Fast and reliable computation of the instantaneous orbital collision probability

Matthieu Masson, Denis Arzelier, Florent Bréhard, Mioara Joldes, Bruno Salvy

Fast and reliable computation of the instantaneous orbital collision probability

Matthieu Masson[1], Denis Arzelier[1], Florent Bréhard[2], Mioara Joldes[1], and Bruno Salvy[3]

[1]LAAS-CNRS, CNRS, Univ. de Toulouse, Toulouse, France.
[2]Crystal Lab., CNRS, Univ. de Lille, Lille, France.
[3]INRIA, LIP, ENS-Lyon, Lyon, France.

Abstract

Due to the increasing number of objects in Low Earth orbit, the fast and reliable estimation of the collision risk is an important challenge for spacecraft owners/operators. Among the available risk indicators, we focus on computing the instantaneous probability of collision, which can be modeled as the integral of a three-dimensional Gaussian probability density function over a Euclidean ball. We propose an efficient and accurate method for evaluating this integral. It is based on the combination of two complementary strategies. For the first one, convergent series and numerical error bounds are computed. These bounds provide a tradeoff between the accuracy needed and the number of terms to compute. The second one, using divergent series, approximates the value of the integral with a good accuracy in most cases with only a few terms computed. Based on those two methods, a hybrid algorithm is tested on cases borrowed from the literature and compared against existing methods. Several numerical results and comparisons confirm both the efficiency and robustness of our approach.

Nomenclature

$\mathcal{P}_{inst}$	=	Instantaneous probability of collision
$\mathcal{L}_f$	=	Laplace transform of function f
$\mathcal{B}(M, \delta)$	=	Closed ball of center M and radius δ
$\mathrm{Re}(z)$	=	Real part of complex number z
$\mathrm{Im}(z)$	=	Imaginary part of complex number z
$\lvert A \rvert$	=	Determinant of the square matrix A
R	=	Combined harbody ball radius, m
Σ	=	Covariance matrix
σ	=	Standard deviation, m
$\mathrm{diag}(\sigma_1, \sigma_2, \sigma_3)$	=	Diagonal matrix with σ_i on the diagonal
$\Pr\{\cdot\}$	=	Probability of event "$\cdot$"
$X \triangleright \mathcal{N}(\mu, \sigma)$	=	Normal random variable with mean μ and standard deviation σ
$\Gamma(\cdot)$	=	Gamma function
$\lceil \cdot \rceil$	=	the ceiling function which maps a real number to the smallest following integer
$n!!$	=	double factorial of a number n, $n!! = n(n-2)(n-4)\cdots$
$O(\cdot)$	=	The usual O-symbol as in $f_n = O(n^\alpha)$, meaning $f_n \leq cn^\alpha$, for $0 < c < \infty$, as $n \to \infty$
$\sim$	=	asymptotically equivalent
χ^2	=	Chi-square distribution

Subscripts

p	=	Primary object
s	=	Secondary object

1 Introduction

Due to an increasing congestion of objects (assets and debris included) orbiting the Earth, space conjunction assessment has become a fundamental task for space agencies and operators of the field. Based on the information obtained from tracking systems (radars, telescopes for instance), the Operators/Owners (O/O) of active on-orbit satellites are able to assess the risk of a possible collision between two objects.

Because of the uncertain nature of the data, the parameter most commonly used for evaluating this risk is the so-called probability of collision between Resident Space Objects (RSOs). When its value exceeds some tolerance threshold, a maneuver is performed. As a consequence, sustained efforts have been carried on during the past three decades, in order to properly characterize and accurately compute this parameter.

Nevertheless, the complexity of computing the collision probability is twofold. First, a clear and sound framework for the encounter has to be defined. This implies setting reasonable assumptions about the statistics (nature of the distributions involved, independence and correlation characteristics, time dependence of the distributions, etc.) of the uncertainty, dura-

tion of the encounter, uncertainty affecting the relative velocities (see [9], [8], [17] for examples of various possible assumptions). Secondly, depending on the context of the encounter, the parameter quantifying the risk of collision needs to be both modeled precisely mathematically as well as numerically evaluated fast and reliably . Two main paradigms — the short-term and the long-term encounters — have been widely accepted and implemented in the field of in-orbit collision risk assessment [7]. In both cases, the collision probability can be modeled as cumulative (the overall probability during a given time range) or instantaneous (at a single time instant). Significant efforts have been devoted to quantifying the cumulative collision probabilities for short-term [2], [30], [3], [7], [33], [14] and long-term encounters [31], [7], [4], [9], [25] as this formulation better captures the overall risk on a time range.

Alternatively, the so-called instantaneous collision probability was defined as an integral computing the probability for any 3-D Gaussian random vector to lie in a Euclidean ball of given radius. This indicator can be used to analyze and validate a maneuver plan, by checking that the collision risk remains low at any given time after an avoidance maneuver. In addition, it is a lower bound to the cumulative collision probability [4], [29]. Its evaluation is thus required to be both very efficient and reliable at each time instant but surprisingly enough, the dedicated literature is comparatively poor.

The available methods may be divided into two categories depending on whether the stress is particularly placed on the computation of the integral itself or on the propagation method used to build the data. The first class of methods mainly gathers the two seminal methods proposed in Chan's book [7], and the more recent equivalent volume cuboid approximation [39]. While Chan's approximation methods are based either on the notion of equivalent area, or on approximating the distributions, the method of [39] uses another equivalent volume formulation, with a single cuboid approximating the hard body shape.

The second class is mainly composed of two works, [20] and [1]. In [20], polynomial chaos expansions are mixed with a Monte Carlo simulation to evaluate the instantaneous collision probability. Adurthi and Singla make use of the conjugate unscented transform, in [1], to estimate the first statistical moments of the Probability Distribution Function (PDF) of the relative position vector which is then approximated via the maximum entropy method. A direct numerical evaluation of the 3-D integral of this approximated PDF completes the picture. All these methods either lack a sufficient accuracy due to the rough approximations involved, or require intensive and expensive computations due to numerical integration or Monte Carlo simulations.

In this study, we propose a new algorithm for the computation of the instantaneous collision probability between two orbiting objects of spherical

shape. Belonging to the first class discussed before, our algorithm focuses on the reliable and efficient evaluation of a 3-D integral of a Gaussian density function on a disk. This mathematical formulation is recalled in Section 2. Our strategy is based on combining two new complementary methods. On the one hand, we extend the 2-D method presented in [33] to the 3-D setting – detailed in Section 3.1. This provides an approximation by a convergent series with positive terms, as well as truncation error bounds. On the other hand, the cases for which conjunction data is ill-conditioned (which increases the number of terms required by the first method), are handled with a saddle-point technique, explained in Section 3.2. The resulting hybrid algorithm is briefly analyzed in Section 4 and then validated on several examples in Section 5. Further comparisons against the existing methods confirm both the efficiency and robustness of our approach, which is highlighted in the concluding Section 6.

2 Instantaneous collision probability between two spherical objects

We first recall the mandatory assumptions defining the general framework in which the instantaneous probability of collision may be soundly defined. The initial (at epoch) relative position vector is a Gaussian independent random vector $X_r \rhd \mathcal{N}(\mu, \Sigma)$ and each object is approximated by a spherical geometrical shape of respective radius R_p and R_s. As it is common in probability theory, a random variable is denoted by a capital letter, and the associated lowercase letter is then defined to assign a value to the random variable.

2.1 Mathematical formulation

In his book [7, Chapter 13], K. Chan has been the first one to formulate the problem of the computation of the instantaneous collision probability as the computation of the integral of a three-dimensional (3-D) Gaussian density – representing the objects' relative position uncertainty – over a Euclidean ball of given combined hard-body radius R (defined as the sum of the respective radii of the two objects i.e. $R = R_p + R_s$):

$$\mathcal{P}_{inst} = \frac{1}{(2\pi)^{3/2}|\Sigma|^{1/2}} \iiint_{\mathcal{B}(0,R)} \exp\left(-\frac{1}{2}(r-\mu)^T\Sigma^{-1}(r-\mu)\right) \mathrm{d}r, \qquad (1)$$

where r is the position vector of the secondary object given in a reference frame whose origin is located at the primary object and Σ is the combined

covariance matrix describing the relative position uncertainty (see Figure 1 for an illustration of the different parameters). Rotating the (r_1, r_2, r_3) axes

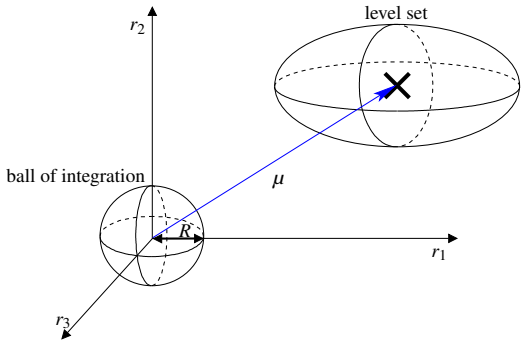


Figure 1: Illustration of the parameters defining the instantaneous collision probability.

to the principal axes (x_1, x_2, x_3) of the covariance matrix Σ, we get that:

$$\mathcal{P}_{inst} = \frac{1}{(2\pi)^{3/2} \prod_{i=1}^{3} \sigma_i} \iiint_{\mathcal{B}(0,R)} \exp\left(-\frac{1}{2} \sum_{i=1}^{3} \frac{(x_i - m_i)^2}{\sigma_i^2} \right) \mathrm{d}x_1 \mathrm{d}x_2 \mathrm{d}x_3, \qquad (2)$$

where the integrand is a 3-D Gaussian function and the domain is a ball of given radius R representing the hardbody radius. To the best of our knowledge, there is no known closed-form solution to the problem of the computation of the integral (2) (see also the account presented in [33] for the 2-D case). The aim is to evaluate the integral (2) which may be written as a function g of the square of the hardbody radius, $\xi = R^2$:

$$\begin{aligned} &g : \mathbb{R}^+ \mapsto \mathbb{R}^+, \\ &g(\xi) = g(R^2) = \mathcal{P}_{inst}. \end{aligned} \qquad (3)$$

This function g is in fact the cumulative density function (cdf) of the random variable $\Xi = X_1^2 + X_2^2 + X_3^2$, i.e. $g(\xi) = \Pr\{\Xi \leq \xi\}$ where $X_1 \rhd \mathcal{N}(m_1, \sigma_1^2)$, $X_2 \rhd \mathcal{N}(m_2, \sigma_2^2)$ and $X_3 \rhd \mathcal{N}(m_3, \sigma_3^2)$ are independent normal random variables. Within this perspective, it is important to notice that various approaches of the literature have been used to derive expansions of central and non-central quadratic forms in normal variables using Laplace transform of the cdf (see [19, Chapter29], [28, Chapter 4], [23] and [24] for a unified and systematic presentation of these approaches).

2.2 Laplace transform of the cumulative density function

A closed-form expression of the Laplace transform $\mathcal{L}_g$ of the function g may be easily obtained as:

$$\mathcal{L}_g(z) = \int_0^{+\infty} \exp(-z\xi) g(\xi) \mathrm{d}\xi = \frac{\exp\left(-z \sum_{i=1}^{3} \frac{m_i^2}{2z\sigma_i^2+1}\right)}{z \prod_{i=1}^{3} \sqrt{2z\sigma_i^2+1}}, \text{ for all } \mathrm{Re}(z) \geq 0. \tag{4}$$

Details of this derivation are available in [33] and are based on the ones given in [27]. Note that formula (4) for the Laplace transform of the cumulative function of Ξ may easily be recovered from the Laplace transform of the distribution function of Ξ [28, Chapter 4, p. 92].

At this point, two main strategies for obtaining a numerical approximation of the integral (2) may be considered. On the one hand, the Laplace transform (4) is expanded in a Laurent series on which a term-by-term application of the inverse Laplace transform provides a convergent power series expansion. On the other hand, the inverse Laplace transform formula to (4) leads to a complex-variable integral, which is evaluated by the so-called saddle-point method [6, Chap. 6].

3 Series expansions for the computation of instantaneous collision probability

3.1 A convergent series expansion using D-finiteness

As mentioned in the reference [33] (see also [5]), a direct expansion of the function g as formulated in [14] or in [23] and [24] for power series suffers from numerical issues owing to the fact that the computation in finite precision arithmetic of such a partial sum is prone to cancellation [15] (not all the coefficients of the series obtained are positive). Indeed, the sum in finite precision arithmetic of consecutive terms that are close in magnitude, but of different signs, contains very few correct significant digits. This makes the power series evaluation practical only for small values of $\xi = R^2$. Instead of expanding directly (3), it has therefore been proposed in the reference [33] to expand the function $\psi \cdot g$ where $\psi : \ R^2 \mapsto \exp(pR^2)$ is a so-called preconditionner, in order to remedy these observed numerical shortcomings. Some insights about the right choice for the parameter p are given in [34].

Here, assuming without loss of generality that $\sigma_1 \leq \sigma_2 \leq \sigma_3$, a good choice is $p = 1/(2\sigma_1^2)$. The function (3) is finally obtained as a simple product between an exponential term and a convergent power series with positive coefficients. As a very interesting by-product, it is worth mentioning that simple closed-form bounds on the truncation error may also be derived. The derivations required for obtaining the series expansion for (3) where all coefficients are positive may be summarized as follows:

- Compute the closed-form of the Laplace transform $h(z) = \mathcal{L}_{\tilde{g}}(z) = \mathcal{L}_g(z-p)$, of the preconditioned function $\tilde{g} = \psi \cdot g$ of the variable $\xi = R^2$, for all z such that $\mathrm{Re}(z) > 0$;
- Derive the Linear Ordinary Differential Equation (LODE) with polynomial coefficients satisfied by h and the associated initial condition (h is a so-called differentially finite function [32]);
- After analyzing the singularities of $h(z)$, consider its Laurent series expansion $h(z) = \sum_{k=0}^{\infty} \alpha_k z^{-(k+5/2)}$;
- From the previous LODE, deduce the recurrence satisfied by the coefficients α_k;
- The term-by-term inverse Laplace transform of the expansion of $h(z)$ provides the power series expansion (5) of g after multiplication by the inverse of the preconditioner;

$$\mathcal{P}_{inst} = g(R) = \exp\left(-\frac{R^2}{2\sigma_1^2}\right) \sum_{k=0}^{+\infty} \frac{\alpha_k}{\Gamma(k+5/2)} R^{2k+3} = \exp\left(-pR^2\right) \sum_{k=0}^{\infty} c_k, \tag{5}$$

where Γ is the Gamma function and the sequence of coefficients $(\alpha_k)_{k\in\mathbb{N}}$, and respectively $(c_k)_{k\in\mathbb{N}}$, satisfy the following linear recurrence relations:

$$2(k+6)\alpha_{k+6} = \sum_{i=1}^{6} \left((k+6-i)q_i - \omega_i\right) \alpha_{k+6-i}, \tag{6}$$

$$2(k+6)c_{k+6} = \sum_{i=1}^{6} \frac{R^{2i}}{\prod_{j=0}^{i-1}(k+6+5/2-j)} \left((k+6-i)q_i - \omega_i\right) c_{k+6-i}, \tag{7}$$

with $q_1, \ldots q_6$, $\omega_1, \ldots, \omega_6$ depending only on the problem parameters and not on k and initial conditions $\alpha_0, \ldots, \alpha_5$, $c_0, \ldots, c_5$ defined directly in Algorithm 1 for the sake of conciseness.

Similar to the one used for the computation of the short-term collision probability presented in [33], the series in (5) is particularly appropriate for an efficient numerical computation of the instantaneous probability of collision $\mathcal{P}_{inst}$. Its key features are the numerical stability of the derived Algorithm 1 in most of the cases and the fact that any partial sum $\mathcal{P}_{inst}^n = \exp(-pR^2) \sum_{k=0}^{n-1} c_k$, provides a lower bound to the exact value of $\mathcal{P}_{inst}$, due to the positivity of the coefficients c_k (see proposition 2 in Appendix A). In addition, the truncation error is lower and upper bounded in the following proposition, whose proof is given in Appendix A for completeness.

Proposition 1 *The following truncation error bounds hold for* $T_n = \mathcal{P}_{inst} - \mathcal{P}_{inst}^n$*:*

- *for all* $n > 0$, $l_n \leq T_n$, *with*

$$l_n = \frac{\alpha_0 p^n R^{2n+3}}{\Gamma(n+5/2)} \exp(-pR^2), \tag{8}$$

or equivalently by replacing the value of α_0*:*

$$l_n = \frac{p^n R^{2n+3}}{2\sqrt{2}\sigma_1\sigma_2\sigma_3\Gamma(n+5/2)} \exp\left(-\frac{m_1^2}{2\sigma_1^2} - \frac{m_2^2}{2\sigma_2^2} - \frac{m_3^2}{2\sigma_3^2} - pR^2\right); \tag{9}$$

- *for all* $n \geq \lceil 2pR^2 \rceil$, *one has* $u_n \geq T_n$, *with*

$$u_n = \frac{\alpha_0 p^n R^{2n+3}}{2^n \Gamma(n+5/2)} \frac{4p}{\sqrt{\overline{\gamma}_3 \overline{\gamma}_2}} \exp\left(\frac{m_1^2 p}{2} + \frac{\theta_2}{2\overline{\gamma}_2} + \frac{\theta_3}{2\overline{\gamma}_3} - pR^2\right), \tag{10}$$

and $\overline{\gamma}_i = \dfrac{1}{2\sigma_1^2} + \dfrac{1}{2\sigma_i^2}$, $\theta_i = \dfrac{m_i^2}{2\sigma_i^4}$.

The bounds u_n and l_n may be used for estimating the number of terms required for evaluating the instantaneous collision probability with a guaranteed accuracy. In short, given an absolute error threshold δ, the objective is to get an estimation of a sufficient number of terms N to be included in the series in order to guarantee that the prespecified accuracy holds. After some tedious but straightforward derivations (see also the Appendix A for a more detailed proof), using equations (8) and (45) with Stirling's inequality

Algorithm 1 PROBACOMPUTATIONCONVERGENTSERIES($(\sigma_i)_{i=1}^3$, $(m_i)_{i=1}^3$,R,N)

Input: combined radius R, mean m_i and standard deviation $\sigma_1 \leq \sigma_2 \leq \sigma_3$, number of terms N in the convergent series.

Output: $\tilde{\mathcal{P}}_{inst}$ – truncated convergent series approximation of $\mathcal{P}_{inst}$.

$p = \frac{1}{2\sigma_1^2}$; $\gamma_2 = \dfrac{1}{2\sigma_2^2} - \dfrac{1}{2\sigma_2^2}$; $\gamma_3 = \dfrac{1}{2\sigma_1^2} - \dfrac{1}{2\sigma_3^2}$;$\gamma_a = \gamma_2 + \gamma_3$; $\gamma_m = \gamma_2\gamma_3$; $\theta_2 = \dfrac{m_2^2}{2\sigma_2^4}$; $\theta_3 = \dfrac{m_3^2}{2\sigma_3^4}$; $\theta_1 = \dfrac{m_1^2}{2\sigma_1^4}$;

$q_0 = -2$; $q_1 = 4\gamma_a + 2p$;$q_2 = -(2{\gamma_a}^2 + 4p\gamma_a + 4\gamma_m)$; $q_3 = (2{\gamma_a}^2 + 4\gamma_m)\, p + 4\gamma_a\gamma_m$; $q_4 = -4p\gamma_a\gamma_m - 2{\gamma_m}^2$;

$q_5 = 2p{\gamma_m}^2$; $q_6 = 0$;$f_0 = -\theta_3 - \theta_2 - \gamma_a$; $f_1 = 2\theta_3\gamma_2 + 2\theta_2\gamma_3 + {\gamma_a}^2 + 2\gamma_m$; $f_2 = -\theta_3{\gamma_2}^2 - \theta_2{\gamma_3}^2 - 3\gamma_a\gamma_m$;

$f_3 = 2{\gamma_m}^2$; $s_0 = f_0 - 2p$; $s_1 = (-f_0 + 4\gamma_a)p + f_1$; $s_2 = (-f_1 - 4\gamma_m - 2{\gamma_a}^2)p + f_2$; $s_3 = (-f_2 + 4\gamma_m\gamma_a)p + f_3$; $s_4 = -4\gamma_m^2 p$; $s_5 = 0$;

$\omega_i = \dfrac{\theta_1}{2} q_{i-1} + s_{i-1}$, for $i = 1, \ldots, 6$;

$C = \dfrac{e^{-\left(\frac{m_1^2}{2\sigma_1^2} + \frac{m_2^2}{2\sigma_2^2} + \frac{m_3^2}{2\sigma_3^2}\right)}}{2\sqrt{2}\sigma_1\sigma_2\sigma_3}$;

$\alpha_0 = C$; $\alpha_1 = \frac{-\omega_1\alpha_0}{2}$; $\alpha_2 = \frac{(q_1-\omega_1)\alpha_1 - \omega_2\alpha_0}{4}$;$\alpha_3 = \frac{(2q_1-\omega_1)\alpha_2 + (q_2-\omega_2)\alpha_1 - \omega_3\alpha_0}{6}$;

$\alpha_4 = \frac{(3q_1-\omega_1)\alpha_3 + (2q_2-\omega_2)\alpha_2 + (q_3-\omega_3)\alpha_1 - \omega_4\alpha_0}{8}$; $\alpha_5 = \frac{(4q_1-\omega_1)\alpha_4 + (3q_2-\omega_2)\alpha_3 + (2q_3-\omega_3)\alpha_2 + (q_4-\omega_4)\alpha_1 - \omega_5\alpha_0}{10}$;

$c_0 = \frac{4\alpha_0 R^3}{3\sqrt{\pi}}$; *Note that* $\Gamma(5/2) = 3\sqrt{\pi}/4$;

$c_i = \dfrac{4\alpha_i R^{2i+3}}{3\sqrt{\pi}\prod_{j=0}^{i-1}(j+5/2)}$ for $i = 1, \ldots, 5$;

for $k \leftarrow 0$ to $N - 7$ **do**

$$c_{k+6} = \frac{1}{2(k+6)} \sum_{i=1}^{6} \frac{R^{2i}}{\prod_{j=1}^{i}(k+6+5/2-j)} \left((k+6-i)q_i - \omega_i\right) c_{k+6-i}.$$

end for

$\tilde{\mathcal{P}}_{inst} \leftarrow \exp\left(-pR^2\right) \sum_{i=0}^{N-1} c_k$;

[13], we get, for all $\delta > 0$, that if $N \geq \max(N_0, N_1, N_2(\delta))$ then $u_N - l_N < \delta$, where

$$\begin{aligned} N_0 &= \lceil 2pR^2 \rceil, N_1 = \lceil epR^2 - 2 \rceil, \\ N_2(\delta) &= \lceil (\Omega + 1) \log_2 e - \log_2(\delta p^2 R \pi \sqrt{2} \sigma_1 \sigma_2 \sigma_3) \rceil, \end{aligned} \tag{11}$$

with $\Omega := \dfrac{\theta_2}{2\overline{\gamma}_2} + \dfrac{\theta_3}{2\overline{\gamma}_3} - pR^2 - \left(\frac{m_1^2}{4\sigma_1^2} + \frac{m_2^2}{2\sigma_2^2} + \frac{m_3^2}{2\sigma_3^2} \right)$.

Despite the undisputable advantages of the convergent series algorithm, it requires sometimes employing a very large number of terms in the series for particular parameter values. Indeed, the sequence $(c_k)_k$ is a convergent sequence but the c_k's will grow until $k \sim R^2/(2\sigma_1^2)$ and then will converge to 0. This is a known issue in the evaluation of certain power series and can also be observed from the previous estimation of the number of terms, which is in $O(pR^2)$. Roughly speaking, for some corner cases like e.g., a combined object radius of $R = 100$, when one of the standard deviations is very small with respect to the others, $\sigma_1 = 0.01$, $\sigma_2 = \sigma_3 = 1$ for instance, and the mean values are not too large, the order of magnitude of the number of terms will be 10^8. Although the complexity of Algorithm 1 is linear with respect to the number of terms, for even more extreme cases, one may argue that the number of terms can be too high for efficient evaluation. A different point of view is to consider directly (for these difficult cases) the inverse Laplace transform formula, which is a complex-variable integral that can be evaluated by the so-called saddle-point method [6, Chap. 6]. Note that a complementary approach consisting in evaluating by a numerical inversion a specific inversion formula of the associated characteristic function, may be found in [11], [12]. Here, we focus on the saddle-point method which allows for a very efficient evaluation, based on very few terms. The downside is that the obtained expansion is divergent and consequently, the accuracy of the evaluation cannot be improved arbitrarily by increasing the number of terms. However, few terms usually suffice to palliate the problematic cases of the approach based on the convergent series formulation of the instantaneous collision probability as demonstrated in the numerical examples section.

3.2 A divergent series expansion using the saddle-point method

The complementary algorithm we propose now approximates the instantaneous collision probability by truncating a divergent series of the inverse Laplace transform of $\mathcal{L}_g$. This series is obtained by a generalization of Laplace's method (see [37, Chap. II] and [6, Chap. 6]) to curvilinear integrals of the form (18) defined in the complex plane. This is known in the

literature under the name of *saddle-point method or steepest-descent method* [37, Sec. II.4]. The proposed method consists of two main steps, each of which are detailed below:

2- First, the function g can be expressed by the Bromwich integral formula as the inverse Laplace transform of $\mathcal{L}_g$ [26, Thm 7.2.1, Chap. 7]:

$$g(\xi) = \frac{1}{2i\pi}\int_{z_0-i\infty}^{z_0+i\infty} e^{\xi z}\mathcal{L}_g(z)\mathrm{d}z = \frac{1}{2i\pi}\int_{z_0-i\infty}^{z_0+i\infty} \frac{\exp\left(\xi z - z\sum_{i=1}^{3}\frac{m_i^2}{2\sigma_i^2 z+1}\right)}{z\prod_{i=1}^{3}\sqrt{2\sigma_i^2 z+1}}\mathrm{d}z. \tag{12}$$

For completeness, note that the (complex) singularities of the function $\mathcal{L}_g$ are a pole at 0 and 3 essential singularities at $-p_i = -1/(2\sigma_i^2)$. The integration path is a straight vertical line, $t : \mathbb{R} \mapsto z_0 + it$, intersecting the real positive axis at $z_0 > 0$, such that the singularities of $\mathcal{L}_g$ are all located to the left of the integration path in the complex plane, i.e. in the open left half-plane. The Cauchy integral theorem [26, Thm. 4.3.1, Chap. 4] ensures that the value of this integral does not depend on the choice of $z_0 > 0$. However, the numerical quality of this specific approximation method highly depends on the chosen integration path.

3- Considering this last issue, the saddle-point method consists in deforming the contour in a new contour in order for the Laplace's method to be applicable.

As a preliminary step, a short and simplified mathematical overview of the Laplace's method is given in the following.

3.2.1 Overview of Laplace's method.

Firstly, let us briefly describe the idea of Laplace, which allows for approximating real-variable integrals of the form:

$$f(\lambda) = \int_{-\infty}^{+\infty} e^{u_\lambda(x)}\mathrm{d}x, \tag{13}$$

where $u_\lambda : \mathbb{R} \to \mathbb{R}$ depends on a real parameter λ which tends to some given value $\lambda_0 \in [0, +\infty]$. The method is mainly based on the observation that when $\lambda \to \lambda_0$, the largest contribution for the computation of integral (13) comes from a small neighborhood of the point $x_0 \in \mathbb{R}$ where u_λ has a peak or a maximum ($u'_\lambda(x_0) = 0$, $u''_\lambda(x_0) < 0$):

$$f(\lambda) = \int_{-\infty}^{+\infty} e^{u_\lambda(x)} \mathrm{d}x \approx \int_{x_0-\epsilon}^{x_0+\epsilon} e^{u_\lambda(x)} \mathrm{d}x. \tag{14}$$

Therefore, one can approximate u_λ (in that neighborhood) by simpler functions like a parabola by expanding u_λ in a truncated Taylor series around x_0 at order 2 (the first non-vanishing term):

$$u_\lambda(x) \approx M(\lambda) - \alpha(\lambda)(x - x_0)^2, \quad M(\lambda) = u_\lambda(x_0), \quad \alpha(\lambda) = -\frac{u''_\lambda(x_0)}{2} > 0,$$
$$\text{for} \quad |x - x_0| \ll 1,$$

in which case the integrand is approximated by a Gaussian function, which can be integrated in closed-form when taking the lower and upper limits of the integral to infinity:

$$f(\lambda) \approx \int_{-\infty}^{+\infty} e^{M(\lambda) - \alpha(\lambda)(x-x_0)^2} \mathrm{d}x = e^{M(\lambda)} \sqrt{\frac{\pi}{\alpha(\lambda)}}. \tag{15}$$

Note that the values of M and α depend on λ and that the Laplace approximation above gives the dominant term of the asymptotic expansion of $f(\lambda)$ when $\lambda \to \lambda_0$. The determination of higher order terms requires the inclusion of more terms in the Taylor series of u_λ in the neighborhood of x_0. Still assuming that the function u_λ has a maximum at x_0 and hence $u'_\lambda(x_0) = 0$, with $u''_\lambda(x_0) < 0$, the following change of variables is defined:

$$u_\lambda(x) = u_\lambda(x_0) - \lambda s^2. \tag{16}$$

Using (16), the integral (13) becomes:

$$\begin{aligned} f(\lambda) &= \int_{-\infty}^{+\infty} e^{u_\lambda(x)} \mathrm{d}x = \int_{s(-\infty)}^{s(+\infty)} e^{u_\lambda(x_0)} e^{-\lambda s^2} \left(\frac{\mathrm{d}x}{\mathrm{d}s}\right) \mathrm{d}s \\ &\approx e^{u_\lambda(x_0)} \int_{-\tilde{\epsilon}}^{\tilde{\epsilon}} e^{-\lambda s^2} \left(\frac{\mathrm{d}x}{\mathrm{d}s}\right) \mathrm{d}s. \end{aligned} \tag{17}$$

Usually, one derives a power series expansion of x and $\left(\frac{\mathrm{d}x}{\mathrm{d}s}\right)$ in terms of the new variable s, followed by interchanging the order of the series summation and the integral. This may be rigorously justified by invoking Watson's lemma [37, Sec. I.6] and [6, Sec. 6.4] (including the extension of the lower and upper limits to infinity ($\tilde{\epsilon} \to \infty$), with a term-by-term integration which finally gives an asymptotic expansion of integral (13).

3.2.2 The saddle-point method.

First of all, the integral (12) to be computed is rewritten in a specific form, which is similar to (13), except for the integration path, which needs to be carefully considered:

$$g(\xi) = \frac{1}{2i\pi} \int_{z_0-i\infty}^{z_0+i\infty} e^{\varphi_\xi(z)} \mathrm{d}z, \tag{18}$$

with

$$\begin{aligned} \varphi_\xi(z) &= \xi z - z \sum_{i=1}^{3} \frac{m_i^2}{2\sigma_i^2 z + 1} - \log z - \frac{1}{2} \sum_{i=1}^{3} \log(2\sigma_i^2 z + 1), \\ &= \xi z + \sum_{i=1}^{3} \left(\frac{m_i^2 p_i^2}{z + p_i} - m_i^2 p_i \right) - \log z + \frac{1}{2} \sum_{i=1}^{3} \left(\log p_i - \log(z + p_i) \right). \end{aligned} \tag{19}$$

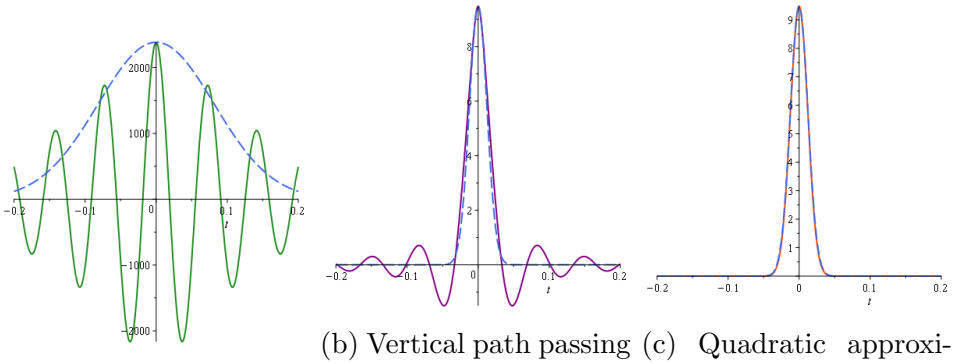
Choice of the integration path. Note that a point z_s satisfying $\varphi'_\xi(z_s) = 0$ is also a zero of the derivative of the function e^{φ_ξ}. In addition, such a z_s is a saddle-point for φ_ξ if φ_ξ is analytic in a complex domain including z_s and because its real part is a harmonic function. When possible (since the singularities of the function need to be taken into consideration), the Cauchy integral theorem is used and the contour is deformed to pass through such a saddle-point. Similarly to the Laplace's method, this choice is most effective on a contour where the value of the integral is very well approximated by considering only a small neighborhood of this point. This can be achieved if the contour chosen is (locally) one of steepest descent. Denoting the real and imaginary parts of the function φ_ξ as follows:

$$\varphi_\xi(z) = \mathrm{Re}\varphi_\xi(z) + i\mathrm{Im}\varphi_\xi(z) := u_\xi(z) + iv_\xi(z), \tag{20}$$

this is defined as a constant phase contour, on which $v_\xi(z)$ is constant and which is everywhere orthogonal to the contour lines of constant u_ξ.

In brief, the contour $\mathcal{C}$ must be (locally) deformed into a new path of integration $\mathcal{C}'$ such that:

- $\mathcal{C}'$ passes through one saddle-point, say z_s;
- $v_\xi(z)$ is constant on $\mathcal{C}'$, with $v_\xi(z) = v_\xi(z_s)$ for all $z \in \mathcal{C}'$.



(a) Vertical path passing through a point $z_0 \neq z_s$.

(b) Vertical path passing through the saddle point z_s.

(c) Quadratic approximation of the steepest descent path.

Figure 2: Integrand $e^{\varphi_\xi(z)}$ evaluated along three different paths and the corresponding Gaussian in dotted blue lines.

Such a deformation of $\mathcal{C}$ to $\mathcal{C}'$ is motivated by the fact that the oscillations of the integrand are eliminated on a path where $v_\xi(z)$ is constant. However, as for the Laplace method, since only a small neighborhood of the saddle point matters, a local knowledge of the structure of the lines of steepest descent may be sufficient (see Figure 2).

In equation (18), the initial path is a vertical line $t : \mathbb{R} \mapsto z_0 + it$, which crosses the positive real axis at $z_0 > 0$. One can prove that the function φ'_ξ:

$$\varphi'_\xi(z) = \xi - \sum_{i=1}^{3} \frac{m_i^2 p_i^2}{(z+p_i)^2} - \frac{1}{z} - \frac{1}{2}\sum_{i=1}^{3} \frac{1}{z+p_i},$$

has only one positive real root[1] , which is denoted by z_s in what follows and can be numerically found with the Newton method, for instance.

A first dominant term approximation. Since no singularities are present in the right half-plane, the vertical line can be shifted to pass through z_s, which together with the variable change $z = z_s + it$, gives:

$$g(\xi) = \frac{1}{2i\pi}\int_{z_s-i\infty}^{z_s+i\infty} e^{\varphi_\xi(z)}\mathrm{d}z \quad = \frac{1}{2\pi}\int_{-\infty}^{+\infty} e^{\varphi_\xi(z_s+it)}\mathrm{d}t. \tag{21}$$

Note that the integral (21) obtained after a deformation of the original integration contour is now defined on $\mathbb{R}$ even if the integrand is still a complex-valued function. The new integration variable t is real and since the value of

[1]It suffices to see that: *(i)* $\varphi_\xi(z) \sim -\log(z) \to +\infty$ when $z \to 0^+$; *(ii)* $\varphi_\xi(z) \sim \xi z \to +\infty$ when $z \to +\infty$; *(iii)* $\varphi''_\xi(z) > 0$ for all $z > 0$.

$g(\xi)$ is real by definition, we are interested only in the real part of $e^{\varphi_\xi(z_s+it)}$ (the contribution of the imaginary part will cancel), for which we can apply the Laplace's method. Expanding the function φ_ξ in Taylor series around z_s, one has

$$\varphi_\xi(z_s + it) \approx \varphi_\xi(z_s) + \frac{\varphi''_\xi(z_s)}{2}(it)^2 := a_0 - a_2 t^2, \quad \text{with } a_2 > 0, \tag{22}$$

and equation (15) directly gives:

$$g(\xi) \approx \frac{e^{a_0}}{2\sqrt{\pi a_2}}. \tag{23}$$

However, this classical Laplace approximation, given in equation (23), may not be sufficiently precise, in which case, a more involved change of variables and/or more coefficients in the Taylor expansion of φ_ξ around z_s need to be considered.

Approximation via a divergent series. Following the steps outlined in subsection 3.2.1 for obtaining higher order terms of the expansion that will play the role of correction terms to the previous approximation, a different change of variables is proposed:

$$\varphi_\xi(z_s + z) = a_0 + a_2(iw)^2, \tag{24}$$

that is, the function $\varphi_\xi(z_s + z)$ exactly matches a parabola with $a_0 = \varphi_\xi(z_s)$, $a_2 = \varphi''_\xi(z_s)/2 > 0$ and $w \in \mathbb{R}$. Along the path of steepest descent passing through the saddle point z_s, $\varphi_\xi(z_s + z) - \varphi_\xi(z_s)$ is real so that one may apply Laplace method on the transformed integral, now parameterized by the new real variable w. Usually, such a change of variables is valid only in a neighborhood of z_s and not on the whole integration path (since the value of the integral is concentrated around z_s this is not an issue in practice). So the overall strategy is:

- The vertical line $t : \mathbb{R} \mapsto z_s + it$, which is the integration path in equation (21) is deformed to locally match a constant phase contour passing through the saddle point z_s, which makes the change of variables in eq. (24) locally bijective for real w; note that in practice, one does not need to compute an exact formula for this new path.

- After the variables change, the limits of integration in w are formally extended to $-\infty$ and $+\infty$;

- The integral (18) is then approximated by:

$$g(\xi) \approx \frac{1}{2i\pi} \int_{-\infty}^{+\infty} e^{a_0 - a_2 w^2} \left(\frac{\mathrm{d}z(w)}{\mathrm{d}w} \right) \mathrm{d}w, \tag{25}$$

with the remaining challenge of expressing z as a function of w in order to get an expansion of $\left(\frac{\mathrm{d}z(w)}{\mathrm{d}w} \right)$ in terms of w. While in various examples in the literature, the function φ_ξ is sufficiently simple to allow for the equation (24) to be solved in closed-form, this is not the case here. Nevertheless, the function φ_ξ may be expanded as a power series near the saddle point z_s:

$$\varphi_\xi(z_s + z) \approx a_0 + a_2 z^2 + a_3 z^3 + \ldots, \quad \text{with } a_n = \frac{\varphi_\xi^{(n)}(z_s)}{n!}, \tag{26}$$

and where the successive derivatives $\varphi_\xi^{(n)}$ can be obtained for $n \geq 2$, in closed-form as:

$$\varphi_\xi^{(n)}(z) = (-1)^n \left(n! \sum_{i=1}^{3} \frac{m_i^2 p_i^2}{(z+p_i)^{n+1}} + \frac{(n-1)!}{z^n} + \frac{(n-1)!}{2} \sum_{i=1}^{3} \frac{1}{(z+p_i)^n} \right). \tag{27}$$

Combining equations (24) and (26), one aims at writing also $z(w)$ as a power series:

$$z(w) = b_0 + b_1(iw) + b_2(iw)^2 + b_3(iw)^3 + \ldots, \tag{28}$$

for which the unknown coefficients b_i can be found by an iterative procedure, which will provide a so-called formal inverse truncated series.

We now focus on the algorithmic procedure for computing the coefficients b_i's, i.e. to obtain z as a function $z := z(w)$.

- The first two coefficients $b_0 = 0$ and $b_1 = 1$ are readily obtained by setting $z(w) = b_0 + b_1(iw)$ in (26) and identifying the first two coefficients with (24).

- To compute b_2, take $z(w) = iw + b_2(iw)^2$ and evaluate $\varphi_\xi(z_s + z(w))$ up to order 3:

$$\varphi_\xi(z_s + z(w)) = a_0 + a_2(iw)^2 + (2a_2 b_2 + a_3)(iw)^3 + O((iw)^4).$$

The coefficient b_2 is then chosen to cancel the third coefficient in the previous expansion:

$$b_2 = -\frac{a_3}{2a_2}. \tag{29}$$

- Similarly, for b_3, one obtains:

$$b_3 = \frac{-4a_2a_4 + 5a_3^2}{8a_2^2}.$$

- More generally, for computing the n-th coefficient b_n, firstly expand $\varphi_\xi(z_s + z(w))$ as a power series in w, and then by identifying each coefficient of w^k $(k = 0, \cdots, n+1)$ in the expression $\varphi_\xi(z_s + z(w)) = a_0 - a_2w^2$, one obtains a triangular system with the last equation being linear in the coefficient b_n.

Algorithm 2 summarizes this procedure, while Tables 6 and 7 in Appendix B list the coefficients a_n and the first few computed coefficients b_n. Note that if more coefficients in the expansion $z(w)$ are needed, then they can be straightforwardly computed by Algorithm 2. Note also that these coefficients could be computed more efficiently by applying Lagrange's formula [10, p. 125] or power series reversion by Newton iterations [22, Sec. 4.7]. In practice however, only a few terms are needed in and Algorithm 2 is sufficient for this purpose.

Algorithm 2 CoefficientsBn(N, $(a_n)_{n=2}^N$)

Input: Coefficients a_n $(0 \le n \le N)$ of the formal series $\varphi(z_s + z) = a_0 + a_2z^2 + \cdots + a_Nz^N + \ldots$ to be inverted.

Output: Coefficients b_n of $z(w) = b_1(iw) + b_2(iw)^2 + \cdots + b_{N-1}(iw)^{N-1}$ such that $\varphi(z_s + z(w)) = a_0 + a_2(iw)^2 + O((iw)^{N+1})$.

1: $b_1 \leftarrow 1$.
2: $z(w) \leftarrow iw$.
3: **for** $n = 2$ **to** $N - 1$ **do**
4: $\quad z(w) \leftarrow z(w) + b_n(iw)^n$. ▷ *$b_n$ is an unknown variable*
5: $\quad c \leftarrow [(iw)^{n+1}]\varphi(z_s + z(w))$. ▷ *the coefficient before $(iw)^{n+1}$ in $\varphi(z_s + z(w))$*
6: $\quad b_n \leftarrow$ solution of $c = 0$.
7: **end for**
8: **Return** $b_1, \ldots, b_{N-1}$.

Computation of the final divergent series coefficients. The change of variables in (28) allows for rewriting (25) as:

$$g(\xi) \approx \frac{e^{a_0}}{2\pi} \int_{-\infty}^{\infty} e^{-a_2w^2} \sum_{n=0}^{\infty} (n+1)b_{n+1}(iw)^n \mathrm{d}w. \tag{30}$$

In general, the integral and the infinite sum cannot be inverted, since the series is not uniformly convergent, but one can nevertheless truncate the series and then evaluate term-by-term the corresponding integrals. A straightforward computation shows that the integrals corresponding to odd powers of w are zero (as expected, since the integral value is real). Computing the remaining even coefficients gives:

$$\frac{1}{2\pi}\int_{-\infty}^{+\infty} e^{-a_2 w^2}(iw)^{2n}\mathrm{d}w = \frac{(-1)^n(2n-1)!!}{\sqrt{\pi}2^{n+1}a_2^{n+1/2}}.$$

Thus, the final approximation for our integral (12) is:

$$g(\xi) \sim \frac{e^{a0}}{2\sqrt{\pi a_2}}\sum_{n=0}^{N} c_n, \quad \text{with} \quad c_n = (-1)^n\frac{(2n+1)!!}{(2a_2)^n}b_{2n+1}. \tag{31}$$

Table 8 in Appendix B gives the first five coefficients c_n in terms of the coefficients a_n and b_n, while Algorithm 3 summarizes the whole procedure. The sum that is obtained diverges when $N \to \infty$ and a finite number of terms of the series has to be chosen and used in approximating the integral (12). While the optimum number of terms (the minimum number of terms to get the better precision) is not known *a priori*, the first five coefficients (running Algorithm 3 with $N = 5$) contain sufficient precision in practice for a precise evaluation of the integral (12) in many cases, as will be seen in the numerical examples of the next section.

Algorithm 3 ProbaComputationDivergentSeries($(\sigma_i)_{i=1}^3$, $(m_i)_{i=1}^3$,R,N)

Input: combined radius R, mean m_i and standard deviation σ_i, number of terms N in the divergent series.

Output: $\tilde{\mathcal{P}}_{inst}$ – truncated divergent series approximation of $\mathcal{P}_{inst}$.

1: Numerically compute the saddle point $z_s > 0$, solution of:

$$R^2 - \sum_{i=1}^{3} \frac{m_i^2}{(2\sigma_i^2\lambda + 1)^2} - \sum_{i=1}^{3} \frac{\sigma_i^2}{2\sigma_i^2\lambda + 1} = 0.$$

2: Compute the coefficients a_n for $0 \leq n \leq 2N$:

$$a_0 = \xi z_s - z_s \sum_{i=1}^{3} \frac{m_i^2 p_i}{z_s + p_i} - \log z_s - \frac{1}{2} \sum_{i=1}^{3} \log\left(\frac{z_s}{p_i} + 1\right), \tag{32}$$

$$a_1 = \xi - \sum_{i=1}^{3} \frac{m_i^2 p_i^2}{(z_s + p_i)^2} - \frac{1}{z_s} - \frac{1}{2} \sum_{i=1}^{3} \frac{1}{z_s + p_i} = 0, \tag{33}$$

$$a_n = (-1)^n \left(\sum_{i=1}^{3} \frac{m_i^2 p_i^2}{(z_s + p_i)^{n+1}} + \frac{1}{n z_s^n} + \frac{1}{2n} \sum_{i=1}^{3} \frac{1}{(z_s + p_i)^n} \right), \quad n \geq 2. \tag{34}$$

3: Compute the coefficients b_n for $0 \leq n \leq 2N - 1$ and odd n with Algorithm 2. See also Table 7 for precomputed coefficients $b_1, \ldots, b_9$.

4: Compute the coefficients c_n for $0 \leq n \leq N - 1$ (see also Table 8 for precomputed coeffs $c_0, \ldots, c_4$):

$$c_n = (-1)^n \frac{(2n+1)!!}{(2a_2)^n} b_{2n+1}. \tag{35}$$

5: **Return** $\tilde{\mathcal{P}}_{inst} \leftarrow \dfrac{e^{a_0}}{2\sqrt{\pi a_2}}(c_0 + c_1 + \cdots + c_{N-1})$.

4 Hybrid algorithm

The overall strategy consists in switching between the convergent-series method [35], which was summarized in Algorithm 1 and the saddle-point alternative, given by Algorithm 3, based on the estimation of a sufficient number of terms required for the first method given in equation (11). The resulting pseudo-code is presented in Algorithm 4.

Algorithm 4 Computation of instantaneous probability of collision($(\sigma_i)_{i=1}^3$, $(m_i)_{i=1}^3$,R,δ).

Input: combined radius R, mean m_i and standard deviation $\sigma_1 \leq \sigma_2 \leq \sigma_3$, required accuracy δ.

Output: $\tilde{\mathcal{P}}_{inst}$ – approximation of $\mathcal{P}_{inst}$.

1: $N_{max} = 4000$; $N_{div} = 5$;

2: $p = \frac{1}{2\sigma_1^2}$; $\gamma_2 = \dfrac{1}{2\sigma_1^2} - \dfrac{1}{2\sigma_2^2}$; $\gamma_3 = \dfrac{1}{2\sigma_1^2} - \dfrac{1}{2\sigma_3^2}$;$\overline{\gamma}_2 = \dfrac{1}{2\sigma_1^2} + \dfrac{1}{2\sigma_2^2}$; $\overline{\gamma}_3 = \dfrac{1}{2\sigma_1^2} + \dfrac{1}{2\sigma_3^2}$;$\theta_2 = \dfrac{m_2^2}{2\sigma_2^4}$; $\theta_3 = \dfrac{m_3^2}{2\sigma_3^4}$;

3: $\Omega = \dfrac{\theta_2}{2\overline{\gamma}_2} + \dfrac{\theta_3}{2\overline{\gamma}_3} - pR^2 - \left(\frac{m_1^2}{4\sigma_1^2} + \frac{m_2^2}{2\sigma_2^2} + \frac{m_3^2}{2\sigma_3^2}\right)$;

4: $N_0 = \lceil 2pR^2 \rceil$; $N_1 = \lceil epR^2 - 2 \rceil$; $N_2(\delta) = \lceil (\Omega + 1)\log_2 e - \log_2(\delta p^2 R\pi\sqrt{2}\sigma_1\sigma_2\sigma_3) \rceil$;

5: $N_{conv} = \max(N_0, N_1, N_2(\delta))$;

6: **if** $N_{conv} \leq N_{max}$ **then**

7: $\tilde{\mathcal{P}}_{inst} \leftarrow$ Algorithm 1 (σ_x, σ_y, x_m, y_m, R, N_{conv});

8: **else**

9: $\tilde{\mathcal{P}}_{inst} \leftarrow$ Algorithm 3 (σ_x, σ_y, x_m, y_m, R, N_{div});

10: **end if**

11: **return** $\tilde{\mathcal{P}}_{inst}$.

More specifically, besides the usual input parameters given by the relative mean, relative covariance and hardbody radius, the user inputs also an absolute error parameter δ, which corresponds to the number of correct digits required (e.g., $\delta = 10^{-15}$ corresponds to 15 correct digits after the decimal dot). This is the value we chose in our experiments to allow for accurate results. Depending on the parameter δ and the corresponding input data, in line 5 of Algorithm 4, a sufficient number of terms N_{conv}, for Algorithm 1 to reach the desired accuracy, is computed. This directly affects the performance of Algorithm 1, which requires $O(N_{conv})$ basic arithmetic operations (addition, multiplication, division), plus a final evaluation of an exponential

function. To compensate for the computational burden when $N_{conv} > N_{max}$, one switches to the second alternative given by Algorithm 3, which is usually faster for $N_{div} = 5$, but cannot guarantee the final accuracy. So a trade-off is to be made, function on the maximum number of terms N_{max}. As previously discussed, N_{conv} is in the order of $\frac{R^2}{2\sigma_1^2}$ and our practical numerical experiments (detailed in the next section) suggest that, the majority of *well-behaved* cases will need less than few hundreds terms (for 15 digits of guaranteed precision). In our implementation, we propose to fix $N_{max} = 4000$ in line 2. This is a choice which favors accuracy over performance, and a different trade-off can be simply set by tuning this parameter.

5 Numerical examples

In order to demonstrate the performance of the Hybrid Algorithm 4, three different test cases borrowed from the references [4] and [38] are used to compare the results with the ones given by three alternative methods of the literature: the Equivalent Volume Cuboïd (EVC) method from [39] and the methods of Equivalent Volumes (EV) and of Approximating Distributions (AD) from Chan's book [7, Chap. 13]. In addition, Monte Carlo simulations are performed to compute the instantaneous collision probability on the given time span and we assume that this result is the reference in each case. The time of closest approach is set to $t = 0$. Every algorithm mentioned in this article is implemented in Matlab R2021b® and all computations were performed on a HP Dragonfly G3 laptop with an Intel i5 12th Gen processor and 16Go RAM except for the Monte-Carlo simulations with a sample size of 10^7 which were performed on a computing platform using 10Go RAM and a parallel pool of 10 workers (Matlab® Parallel Computing Toolbox). All details pertaining to the computation of $\mathcal{P}_{inst}$ by the various algorithms and to the uncertainty propagation are given in the next two subsections.

5.1 Uncertainty propagation

When computing the instantaneous collision probability, the propagation, from the epoch t_0 to the desired time t, of a given set of initial orbit uncertainty data related to each object (mean state vectors of positions and velocities and the associated covariance matrix) requires a particular care. Here, we use the Unscented Transform (UT) [21], possibly combined with Gaussian Mixture Models (GMM) when the time span of the propagation is too long. UT is known to be a valuable alternative to the usual and much simpler propagation based on linearization. For the following numerical ex-

amples, it is implemented using equation (3) from [21] with $\kappa = 0.5$ so that the sample mean and covariance are unweighted. Note also that the sigma points are computed again at each time step.

Still, propagating one single Gaussian distribution for each space object over long orbital nonlinear arcs might produce uncertainties which are not well represented by Gaussian distributions at time t [18]. For the third example, a Gaussian Mixture Model (GMM) was implemented based on corollary 2.3 from [18] and the Gauss-Hermite quadrature scheme from [16] [2] to split the initial Gaussian PDF of the primary and the secondary. The splitting can be tuned through two parameters: the order of the Gauss-Hermite quadrature [3] and the number d of eigendirections of the initial covariance matrix along which to perform a split. Following [18], we chose the d eigendirections in decreasing order of eigenvalues magnitude (the "d largest eigendirections"). Each component of each space object is then propagated using the UT without updating the weights in the GMM.

5.2 Algorithms for the computation of $\mathcal{P}_{inst}$

The main tuning parameters of our implementation of the different methods are given here in order for the presented results to be easily replicated if needed. First, Chan's algorithms are implemented following [7, Chap. 13]. In particular, the Equivalent Volume method is implemented using Matlab® function *integrate* with default values and the choice of the least aspect ratio rule, i.e. the direction of integration is such that the remaining two eigenvalues yield the least aspect ratio. Both EV and AD implementations were first validated on the numerical examples presented in [7, Chap. 13].

For the Monte-Carlo sample generation, we follow [4, Section III-D]. First, the random samples are generated based on the known PDFs for each space object at epoch. Secondly, each sample is then propagated in the Keplerian dynamics, by a numerical scheme from [36]. Finally, the probability of collision at time t is computed by counting the number of relative state vector samples at time t whose Euclidian norm is lower than the hard-body radius R.

[2] A Matlab® library is available at http://sparsegrids.de

[3] If k is the order, then the first k moments of the GMM and of the initial PDF are equal [18].

5.3 Two Satellites in Geosynchronus Equatorial Orbits (GEO)

The first numerical example is the test case 4 of reference [4], which was also analyzed in [39]. It describes two satellites in GEO with low relative velocity (around 0.02 m/s at TCA). The number of samples for Monte-Carlo simulation was set to 2×10^6 in compliance with the Dagum bound given in [4] while the time step was set to 10 min, for computational reasons. As in [4] and [39], $\mathcal{P}_{inst}$ has been computed from $t_0 = 0$ (TCA) to $t_f = 13200$ s (220 minutes). As mentioned in [4], the probability starts to accumulate 50 minutes after TCA. Since the time range is relatively small (less than one fifth of one orbital period for each space object), we have assumed that the initial density function of each object remains Gaussian for all time $t \in [t_0, t_f]$.

Firstly, the efficiency of our Algorithm 4 is analyzed, by fixing a target accuracy of $\delta = 10^{-15}$ and tuning the value of the parameter N_{max}, function of which either the method of convergent or respectively, divergent series is employed. The ratio of calls to either method is tested by running a total of 301 calls to Algorithm 4, on a uniform discretization of the time span chosen above. Table 1 summarizes both the ratio and corresponding timings for different choices of N_{max}. Note that for the sake of more reliable timings, the computations were repeated 30 times and the mean/median timings obtained are recorded for completeness. As expected (since only 5 terms are employed

N_{max}	Mean/median timings (s)	% calls of divergent series
10	0.2153/0.2509	100
400	0.2350/0.2705	52
1000	0.2495/0.2945	32.1
4000	0.3804/0.4503	0

Table 1: Mean/median timings (s) over 30 runs and percentages of calls of the divergent series of algorithm 4.

in the divergent series), Table 1 shows a speed up of the computations when switching more often for the divergent series. The trade-off is that only the convergent series method can provide arbitrarily precise results. However, it is interesting to note that in this particular test case, the results given by the divergent series method are also very accurate as shown in Figure 3.

Secondly, all the collision probability computation methods - algorithm 4, EVC, EV and AD algorithms - are compared both in terms of accuracy and efficiency. The results are depicted in Fig. 3 with a time step of 1 minute. It

is clear that the hybrid algorithm gives the best results as the red dashed line almost fits perfectly the Monte-Carlo simulations represented by the green stars. On the contrary, the approximations given by both of Chan's method are less precise. The EV method (dotted orange line) strongly underestimates the instantaneous collision probability on the whole time span while the AD method (dashed dot purple line) is more precise, still giving overestimated estimations of $\mathcal{P}_{inst}$ on the whole time span. In this last case, two particular features of the results are interesting: first, the AD method exhibits a non negligible risk on time intervals (at the beginning and the end of the encounter time span) where there is no actual risk; secondly, the instantaneous collision probability produced by the AD algorithm is not a lower bound of the cumulative probability given in [4], as it should be, on more than half of the time span. Finally, the EVC algorithm (solid blue line) gives a better result. Nevertheless, it is affected by inaccuracies on the entire range, due to the assumed approximations. Note that the assumption about the preservation of the Gaussian nature of the uncertainty throughout the encounter is *a posteriori* validated by the good agreement between the results of the hybrid algorithm and of the Monte Carlo simulations. Except for the MC method, each method requires less than 1 s for computing the complete set of data necessary to get the graph 3 as it may be seen in Table 2. This test case is a first illustration of the capability of the hybrid algorithm to provide a fast and accurate way to compute the instantaneous collision probability.

Alg. & propag.	Alg. 4 + UT	EVC + UT	EV + UT	AD + UT	MC
Timings (s.)	0.2153	0.2045	0.70312	0.54688	1447

Table 2: Comparison of the different timings for the different algorithms.

5.4 Two satellites in Highly-Eccentric Orbits (HEO)

This is test case 9 of reference [4], in which the instantaneous collision probability, computed by the voxel method, is presented alongside the cumulative probability. It describes two satellites in HEO with low relative velocity (around 0.002 m/s at TCA). The same technical choices as the ones made for test case 4 have been adopted here and Table 3 shows the timings and percentages of instances the algorithm 4 chooses to use the divergent series for different choices of N_{max}. Even if the evolution of timings is not as monotonic as in the previous example, there is an expected timing improvement when the divergent series method is called more often.

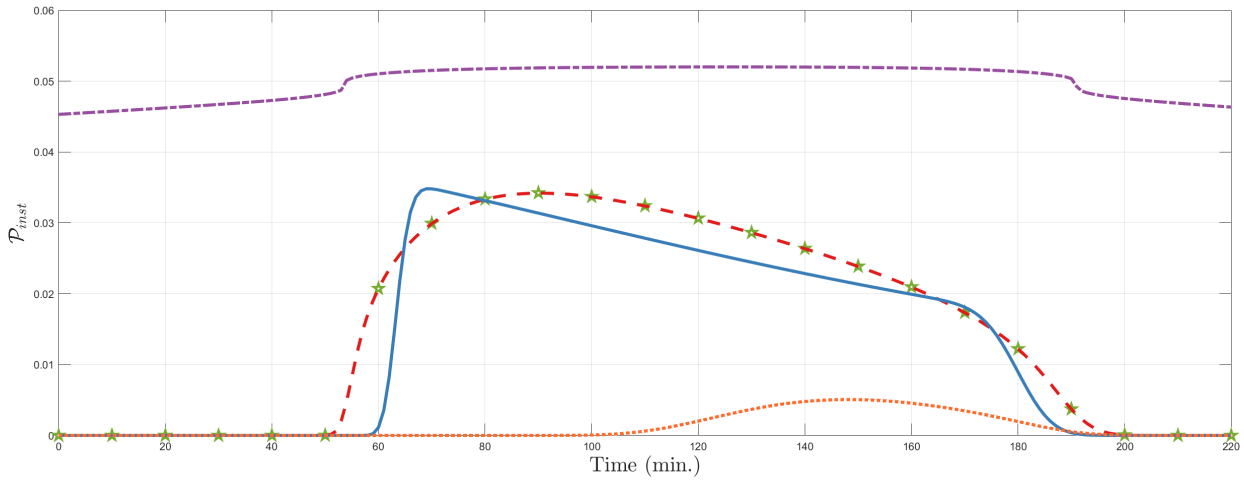


Figure 3: Alfano's [4] test case 4: $\mathcal{P}_{inst}$ - Monte Carlo simulations (green stars), Algorithm 4 with divergent series (red dashed line), AD method (dashed dot purple line), EV method (dotted orange line), EVC method (solid blue line).

N_{max}	Mean/median timings (s)	% calls of divergent series
50	0.3107/0.3396	100
75	0.2828/0.3411	73.4
100	0.3355/0.3438	57.1
300	0.3239/0.3584	23.6
700	0.3687/0.3877	0

Table 3: Mean/median timings (s) over 30 runs and percentages of calls of the divergent series of algorithm 4.

For the comparisons with all other methods, note that the number of samples for Monte-Carlo simulation was set to 2×10^6 in compliance with the Dagum and the Chernoff bounds given in [4] and that the time step was set to 10 minutes for computational reasons. The overall results can be seen in Fig. 4 and comparisons of timings are given in Table 4. First, note that the results of algorithm 4 and of the Monte-Carlo simulations are very similar to the ones exposed in [4]. In particular, the respective maximum probabilities of collision are similar up to the third digit (less than 0.02% of relative error). In terms of accuracy, the results of the other algorithms are quite consistent with the ones obtained for the first example except for the EV method, which suffers from severe numerical difficulties. For the whole computation, Algorithm 4 is the fastest while the EVC method is 4 times slower. The two other methods, AD and EV are faster than the EVC method but, AV strongly underestimates $\mathcal{P}_{inst}$ on most of the time span, while EV produces no usable results due to numerical instabilities. As expected, the Monte Carlo method requires a heavy load of computations and is therefore not competitive with respect to the timing criterion. As previously, only one Gaussian density may be used to describe the initial relative uncertainty and propagated during the whole time span.

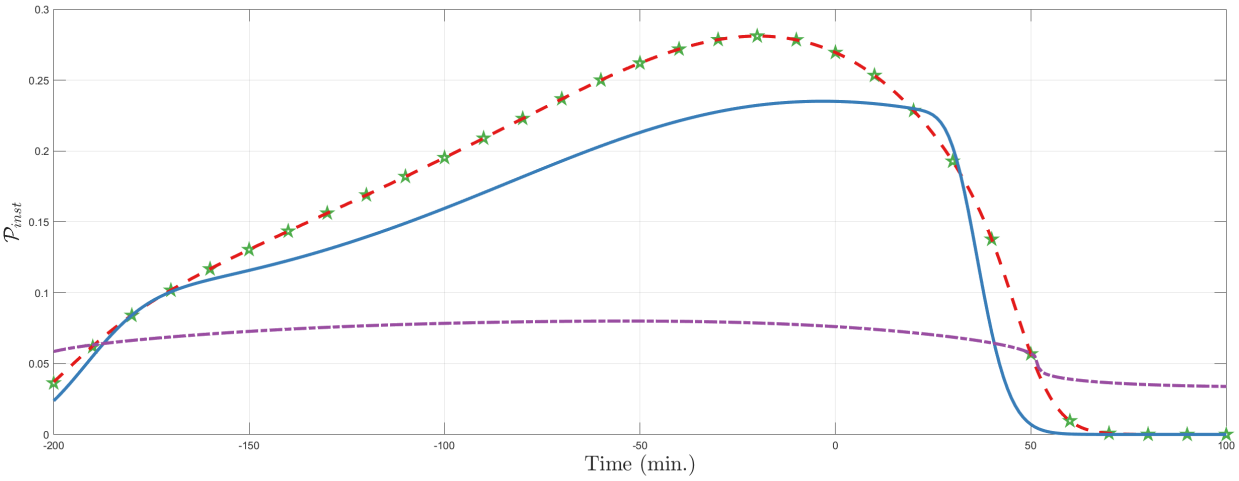


Figure 4: Alfano's [4] test case 9: $\mathcal{P}_{inst}$ - Monte Carlo simulations (green stars), algorithm 4 (red dashed line), AD method (dashed dot purple line), EVC method (solid blue line).

Alg. & propag.	Alg. 4 + UT	EVC + UT	EV + UT	AD + UT	MC
Timings (s.)	0.3107	1.2031	0.90625	0.54688	2172

Table 4: Comparison of the different timings for the different algorithms.

5.5 Two Satellites in Low-Earth Orbits

This test case was first presented and analyzed in [38]. It describes two satellites in LEO and it is built from two test cases from [4] : the values of the means are from test case 5 while the covariance matrices are from test case 7. For this case the propagation was performed from $t_0 = 0$ s (epoch) to $t_f = 102600\ s = 28\ h\ 30\ min$ that is, a long propagation period of more than 18 orbit periods for both objects. Since their respective PDFs are not likely to remain Gaussian, we implemented a Gaussian Mixture Model ('GMM') propagation method, beside the previously introduced single Gaussian model ('1G'). As mentioned earlier in Subsection 5.1, a GMM, through its initial splitting procedure, can be characterised by two parameters : the order k of the Gauss-Hermite quadrature and the number d of eigendirections along which to perform a split. Here, we choose a GMM with parameters $k = 5$ and $d = 3$, which has good propagation performances [4] while having a limited number of components (25 components with Genz-Keister's splitting algorithm), and compare it with the 1G model. Both propagation methods were combined either with the Equivalent Volume Cuboids or the Algorithm 4 ($N_{max} = 4000$) for computing the collision probability over the whole time span. The results are plotted in Fig. 5. The time window for the plot was centered around a local maximum of probability to better visualize the performance of each method.

As expected, the propagation method is of paramount importance here. For both probability computation methods, using GMM propagation model yields closer results to Monte-Carlo simulations than the ones obtained using a single Gaussian propagated by UT method. The relative error with respect to the Monte-Carlo result for the computation of maximum probability of collision, $\mathcal{P}_{inst}^{max}$, is given in Table 5 and clearly illustrates this point. Indeed, as depicted in Fig. 6, the difference between the 3-sigma PDF contours respectively propagated by the UT method, the GMM and the distribution of the relative position uncertainty projected in the radial-intrack plane at $t = t_f$ propagated by the Monte-Carlo method readily explains the obtained results.

[4] On this test case, the third largest eigenvalues of the combined covariance matrix was significantly higher than the fourth, hence the choice of $d = 3$.

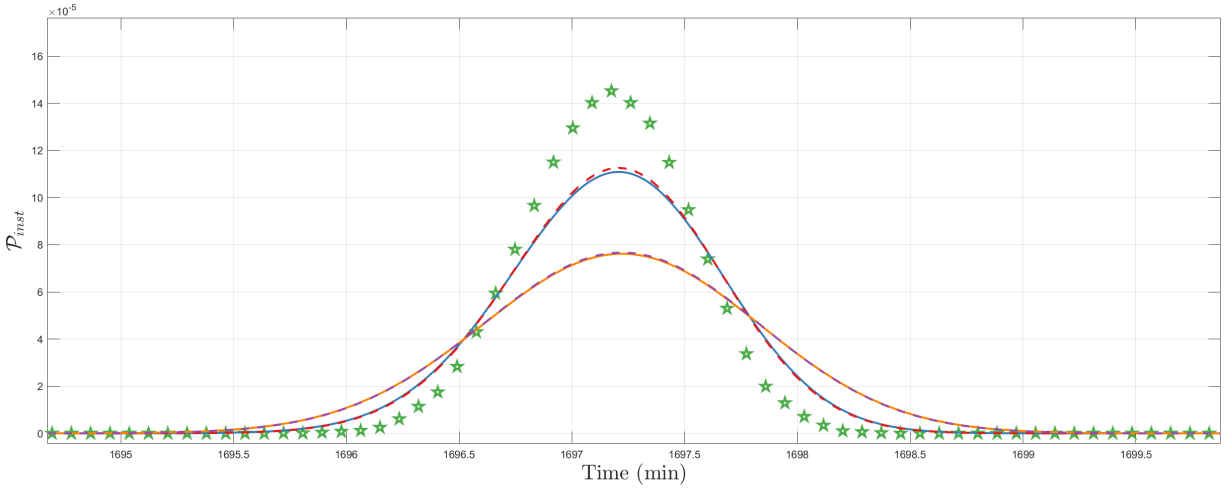


Figure 5: 2 satellites in LEO orbits: $\mathcal{P}_{inst}$ - Monte Carlo (green stars), GMM+algorithm 4 (red dash), GMM+EVC (blue line), 1G+ algorithm 4 (purple dash), 1G+EVC (orange line).

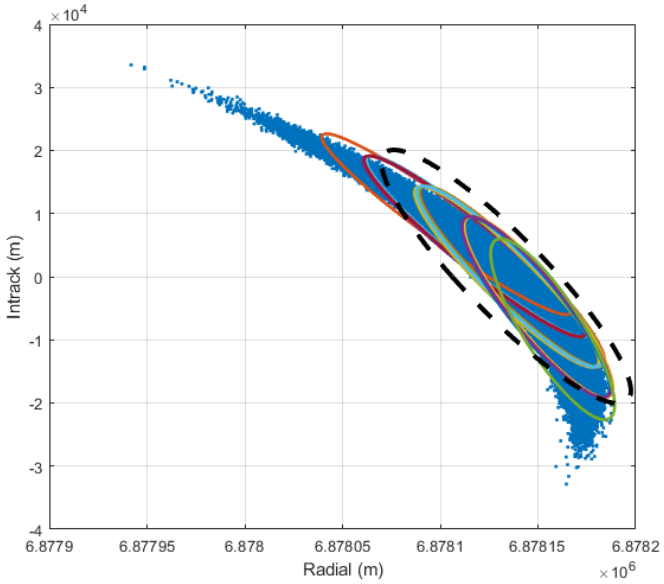


Figure 6: Distribution of the relative position uncertainty projected in the radial-intrack plane at $t = t_f$ propagated by the Monte-Carlo method (10^6 samples, blue dots) and 3-sigma PDF contours propagated by UT (black dashed line) and by GMM (colored lines).

Method	Timings	$\mathcal{P}_{inst}^{max} \times 10^4$ (error)
1G+EVC	4.3	0.76163 (47.5%)
1G+algo. 4	4.5	0.76654 (47.2%)
GMM+EVC	177	1.1087 (23.6%)
GMM+algo. 4	406	1.1255 (22.5%)
Monte-Carlo	25200	1.4520 (0%)

Table 5: Timings and error for two satellites in LEO orbits.

The illustrative timings of Table 5 show that Algorithm 4 with $N_{max} = 4000$ is here slower than the EVC method but with a slightly better accuracy. Nevertheless, the results of both Table 5 and Figure 5 show that it can be efficiently combined with more complex and demanding propagation techniques, while maintaining good accuracy and performance metrics.

6 Conclusion

In this work, the problem of the fast and reliable computation of the instantaneous probability of collision has been addressed. This can be modeled as the integral of a three-dimensional Gaussian probability density function over a Euclidean ball representing the hardbody. The method proposed for evaluating this integral is based on the combination of two semi-analytical complementary strategies. Firstly, a preconditioned convergent power series expansion in the square of the hardbody radius has been employed. From a computational viewpoint, this expression has two important features: (1) all its coefficients are positive, which prevents the well-known cancellation phenomenon which occurs in finite-precision evaluation; (2) the computation of these coefficients is very simple since they satisfy a linear recurrence relation, with known initial conditions. In addition, the proposed method allows for an evaluation with guaranteed accuracy, function of an absolute error threshold provided by the user. This is possible thanks to the derivation of rigorous lower and upper bounds on the truncation error, which can be also used for an *a priori* estimation of a sufficient number of terms required in the series.

Still, for some numerically challenging cases, this number can be too large (with respect to the efficiency constraints imposed). We have therefore proposed an alternative approximation method based on saddle point techniques, which can be seen as a generalization of the classical Laplace approximation method for integral evaluation. This method requires few terms (only five terms have been used in our algorithm) obtained via straightforward alge-

braic operations to get a very good approximations. However, these terms correspond to a divergent series, so this strategy cannot provide arbitrarily good accuracy.

Hence, a hybrid algorithm was designed, which selects one of these two complementary methods, depending on the accuracy required and a fixed threshold value for the maximum number of terms to be used in the series.

The results from three test cases directly borrowed or adapted from the literature have then been presented and comparisons with the current state of the art, including a Monte Carlo method taken as a reference, have been made. The first two test cases show a very good accuracy of the proposed hybrid algorithm with respect to the reference results when the assumption that the uncertainty remains Gaussian all along the time span is validated. These numerical tests show that our method is more accurate than the current state of the art, while also alleviating the difficult cases which required a high number of terms. For the third case, the propagation time span is longer, so a refined Gaussian mixture representation of the uncertainty is necessary. The experimental results show that the proposed hybrid algorithm scales up in this computationally challenging context. Moreover, on this particular example, the hybrid algorithm turns out to be slightly slower than the EVC method but also slightly more accurate.

Future studies include providing truncation error bounds for the saddle-point approach and the use of these computational algorithms for designing collision avoidance maneuvers.

Acknowledgement

The Authors would like to thank CNES and ANR-20-CE-48-0014NuSCAP for a partial financial support of this study.

A Proofs for the convergent series recurrence

Proofs of results pertaining to the convergent series approach are now presented.

Proposition 2 *The coefficients* $(\alpha_k)_{k\geq 0}$ *and* $(c_k)_{k\geq 0}$ *in equation* (5) *are positive.*

Proof 1 *Let* $\bar{h}(z) := \dfrac{1}{z^{\frac{5}{2}}}\mathcal{L}_{\tilde{g}}(\frac{1}{z})$. *The function* $\bar{h}$ *has a convergent power series expansion* $\bar{h}(z) = \displaystyle\sum_{k=0}^{\infty} \alpha_k z^k$ *for* $|z| \leq p^{-1}$ *and satisfies the following linear*

Ordinary Differential Equation (ODE) of order 1:

$$\bar{h}'(z) = \frac{P(z)}{Q(z)}\bar{h}(z), \quad \bar{h}(0) = C, \tag{36}$$

with the partial fraction decomposition of $\frac{P(z)}{Q(z)}$ *given by:*

$$\frac{P(z)}{Q(z)} = \frac{p}{1-pz} + \frac{\theta_2}{2(\gamma_2 z-1)^2} + \frac{\theta_3}{2(\gamma_3 z-1)^2} + \frac{\gamma_2}{2(1-\gamma_2 z)} + \frac{\gamma_3}{2(1-\gamma_3 z)} + \frac{\theta_1}{2}. \tag{37}$$

The general term of the series expansion of the function $\frac{P(z)}{Q(z)} = \sum_{i=0}^{\infty} \beta_i z^i$ *is therefore given by:*

$$\beta_i = p^{i+1} + \frac{(i+1)\theta_2\gamma_2^i}{2} + \frac{(i+1)\theta_3\gamma_3^i}{2} + \frac{\gamma_2^{i+1}}{2} + \frac{\gamma_3^{i+1}}{2}, \ + \frac{\theta_1}{2} \ \textit{if}\{i=0\}, \tag{38}$$

Injecting in the differential equation (36)*, we have:*

$$(k+1)\alpha_{k+1} = \sum_{i=0}^{k} \beta_i \alpha_{k-i},\ k \geq 0,\ \alpha_0 = C > 0. \tag{39}$$

By induction, we get that $\alpha_k > 0$ *for all* k *since* $\beta_k > 0$ *by* (38)*.*

Proof of Proposition 1. Denoting $x = R^2$, let the series T_n be defined by $T_n = R\exp(-pR^2)\sum_{k=0}^{\infty} \frac{\alpha_{k+n}x^{k+n}}{\Gamma(k+n+5/2)}$, whose coefficients are positive and $x \geq 0$. Furthermore, it is easily seen that the coefficients β_i in eq. (38) are lower-bounded by $p^{i+1} \leq \beta_i$, for $i \geq 0$, which gives $\alpha_0 p^i \leq \alpha_i$, for $i \geq 0$ by induction. Then, the inequality

$$\sum_{k=0}^{\infty} \frac{\alpha_{k+n}x^{k+n}}{\Gamma(k+n+5/2)} \geq \sum_{k=0}^{\infty} \frac{\alpha_0 p^{k+n}x^{k+n}}{\Gamma(k+n+5/2)} \geq \frac{\alpha_0(px)^n}{\Gamma(n+5/2)}, \tag{40}$$

provides a lower bound l_n for T_n.

Secondly, for an upper bound on T_n, one can use the following inequalities:

$$\begin{aligned} \sum_{k=0}^{\infty} \frac{\alpha_{k+n}x^{k+n}}{\Gamma(k+n+5/2)} &\leq \frac{(\frac{x}{\rho})^n}{\Gamma(n+5/2)} \sum_{k=0}^{\infty} \alpha_{k+n}x^k\rho^n \frac{\Gamma(n+5/2)}{\Gamma(k+n+5/2)} \\ &\leq \frac{(\frac{x}{\rho})^n}{\Gamma(n+5/2)} \sum_{k=0}^{\infty} \alpha_{k+n}\rho^{k+n} \frac{x^k}{\rho^k(n+5/2)^k}, \end{aligned} \tag{41}$$

which provide an upper-bound of the form:

$$u_{n,\rho} = \frac{R^{2n+3}}{\rho^n \Gamma(n+5/2)} \bar{h}(\rho) \exp(-pR^2), \tag{42}$$

for any $\rho < 1/p$ and $n \geq n_0$, with $n_0 = \lceil \frac{R^2}{\rho} \rceil$. In particular, setting $\rho = \frac{1}{2p}$ leads to:

$$u_{n,\frac{1}{2p}} := \frac{\alpha_0 p^n R^{2n+3}}{2^n \Gamma(n+5/2)} \frac{4p}{\sqrt{\overline{\gamma}_3 \overline{\gamma}_2}} \exp\left(\frac{\theta_1}{4p} + \frac{\theta_2}{2\overline{\gamma}_2} + \frac{\theta_3}{2\overline{\gamma}_3} - pR^2 \right), \tag{43}$$

for all $n \geq \lceil 2pR^2 \rceil$.

Proof of equation (11)**.** One has

$$u_n \leq \frac{2p^n R^{2n+3}}{\sqrt{2}\sigma_1\sigma_2\sigma_3 2^n \Gamma(n+5/2)} \exp\left(\frac{\theta_1}{4p} + \frac{\theta_2}{2\overline{\gamma}_2} + \frac{\theta_3}{2\overline{\gamma}_3} - pR^2 - \left(\frac{m_1^2}{2\sigma_1^2} + \frac{m_2^2}{2\sigma_2^2} + \frac{m_3^2}{2\sigma_3^2} \right) \right) \tag{44}$$

Using Stirling's inequality $\Gamma(n+1/2) \in \left[\frac{2\pi}{e}, e\right] \frac{n^n}{e^n}$, one has

$$u_n \leq \frac{eR^3}{\pi\sqrt{2}\sigma_1\sigma_2\sigma_3} \frac{(pR^2)^n}{2^n} \frac{e^{n+2}}{(n+2)^{n+2}} \exp(\Omega), \tag{45}$$

$$\leq \frac{4}{p^2 R \pi \sqrt{2}\sigma_1\sigma_2\sigma_3} \left(\frac{epR^2}{2(n+2)} \right)^{n+2} \exp(\Omega + 1), \tag{46}$$

where $\Omega = \frac{\theta_2}{2\overline{\gamma}_2} + \frac{\theta_3}{2\overline{\gamma}_3} - pR^2 - \left(\frac{m_1^2}{4\sigma_1^2} + \frac{m_2^2}{2\sigma_2^2} + \frac{m_3^2}{2\sigma_3^2} \right)$.

The term $\frac{epR^2}{2(n+2)}$ is less than $\frac{1}{2}$ as soon as $n \geq N_1 := \lceil epR^2 - 2 \rceil$. Therefore, the inequality $u_n \leq \delta$ is satisfied for $n \geq N_2 := \lceil (\Omega+1)\log_2 e - \log_2(\delta p^2 R \pi \sqrt{2}\sigma_1\sigma_2\sigma_3) \rceil$.

B Tables of precomputed coefficients a_n, b_n and c_n involved in the computation of $\mathcal{P}_{inst}$ via the saddle point method

This appendix gathers the different tables of the various coefficients involved in the steps of Algorithm 3.

$$
\begin{aligned}
b_1 &= 1, \\
b_3 &= \frac{-4a_2a_4 + 5a_3^2}{8a_2^2}, \\
b_5 &= \frac{-64a_2^3a_6 + 224a_2^2a_3a_5 + 112a_2^2a_4^2 - 504a_2a_3^2a_4 + 231a_3^4}{128a_2^4}, \\
b_7 &= \frac{1}{1024a_2^6}\left[-512a_2^5a_8 + (2304a_3a_7 + 2304a_4a_6 + 1152a_5^2)a_2^4 + \right. \\
&\quad (-6336a_3^2a_6 - 12672a_3a_4a_5 - 2112a_4^3)a_2^3 + (13728a_3^3a_5 + 20592a_3^2a_4^2)a_2^2 \\
&\quad \left. -25740a_2a_3^4a_4 + 7293a_3^6\right], \\
b_9 &= \frac{1}{32768a_2^8}\left[-16384a_2^7a_{10} + (90112a_3a_9 + 90112a_4a_8 + 90112a_5a_7 + \right. \\
&\quad 45056a_6^2)a_2^6 + (-292864a_3^2a_8 + (-585728a_4a_7 - 585728a_5a_6)a_3 - \\
&\quad 292864a_4(a_4a_6 + a_5^2))a_2^5 + (732160a_3^3a_7 + (2196480a_4a_6 + \\
&\quad 1098240a_5^2)a_3^2 + 2196480a_3a_4^2a_5 + 183040a_4^4)a_2^4 - 1555840a_3^2(a_3^2a_6 + \\
&\quad 4a_3a_4a_5 + 2a_4^3)a_2^3 + (2956096a_3^5a_5 + 7390240a_3^4a_4^2)a_2^2 - \\
&\quad \left. 5173168a_2a_3^6a_4 + 1062347a_3^8\right].
\end{aligned}
$$

Table 7: b_n coefficients of the change of variables (24), as functions of the coefficients a_n.

$$
\begin{aligned}
a_0 &= \xi z_s - z_s \sum_{i=1}^{3} \frac{m_i^2 p_i}{z_s + p_i} - \log z_s - \frac{1}{2}\sum_{i=1}^{3} \log\left(\frac{z_s}{p_i} + 1\right), \\
a_1 &= \xi - \sum_{i=1}^{3} \frac{m_i^2 p_i^2}{(z_s + p_i)^2} - \frac{1}{z_s} - \frac{1}{2}\sum_{i=1}^{3} \frac{1}{z_s + p_i} = 0, \\
a_n &= (-1)^n \left(\sum_{i=1}^{3} \frac{m_i^2 p_i^2}{(z_s + p_i)^{n+1}} + \frac{1}{n z_s^n} + \frac{1}{2n}\sum_{i=1}^{3} \frac{1}{(z_s + p_i)^n}\right), \quad n \geq 2.
\end{aligned}
$$

Table 6: a_n coefficients as functions of $\xi = R^2$, z_s, m_i and $p_i = 1/2\sigma_i^2$.

$c_0 = b_1,$	$c_1 = -\frac{3}{2}\frac{b_3}{a_2},$	$c_2 = \frac{15}{4}\frac{b_5}{a_2^2},$	$c_3 = -\frac{105}{8}\frac{b_7}{a_2^3},$	$c_4 = \frac{945}{16}\frac{b_9}{a_2^4}.$

Table 8: The first five c_n coefficients of the divergent series (31).

References

[1] N. Adurthi and P. Singla. Conjugate unscented transformation-based approach for accurate conjunction analysis. *Journal of Guidance Control and Dynamics*, 38(9):1642–1658, 2015. https://doi.org/10.2514/1.G001027.

[2] M. Akella and K. Alfriend. Probability of collision between space objects. *Journal of Guidance, Control and Dynamics*, 23(5):769–772, 2000. https://doi.org/10.2514/2.4611.

[3] S. Alfano. A numerical implementation of spherical objet collision probability. *Journal of Astronautical Sciences*, 53(1), January-March 2005. https://doi.org/10.1007/BF03546397.

[4] S. Alfano. Satellite conjunction Monte Carlo analysis. *Advances in the Astronautical Sciences*, 134:2007–2024, jan 2009. https://doi.org/10.1007/BF03546397.

[5] D. Arzelier, F. Bréhard, and M. Joldes. Comment on "series for collision probability in short-encounter model". *Journal of Guidance Control and Dynamics*, 43(5):1034–1037, 2020. https://doi.org/10.2514/1.G004560.

[6] C. Bender and S. Orszag. *Advanced mathematical methods for scientists and engineers I: Asymptotic methods and perturbation theory*, volume 1. Springer Science & Business Media, 1999. https://doi.org/10.1007/978-1-4757-3069-2.

[7] F. Chan. *Spacecraft Collision Probability*. American Institute of Aeronautics and Astronautics, Reston, Virginia, USA, 2008. http://dx.doi.org/10.2514/4.989186.

[8] V. Coppola. Evaluating the short encounter assumption of the probability of collision. In *22nd AAS/AIAA Spaceflight Mechanics Meeting*, Charleston, SC, USA, 2012.

[9] V. Coppola. Including Velocity Uncertainty in the Probability of Collision between Space Objects. *Advances in the Astronautical Sciences*, 143:2159–2178, 2012.

[10] E. Copson. *An Introduction to the theory of functions of a complex variable*. Oxford University Press, London, UK, 1935. https://doi.org/10.1063/1.3061616.

[11] R. Davies. Numerical inversion of a characteristic function. *Biometrika*, 60(2):415–417, August 1973. https://doi.org/10.1093/biomet/60.2.415.

[12] R. Davies. Algorithm as 155: The distribution of a linear combination of ξ^2 random variables. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 29(3):323–333, 1980. https://doi.org/10.2307/2346911.

[13] W. Feller. *An Introduction to Probability Theory and Its Applications, Volume 1,*. John Wiley, 1957. https://doi.org/10.1017/S0020268100037586.

[14] R. García-Pelayo and J. Hernando-Ayuso. Series for collision probability in short-encounter model. *Journal of Guidance Control and Dynamics*, 39(8):1908–1916, 2016. https://doi.org/10.2514/1.G001754.

[15] W. Gawronski, J. Müller, and M. Reinhard. Reduced cancellation in the evaluation of entire functions and applications to the error function. *SIAM J. Numerical Analysis*, 45(6):2564–2576, 2007. https://www.jstor.org/stable/40233192.

[16] A. Genz and B. Keister. Fully symmetric interpolatory rules for multiple integrals over infinite regions with gaussian weight. *Journal of Computational and Applied Mathematics*, 71(2):299–309, 1996. https://doi.org/10.1016/0377-0427(95)00232-4.

[17] D. Hall, M. Hejduk, and L. Johnson. Time dependence of collision probabilities during satellite conjunctions. In *Proceedings of AAS/AIAA Space Flight Mechanics Meeting*, number AAS 17-271, Austin, TX, USA, February 2017.

[18] J. T. Horwood and A. B. Poore. Adaptive gaussian sum filters for space surveillance. *IEEE Transactions on Automatic Control*, 56(8):1777–1790, 2011. https://doi.org/10.1109/TAC.2011.2142610.

[19] N. Johnson and S. Kotz. *Continuous univariate distributions, volume 2*. Wiley Series in Probability and Mathematical Statistics. New York: John Wiley & Sons, 1970. https://doi.org/10.1080/00224065.1996.11979675.

[20] B. Jones and A. Doostan. Satellite collision probability estimation using polynomial chaos expansions. *Advances in Space Research*, 52(11):1860–1875, 2013. http://dx.doi.org/10.1016/j.asr.2013.08.027.

[21] S. Julier, J. Uhlmann, and H. Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 45(3):477–482, 2000. https://doi.org/10.1109/9.847726.

[22] D. Knuth. *Art of Computer Programming, Volume 2: Seminumerical Algorithms*. Addison Wesley, 1998.

[23] S. Kotz, N. Johnson, and D. Boyd. Series representation of the distribution of quadratic forms in normal variables i. central case. *Ann. Math. Statist.*, 38:823–837, 1967. https://doi.org/10.1214/aoms/1177698877.

[24] S. Kotz, N. Johnson, and D. Boyd. Series representation of the distribution of quadratic forms in normal variables ii. non-central case. *Ann. Math. Statist.*, 38:838–848, 1967. https://doi.org/10.1214/aoms/1177698878.

[25] G. Krier. Satellite Collision Probability for Long-term Encounters and Arbitrary Primary Satellite Shape. Paper presented at the 7th European Conference on Space Debris, April 2017.

[26] Y. Kwok. *Applied Complex Variables for Scientists and Engineers*. Cambridge University Press, Cambridge, UK, 2002. https://doi.org/10.1017/CBO9780511844690.

[27] J. B. Lasserre and E. S. Zeron. Solving a class of multivariate integration problems via laplace techniques. *Applicationes Mathematicae*, 28(4):391–405, 2001. https://doi.org/10.4064/am28-4-2.

[28] A. Mathai and S. Provost. *Quadratic forms in random variable: Theory and Applications*. Statistics: Texbooks and Monographs. Marcel Dekker, New York, USA, 1992. https://doi.org/10.2307/2290674.

[29] U. Núñez Garzón and E. Lightsey. Relating collision probability and separation indicators in spacecraft formation collision risk analysis. *Journal of Guidance, Control, and Dynamics*, 45(3):517–532, March 2022. https://doi.org/10.2514/1.G005744.

[30] R. Patera. General Method for Calculating Satellite Collision Probability. *Journal of Guidance, Control, and Dynamics*, 24(4):716–722, July 2001. http://dx.doi.org/10.2514/2.4771.

[31] R. Patera. Satellite collision probability for nonlinear relative motion. *Journal of Guidance Control and Dynamics*, 26(5):728–733, 2003. https://doi.org/10.2514/2.5127.

[32] B. Salvy. D-finiteness: Algorithms and applications. In M. Kauers, editor, *ISSAC 2005: Proceedings of the 18th International Symposium on Symbolic and Algebraic Computation, Beijing, China, July 24-27, 2005*, pages 2–3. ACM Press, 2005. http://dx.doi.org/10.1145/1073884.1073886.

[33] R. Serra, D. Arzelier, M. Joldes, J. Lasserre, A. Rondepierre, and B. Salvy. Fast and accurate computation of orbital collision probability for short-term encounters. *Journal of Guidance Control and Dynamics*, 39(9):1009–1021, 2016. https://doi.org/10.2514/1.G001353.

[34] R. Serra, D. Arzelier, M. Joldes, J.-B. Lasserre, A. Rondepierre, and B. Salvy. A Power Series Expansion based Method to compute the Probability of Collision for Short-term Space Encounters. Research report, LAAS-CNRS 15072, March 2015.

[35] R. Serra, D. Arzelier, M. Joldes, and A. Rondepierre. Probabilistic collision avoidance for long-term space encounters via risk selection. In *Advances in Aerospace Guidance, Navigation and Control*, pages 679–698. Springer, December 2015. https://doi.org/10.1007/978-3-319-17518-8_39.

[36] S. W. Shepperd. Universal keplerian state transition matrix. *Celestial mechanics*, 35:129–144, 1985. https://doi.org/10.1007/BF01227666.

[37] R. Wong. *Asymptotic approximation of integrals*. Computer Science and Scientific Computing. Academic Press, London, UK, 1989. https://doi.org/10.1137/1.9780898719260.

[38] T. Yan, J. Gong, S. Liu, R. Wang, and L. Shi. Gaussian sum reapproximation applied to the probability of collision calculations. *Advances*

in Space Research, 68:3846–3858, 2021. https://doi.org/10.1016/j.asr.2021.07.023.

[39] S. Zhang, T. Fu, D. Chen, and H. Cao. Satellite instantaneous collisionprobability computation using equivalent volume cuboids. *Journal of Guidance, Control and Dynamics*, 43(9):1757–1763, 2020. https://doi.org/10.2514/1.G004711.