag{9}$$

where $\Delta W_n \sim \mathcal{N}(0, h_{EM})$ is the increment in the Brownian path. The step-length size h_{EM} is selected a-priori. On the other hand, dynamics in Eq. (2a) is integrated exploiting an ordernary Runge-Kutta scheme, such as Dormand-Prince method[21] or the Verner's most efficient Runge-Kutta 8(7) pair,[22] as done in this work. However, the deterministic time-stepping algorithms used commonly for solving ODEs cannot be exploited, since they will throw away information about the future Wiener process which biases the sample statistics for the path. For this reason, a method preserving information about the future path is built, adapting the Rejection Sampling with Memory (RSwM) algorithm[16] to the hybrid integrators. The main differences are that the step-size is determined by the deterministic part of the RODE and the interpolation of the SDE solution onto the deterministic one must be taken into account. The Brownian bridge[23] will be exploited both to manage the variable step-size and the noise interpolation. By the properties of the Brownian bridge, considering a general Wiener process with $W(t_1) = a$ and $W(t_2) = b$, then

$$W(t) \sim \mathcal{N}\left(a + \frac{t - t_1}{t_2 - t_1}\,(b - a)\,, \frac{(t_2 - t)\,(t - t_1)}{t_2 - t_1}\right) \tag{10}$$

for $t \in [t_1, t_2]$.

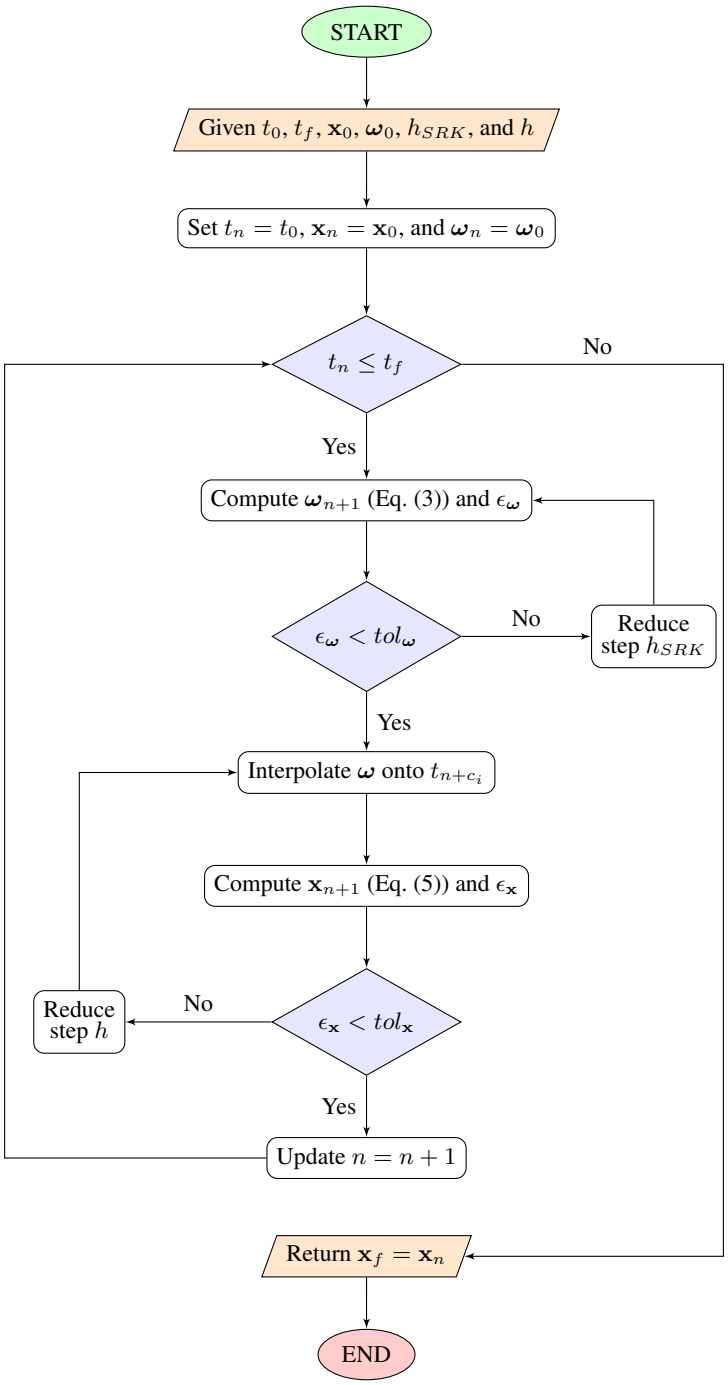


Figure 1: Outline of the general deterministic-stochastic hybrid integrator with double adapting time-step.

Modified Rejection Sampling with Memory

Following the scheme outlined by the RSwM, the odified Rejection Sampling with Memory (mRSwM) algorithms uses two stacks S_1 and S_2, with the first containing future information and the latter the re-popped ones. Each element of the stacks is a tuple containing the integration time at which it is associated and the values of the Wiener process in Eq. (9), e.g., $S_{1,n} = (t_n, \Delta W_n)$. It is important to note that the value of the noise $\boldsymbol{\omega}$ is not saved in the stacks, but rather re-computed from scratch when needed. This is related to the fact that the Brownian bridge can be applied directly only to Wiener paths and not on other type of processes.
An overview of the mRSwM is given in Figure 2:

(a) At a general time 0, the stacks S_1 and S_2 are not empty and contains future (i.e., after the time-span h) and re-popped information (i.e., within the time-span h, respectively. Note that even if S_1 could be empty, S_2 contains at least 2 elements, that are the values of the process at 0 and h;

(b) The Wiener process is interpolated in the time instants required by the ordinary RK algorithm (red dots in the figure) by exploiting the Brownian bridge and inserted into S_2. Eq. (2b) is integrated using Eq. (9) and the information contained in S_2. Later, Eq. (2a) is solved by Eq. (5). The acceptable error is computed by

$$\tau = \mathrm{tol} \cdot \|\mathbf{x}\|_\infty \tag{11}$$

where tol is the desired tolerance, and the step-size variation as

$$q = 0.9 \left(\frac{\tau}{\epsilon_{\mathbf{x}}} \right)^{\frac{1}{8}} \tag{12}$$

If $\tau \leq \epsilon_{\mathbf{x}}$, the step is accepted:

(c) S_2 is emptied. The information from S_1 within the new step-length $h_{new} = qh$ are later moved onto S_2. Additionally, in general, the element associated to the end of the new step should be added on top of S_2 by interpolating its last element with the first element of S_1.
If $\tau > \epsilon_{\mathbf{x}}$, the step is rejected:

(d) The information from S_1 within the new step-length $h_{new} = qh$ are moved onto S_2, and the element associated to the end of the new step is added on top of S_2 by interpolating its last element with the first element of S_1.
After that this cycle is repeated until the final time is reached.

Moreover, special care should be paid when adding new elements to S_2. Since the integration step for the stochastic part must be equal or lower than h_{EM}, if two elements in S_2 are separated in time by a greater interval, additional time-step should be added by exploiting the Brownian bridge. A pseudocode version of the mRSwM is given in Algorithm 1.

RESULTS

In order to evaluate the validity of the hybrid Euler-Maruyama-Runge-Kutta method, test on both correctness and efficiency are performed. The dynamics under test is a 2-body problem, with

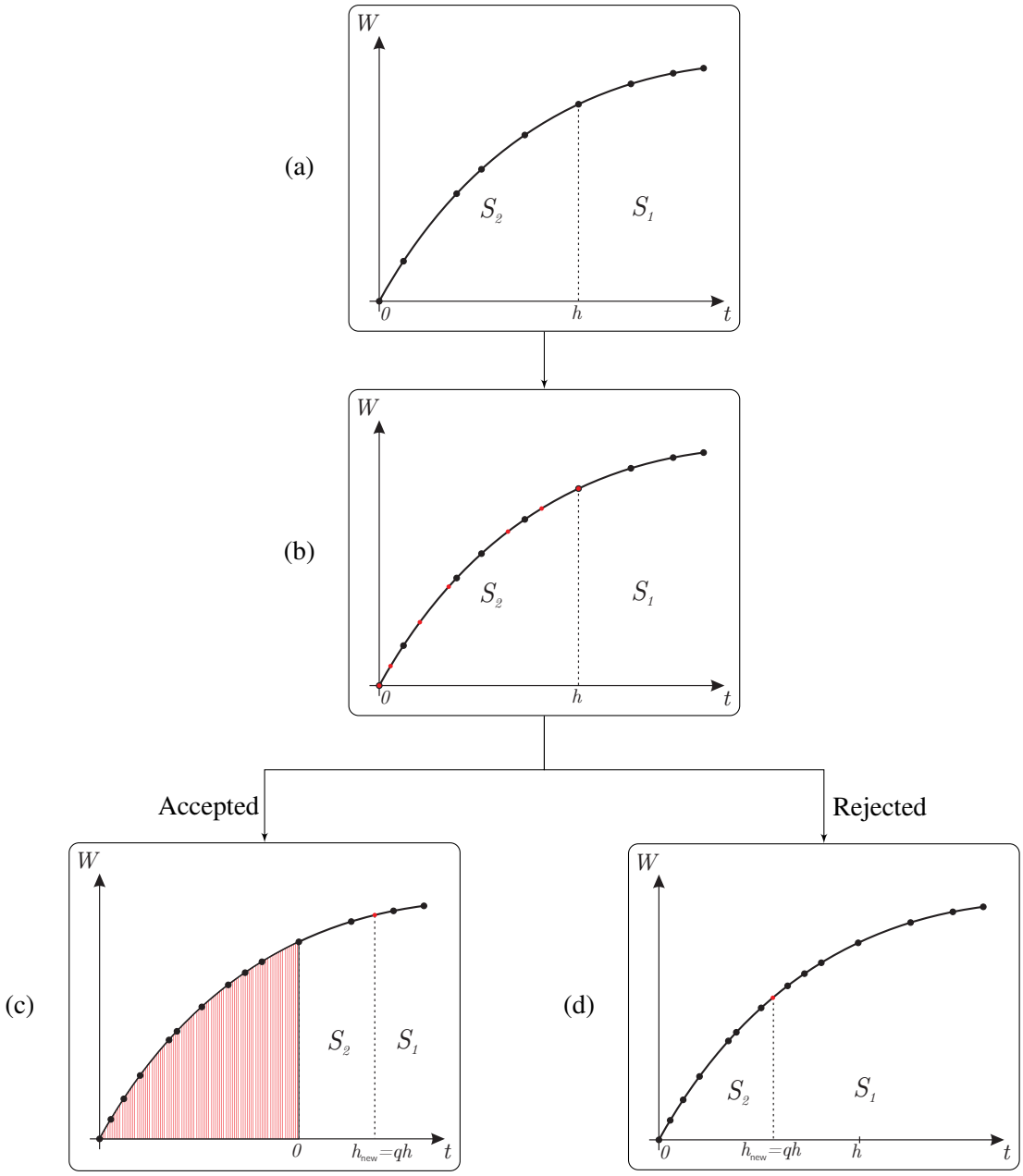


Figure 2: Scheme for the adaptive time-step algorithm embedding the modified Rejection Sampling with Memory.

Algorithm 1: Euler-Maruyama-Runge-Kutta scheme

Set the values $\epsilon_{\mathbf{x}}, \epsilon_{\boldsymbol{\omega}}, h_{\max}, h_{EM}$
Set the initial values $\mathbf{x} = \mathbf{x}_0, \boldsymbol{\omega} = \boldsymbol{\omega}_0$
Take an initial h
Initialize the time vector $t_{EM} = (0, h_{EM}, \ldots, Nh_{EM})/h$ with $N = h/h_{EM}$
Initialize $\Delta W \sim \mathcal{N}_N(0, h_{EM})$
Initialize S_1 empty and $S_2 = (t_{EM}, \Delta W)$
while $t_n \leq t_f$
  for $j = 1, \ldots, s$ ▷Find the interval in S_2 for c_j and interpolate the noise
    for $k = 1, \ldots, N$
      Set $P = S_{2,k}$
      if $P_1 - c_j \geq 0$
        break
      end
    end
    Set $Q = S_{2,k+1}$
    Take $\Delta W \sim \mathcal{N}\left(\frac{c_j - P_1}{Q_1 - P_1} P_2, \frac{\left(c_j - P_1\right)\left(Q_1 - c_j\right)}{Q_1 - P_1} h\right)$
    Insert $(c_j, \Delta W)$ into S_2
  end
  for $k = 1, \ldots, \texttt{length}(S_2)$ ▷Integrate $\boldsymbol{\omega}$ with time steps given by S_2
    Advance from $S_{2,k}$ to $S_{2,k+1}$ according to Eq. (9)
  end
  Attempt the step with h and $\boldsymbol{\omega}_{c_j}$ to get $\mathbf{x}_{tmp}$ according to Eq. (5) and (7)
  Estimate the error $\epsilon_{\mathbf{x}}$ and the acceptable error $\tau = \mathrm{tol} \cdot \|\mathbf{x}\|_\infty$
  Update $q = 0.9\left(\tau/\epsilon_{\mathbf{x}}\right)^{1/8}$
  if $\epsilon_{\mathbf{x}} \leq \tau$ ▷Accept the step
    Update $t = t + h$, and $\mathbf{x} = \mathbf{x}_{tmp}$
    Update $c = \min\left(h_{\max}, qh\right), h = \min(c, t_f - t)$
    Set $S = S_1$, and empty S_1 and S_2
    for $k = 1, \ldots, \texttt{length}(S)$
      Set $P = S_k$
      Update $P_1 = (P_1 - 1)q$ ▷Scale for the new time
      if $P_1 < 1$ ▷If below 1 is a repopped time
        Push P onto S_2
      else
        if S_1 *is empty* ▷Switching from S_2 to S_1
          Pop the bottom of $S_{2,\mathrm{end}}$ as Q
          Let $\Delta W_{tmp} \sim \mathcal{N}\left(\frac{1-Q_1}{P_1-Q_1} Q_2, \frac{(1-Q_1)(P_1-1)}{P_1-Q_1} h\right)$
          Push $(Q_1, \Delta W_{tmp})$ onto S_2
          Push $(1, Q_2 - \Delta W_{tmp})$ at the bottom of S_1
        else
          Push P onto S_1
        end
      end
    end
    if S_1 *is empty* ▷If there is no time after 1, extend S_2
      if S_2 *is empty* ▷If there is no repopped time, create S_2 from scratch
        Set the time vector $t_{EM} = (0, h_{EM}, \ldots, Nh_{EM})/h$ with $N = h/h_{EM}$
      else ▷Or from the last point
        Set $P = S_{end}$
        Set the time vector $t_{EM} = (1 - P_1, h_{EM}, \ldots, Nh_{EM})/h$ with $N = (h - P_1)/h_{EM}$
      end
      Set $\Delta W \sim \mathcal{N}_N(0, h_{EM})$
      Push $(t_{EM}, \Delta W)$ onto S_2
    end
  else ▷Reject the step
    Set $h = qh$ ▷Scale for the new time
    Merge S_1 and S_2 in S and empty them
    while S *is not empty*
      Pop the top of S as P
      Update $P_1 = P_1 q$
      if $P_1 < 1$ ▷If below 1 is a repopped time
        Push P onto S_2
      else
        Push P onto S_1
      end
    end
    Pop the top of S_2 as Q and the bottom of S_1 as P ▷Create the switch point from S_2 to S_1
    Let $\Delta W_{tmp} \sim \mathcal{N}\left(\frac{1-Q_1}{P_1-Q_1} Q_2, \frac{(1-Q_1)(P_1-1)}{P_1-Q_1} h\right)$
    Push $(Q_1, \Delta W_{tmp})$ onto S_2
    Push $(1, Q_2 - \Delta W_{tmp})$ at the bottom of S_1
  end
end

uncertainty associated to unmodeled accelerations, cosidered as a Gauss–Markov process[24]

$$\begin{bmatrix}\dot{\mathbf{r}}\\ \dot{\mathbf{v}}\end{bmatrix} = \begin{bmatrix}\mathbf{v}\\ -\mu\frac{\mathbf{r}}{\|\mathbf{r}\|^3} + \boldsymbol{\omega}_t\end{bmatrix} \tag{13}$$

and

$$\mathrm{d}\boldsymbol{\omega}_t = -\frac{1}{\tau}\boldsymbol{\omega}_t\mathrm{d}t + \sigma I_2 \mathrm{d}W_t \tag{14}$$

with I_2 being the identity matrix, and relevant parameters listed in Table 1.

Table 1: Perturbed two-body dynamics parameters.

Parameter	Symbol	Value
Sun gravitational parameter	μ	$1.327 \times 10^{11}\,\mathrm{km}^3/\mathrm{s}^2$
Noise correlation time	τ	$1\,\mathrm{d}$
Noise standard deviation	σ	$10^{-10}\,\mathrm{km}/\mathrm{s}^2$

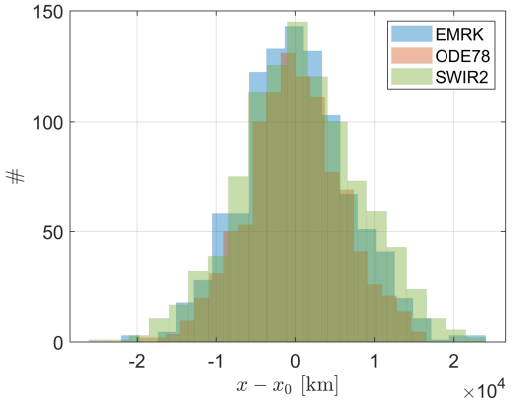
Solutions of the EMRK are compared against two different schemes: 1) a 1.5-order stochastic scheme by Rossler[14] (labeled SRIW2), exploiting its adaptive-step implementation in Julia,[25] and 2) a standard Runge-Kutta(8)7 scheme (labeled ODE78). For the ODE78 case, the process noise is computed iteratively a-priori[20] in some prescribed points, meaning that the values

$$\boldsymbol{\omega}_{k+1} = \boldsymbol{\omega}_k e^{-\beta(t_{k+1}-t_k)} + \mathbf{u}_k\sqrt{\frac{1}{2\beta}\left(1 - e^{-2\beta(t_{k+1}-t_k)}\right)} \tag{15}$$

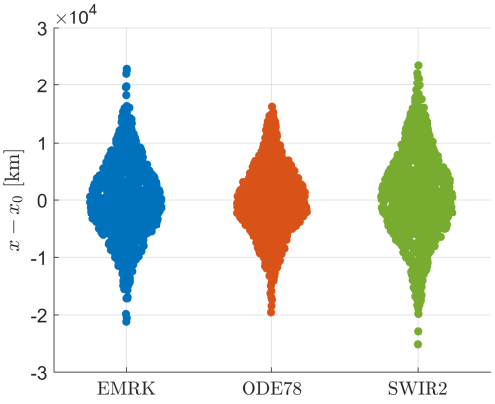
with $k = 0, \ldots, N$, and $\mathbf{u}_k \sim \mathcal{N}(0, 2\beta\sigma^2)$. These values are later interpolated during the integration runtime at the time requested by the integrator, exploiting the same formula.
Both accuracy and efficiency have been tested by running a Monte Carlo simulation with 1000 samples of an Hohmann transfer from the Earth to Mars.
Figure 3 shows the statistics of the position. As shown both in the histograms and the swarm chart, all the schemes are able to retrieve the same statistics for the transfer. A two-way Kolmogorov-Smirnov test has been performed to confirm this result. Since SWIR2 has been proven to be effective in retrieving the correct statistics of a generic SDE,[16] it is possible to conclude that EMRK is accurate to solve RODE problems under the assumptions made in Section .
Figure 4 shows the statistics of the computational time for EMRK and ODE78 on a Intel i7@2.9 GHz with 16 GB RAM. SWIR2 has not been shown since it is implemented in Julia and the comparison of its computational time with the other methods would not have been significant. However, for completeness's sake, it was about 1.5 times slower than EMRK. It can be inferred from the box plot that EMRK is able to integrate a single transfer in about $1\,\mathrm{s}$, while ODE78 20 times more.

CONCLUSIONS

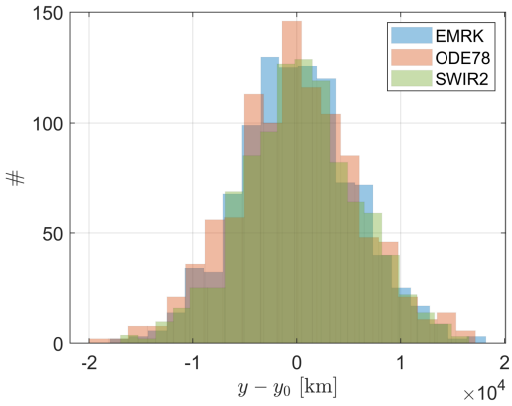
In this work, a hybrid stochastic-deterministic integrator, labeled Euler-Maruyama-Runge-Kutta, has been presented. It is devised to solve random ordinary differential equations characterized by simple stochastic dynamics and requiring moderate-to-high tolerances, exploiting a modified RSwM algorithm to allow the step-size and increase the efficiency of the method.
EMRK is able to provide solutions that are statistically correct, while reducing of at least one order of magnitude the computational time with respect to state-of-the-art SDE solvers. Its implementation will be beneficial in running massive Monte Carlo simulations for astrodynamics application, such as risk assessments, end-of-life evaluation and dispersion analysis.



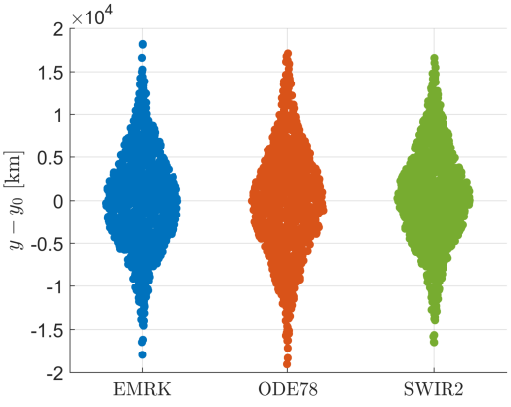
(a) Histogram for the x-component of the state



(b) Swarm chart for the x-component of the state



(c) Histogram for the y-component of the state



(d) Swarm chart for the y-component of the state

Figure 3: Relevant statistics for the uncertain 2-body problem.

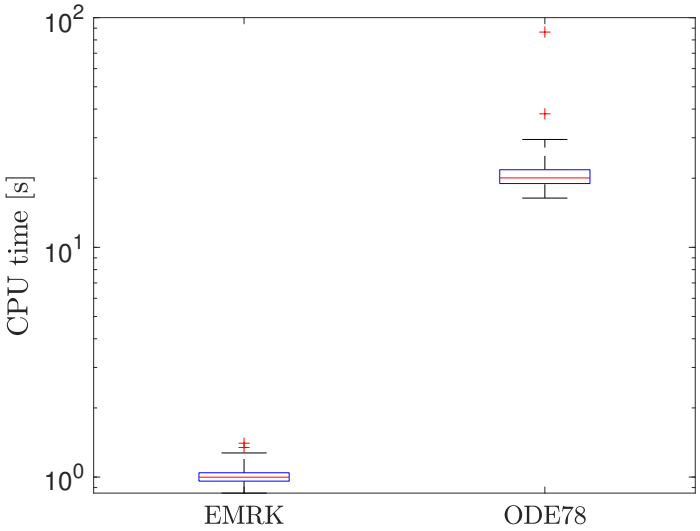


Figure 4: Computational time statistics for the uncertain 2-body problem.

ACKNOWLEDGMENT

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 864697).

REFERENCES

[1] W. Fehse, *Automated Rendezvous and Docking of Spacecraft*, Vol. 16, pp. 79–80. Cambridge University Press, 2003, 10.1017/CBO9780511543388.

[2] Y.-z. Luo and Z. Yang, "A review of uncertainty propagation in orbital mechanics," *Progress in Aerospace Sciences*, Vol. 89, 2017, pp. 23–39, 10.1016/j.paerosci.2016.12.002.

[3] D. A. Dei Tos, M. Rasotto, F. Renk, and F. Topputo, "LISA Pathfinder mission extension: A feasibility analysis," *Advances in Space Research*, Vol. 63, No. 12, 2019, pp. 3863–3883, 10.1016/j.asr.2019.02.035.

[4] C. Bottiglieri, F. Piccolo, C. Giordano, and F. Topputo, "Applied Trajectory Design for CubeSat Missions to Asteroids," *Journal of Spacecraft and Rockets*, 2022. Submitted.

[5] A. Poghosyan and A. Golkar, "CubeSat evolution: Analyzing CubeSat capabilities for conducting science missions," *Progress in Aerospace Sciences*, Vol. 88, 2017, pp. 59–83, 10.1016/j.paerosci.2016.11.002.

[6] R. Walker, D. Binns, C. Bramanti, M. Casasco, P. Concari, D. Izzo, D. Feili, P. Fernandez, J. G. Fernandez, P. Hager, *et al.*, "Deep-space CubeSats: thinking inside the box," *Astronomy & Geophysics*, Vol. 59, No. 5, 2018, pp. 24–30, 10.1093/astrogeo/aty232.

[7] P. E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*, ch. 10–17. Berlin, Germany: Springer-Verlag, 1992, 10.1007/978-3-662-12616-5.

[8] G. N. Milstein and M. V. Tretyakov, *Stochastic Numerics for Mathematical Physics*. Berlin, Germany: Springer-Verlag, 2004, 10.1007/978-3-662-10063-9.

[9] K. Burrage and P. Burrage, "General order conditions for stochastic Runge-Kutta methods for both commuting and non-commuting stochastic ordinary differential equation systems," *Applied Numerical Mathematics*, Vol. 28, No. 2-4, 1998, pp. 161–177, 10.1016/S0168-9274(98)00042-7.

[10] Y. Komori, "Weak second-order stochastic Runge–Kutta methods for non-commutative stochastic differential equations," *Journal of computational and applied mathematics*, Vol. 206, No. 1, 2007, pp. 158–173, 10.1016/j.cam.2006.06.006.

[11] Y. Komori, "Weak order stochastic Runge–Kutta methods for commutative stochastic differential equations," *Journal of computational and applied mathematics*, Vol. 203, No. 1, 2007, pp. 57–79, 10.1016/j.cam.2006.03.010.

[12] A. Rößler, "Second order Runge–Kutta methods for Itô stochastic differential equations," *SIAM Journal on Numerical Analysis*, Vol. 47, No. 3, 2009, pp. 1713–1738, 10.1137/060673308.

[13] N. J. Newton, "Asymptotically efficient Runge–Kutta methods for a class of Itô and Stratonovich equations," *SIAM Journal on Applied Mathematics*, Vol. 51, No. 2, 1991, pp. 542–567, 10.1137/0151028.

[14] A. Rößler, "Runge–Kutta methods for the strong approximation of solutions of stochastic differential equations," *SIAM Journal on Numerical Analysis*, Vol. 48, No. 3, 2010, pp. 922–952.

[15] K. Burrage and P. M. Burrage, "High strong order explicit Runge-Kutta methods for stochastic ordinary differential equations," *Applied Numerical Mathematics*, Vol. 22, No. 1-3, 1996, pp. 81–101, 10.1016/S0168-9274(96)00027-X.

[16] C. Rackauckas and Q. Nie, "Adaptive methods for stochastic differential equations via natural embeddings and rejection sampling with memory," *Discrete and continuous dynamical systems. Series B*, Vol. 22, No. 7, 2017, p. 2731.

[17] T. Tripura, A. Gogoi, and B. Hazra, "An Itô–Taylor weak 3.0 method for stochastic dynamics of nonlinear systems," *Applied Mathematical Modelling*, Vol. 86, 2020, pp. 115–141, 10.1016/j.apm.2020.05.014.

[18] M. Wiktorsson, "Joint characteristic function and simultaneous simulation of iterated Itô integrals for multiple independent Brownian motions," *The Annals of Applied Probability*, Vol. 11, No. 2, 2001, pp. 470–487, 10.1214/aoap/1015345301.

[19] P. S. Maybeck, *Stochastic Models, Estimation, and Control*, Vol. 3, pp. 180–185. New York, NY: Academic Press, 1982.

[20] M. Deserno, "How to generate exponentially correlated Gaussian random numbers," *Department of Chemistry and Biochemistry UCLA, USA*, 2002.

[21] J. R. Dormand and P. J. Prince, "A family of embedded Runge-Kutta formulae," *Journal of computational and applied mathematics*, Vol. 6, No. 1, 1980, pp. 19–26.

[22] J. H. Verner, "Numerically optimal Runge–Kutta pairs with interpolants," *Numerical Algorithms*, Vol. 53, No. 2, 2010, pp. 383–396.

[23] B. Oksendal, *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

[24] B. Schutz, B. Tapley, and G. H. Born, *Statistical orbit determination*, ch. 5. Oxford, UK: Elsevier, 2004, https://doi.org/10.1016/B978-0-12-683630-1.X5019-X.

[25] C. Rackauckas and Q. Nie, "DifferentialEquations.jl–a performant and feature-rich ecosystem for solving differential equations in julia," *Journal of open research software*, Vol. 5, No. 1, 2017.