Rounding Error Analysis of an Orbital Collision Probability Evaluation Algorithm

Denis Arzelier, Florent Bréhard, Mioara Joldes, Marc Mezzarobba

Rounding Error Analysis of an Orbital Collision Probability Evaluation Algorithm

Denis Arzelier[1], Florent Bréhard[2], Mioara Joldes[3], and Marc Mezzarobba[4]

[1]LAAS-CNRS, Toulouse, France, arzelier@laas.fr
[2]CRIStAL, CNRS, Lille, France, florent.brehard@univ-lille.fr
[3]LAAS-CNRS, Toulouse, France, joldes@laas.fr
[4]LIX, CNRS, Palaiseau, France, marc@mezzarobba.net

Abstract

We present an error analysis of an algorithm due to Serra *et al.* (*Journal of Guidance Control and Dynamics*, 2016) for computing the orbital collision probability in the short term encounter model. The algorithm reduces the numerical computation of the collision probability to that of the sum of a series whose coefficients are produced by a linear recurrence relation, and is specifically designed to avoid cancellation issues in the evaluation of the sum. The authors derived a bound on the method error arising from the truncation of the series, and observed experimentally that the computation of the *terms* of the sum is numerically stable, but did not study the evaluation error. Here we give a rigorous bound on the accumulated rounding error when Serra *et al.*'s algorithm is implemented in floating-point arithmetic. For a unit roundoff u and a truncation order N, the bound is of the form $(N + A)u + o(u)$ where A is an explicit constant depending on the problem parameters and $o(u)$ stands for explicitly bounded small terms compared to u. Our analysis is based on the observation that the generating series of the errors affecting each individual term is solution to a perturbed form of a differential equation satisfied by the Laplace transform of a function related to the collision probability.

1 Introduction

Due to the drastic increase in the space debris number during the last decades, collision avoidance has become a usual and necessary procedure for many active satellites. The uncertainty affecting the measured data characterizing an encounter is a strong incentive to rely on a probability of collision as the decision variable to trigger a possible avoidance maneuver. When modeling conjunctions, two main paradigms — the short-term and the long-term encounters — are widely accepted and implemented in the field of orbital collision risk assessment [3]. The first one is most frequently used in practice and assumes that the relative velocity between the two objects is sufficiently high, so that the encounter time is short. In this framework, the orbital collision probability is modeled as a 2-D integral on a disk, which can be efficiently evaluated using an approximation by a power series. The corresponding algorithm of Serra *et al.* [12] was implemented in Floating Point (FP) arithmetic and has been used in practice by the French Space Agency (CNES) for ground space surveillance operations. More recently, an on-board implementation was successfully tested on an experimental satellite [14].

While the parameters of this algorithm are only estimations of physical quantities, it is however important to provide guaranties about the accuracy and reliability of its numerical implementation. This can be seen by analogy to the need for accurate implementations of special functions (like erf, Airy, Bessel, etc.) used in calculations of other physical phenomena. One would like to estimate and bound independently the numerical evaluation and truncation error for such a mathematical function, compared to other model errors.

With this in mind, the mathematics for this problem were well-studied (truncation error bounds, positivity of the coefficients), but the round-off error analysis was so-far ignored. This was probably due to the difficulty of the task, since it involves a loop, which implements the evaluation of a linearly recursive sequence.

It is known that the naive rounding error analysis of such recurrences can result in overestimation of the bounds [11] because rounding errors generated in the evaluation of the loop typically cancel out to a large extent, instead of purely adding up. Taking into account this phenomenon usually involves a careful study of the propagation of local errors in following steps of the algorithm, implying complicated manipulations of nested sums and yielding opaque expressions. To alleviate this issue, the main idea of the recent work [11] is to encode as generating series both the sequence of *local* errors committed at each step and that of *global* errors resulting from the accumulation of local errors. While far from classical in the context of rounding-error

analysis, this technique proves to be very adequate for studying algorithms which originate from numerical methods implementing truncated series approximations, with coefficients satisfying linear recurrences.

Different alternative methods are now briefly recalled. Firstly, unrolling a linear recurrence can be seen as a special case of solving a triangular (banded Toeplitz) system of linear equations. Therefore, a first result based on [8, Chap. 8], bounds the maximum relative error for evaluating n terms of an order m recurrence by the product of $\frac{mu}{1-mu}$ with the condition number of the associated n by n matrix. In this sense, in [2], a more refined analysis gives a first-order bound (meaning that the terms of order $O(u^2)$ are omitted). However, one has to resort to more complicated formulas, expressed in terms of quantities that may be difficult to estimate (inverting the associated triangular matrix, computing the so-called reverse homogeneous recurrence for instance).

A complementary class of approaches concerns the use of static error analyzers, which automatically provide sound (and often formally-proven) error bounds on FP rounding errors (see for instance [1, 13, 15] and references therein for existing software). While these tools are aimed at generic numerical codes, they are not very efficient for handling a very large number of loop iterations due to the intrinsic high depth of the expression graph. For instance, one of the currently fastest tools, SATIRE [4], reports a minimum execution time of $50s$ for unrolling 70 iterations of the Lorenz system. By comparison, the algorithm analyzed in this article sometimes requires hundreds (or even thousands) of iterations. Furthermore, the parameters involved have rather high ranges and we would like an error bound which depends explicitly on these parameters, without additional runs of the program.

Finally, let us also mention that a basic automatic evaluation in interval arithmetic highly overestimates the bounds as the iterations directly reuse the previously-computed values.

All in all, we believe that the adaptation of [11] to this particular algorithm, offers a good remedy to these limitations and that the mathematical tools employed herein may be of interest to the rounding-analysis spectrum of methods.

The structure and contributions of this article are as follows. Firstly, we recall in Section 2 the description of a Laplace transform technique from [12], which proves that the terms of the recursive sequence implemented as the main loop of the algorithm are the coefficients of a series solution to a simple first-order Linear Differential Equation (LDE). From that, the main contribution of this article is to make heavy use of this equation to interpret the individual rounding errors on each term as the coefficients of another series

whose analytic behavior is essentially similar, up to a factor proportional to the roundoff unit. This is the key point for deriving realistic total relative error bounds. To do that, a preliminary step in Section 3 is a classical rounding error analysis for the loop-independent parameters and the body of the loop (*local* errors).

Then, we bound the *global* errors accumulating when executing the main loop in Section 4. There, we make the key observation that the generating series in the Laplace plane associated to global errors is solution to the same previously mentioned LDE, but with an inhomogeneous term generated by the local errors. Working with series in the Laplace plane allows for sufficiently simple closed-form formulas, even if it sometimes means performing some crafty term-by-term majorizations. This technique allows for the computation of explicit *a priori* rounding error bounds depending on the input parameters, without any additional runs or restrictive imposed ranges on the parameters. Given a unit roundoff u and a truncation order N, the bound proposed in Theorem 1 is of the form $(N+A)u+o(u)$, where A is an explicit constant depending on the parameters and $o(u)$ stands for explicitly bounded small terms compared to u.

Finally, practical aspects are considered: in Section 5 we comment on how the analyzed implementation simulates an increased exponent range, as to avoid overflows in practice; then, a numerical validation of the quality of the bound is provided on a range of examples in Section 6.

2 Computing the Orbital Collision Probability

In this section, the main steps of the mathematical derivations used to build the reviewed algorithm are briefly reminded. This is particularly useful since the proposed rounding error analysis partly exploits the very same ingredients.

2.1 The short-term encounter model

The short-term encounter model (whose complete set of assumptions is recalled in detail in [3] or [12]) for the computation of the probability of collision between two spherical objects mainly consists in assuming that the relative trajectory is a straight line during the encounter and in projecting it onto the encounter plane defined to be perpendicular to the relative velocity vector. Let (x, y) denote the mean coordinates of the relative position of the secondary object with respect to the primary object in the encounter frame (see [12] for its definition). The relative position uncertainty is described by

the following bivariate Gaussian density function

$$\rho(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp\left[-\frac{1}{2}\left(\frac{(x-x_m)^2}{\sigma_x^2}+\frac{(y-y_m)^2}{\sigma_y^2}\right)\right],$$

where (x_m, y_m) is the mean position of the secondary object relative to the primary object in the covariance frame and $\sigma_x,\ \sigma_y \in \mathbb{R}_*^+$ are the standard deviations of the relative coordinates in the encounter plane. The probability of collision is then given by a two-dimensional integral parameterized by the radius R of the combined spherical object:

$$\mathcal{P}(R) = \int_{x^2+y^2\leqslant R} \rho(x,y)\mathrm{d}x\mathrm{d}y. \tag{1}$$

This integral is the cumulative density function of the random variable $\Xi = X^2 + Y^2$ (i.e., $\mathcal{P}(R) = \Pr\{\Xi \leqslant R^2\}$) where $X \rhd \mathcal{N}(x_m, \sigma_x^2)$, $Y \rhd \mathcal{N}(y_m, \sigma_y^2)$ are independent normal random variables. By rescaling Ξ, we get $\mathcal{P}(R) = g(1)$ where the function $g : \mathbb{R}^+ \mapsto \mathbb{R}^+$ is defined as $g(\xi) = \mathcal{P}(R\sqrt{\xi})$.

Several methods for computing the integral (1) have been proposed in the aerospace literature (see for instance [6, 12] and references therein). Here we focus on the algorithm of [12], where one obtains a convergent series expansion of $g(\xi)$ by considering its Laplace transform. Various versions of this idea have been applied to both central and non-central quadratic forms; see for instance [10, Chapter 4] and references therein.

2.2 Algorithm

Due to the cancellation phenomenon occurring when summing the terms of a series of different signs and similar magnitude, the direct evaluation of the power series expansion of $g(\xi)$ is only practical for small values of R^2. The idea of [12] to remedy this, inspired by [7], is to introduce a *preconditionner* of the form $\Pi(\xi) = \exp(p\xi R^2)$ and consider the expansion of the function $f = \Pi g$, which then has nonnegative coefficients. Assuming without loss of generality that $0 < \sigma_y \leqslant \sigma_x$, a good choice is $p = 1/(2\sigma_y^2)$.

We now summarize how the authors of [12] obtain a linear-time algorithm for computing the first N terms of the series f. We use the following notation:

$$p = \frac{1}{2\sigma_y^2}, \quad \phi = 1 - \frac{\sigma_y^2}{\sigma_x^2}, \quad \omega_x = \frac{x_m^2}{4\sigma_x^4}, \quad \omega_y = \frac{y_m^2}{4\sigma_y^4},$$

and observe that $0 \leqslant \phi < 1$, $\omega_x \geqslant 0$, $\omega_y \geqslant 0$.

The first step is to compute the Laplace transform of the preconditioned function in closed form. For $|\lambda| > p$, one has

$$\mathcal{L}_f(\lambda) = \mathcal{L}_g(\lambda - pR^2) = \frac{\pi R^2 \rho(0,0)\exp\left[\frac{\omega_y R^2}{\lambda} + \frac{\omega_x R^2}{\lambda - p\phi R^2}\right]}{\sqrt{\lambda(\lambda - p\phi R^2)}(\lambda - pR^2)}.$$

The power series expansion of $f(\xi)$ is the termwise inverse Laplace transform of the expansion at infinity of $\mathcal{L}_f(\lambda)$ (see [17, Chap. 9], [16, Chap. 2.14]), and the coefficients of the latter are the same as those of the Taylor expansion of

$$\mathcal{L}_f(\lambda^{-1}) = \frac{\pi R^2 \rho(0,0) \lambda^2 \exp\left[\omega_y R^2 \lambda - \frac{\omega_x}{p\phi} - \frac{\omega_x}{p\phi(p\phi R^2 \lambda - 1)}\right]}{\sqrt{1 - p\phi R^2 \lambda}(1 - pR^2\lambda)}.$$

Since the first two coefficients of $\mathcal{L}_f(\lambda^{-1})$ are zero, let $\hat{f}(\lambda) = \lambda^{-2}\mathcal{L}_f(\lambda^{-1})$. Letting $f(\xi) = \sum_{n=0}^{\infty} c_n \xi^{n+1}$ (note the $n+1$), the series expansion of $\hat{f}$ reads

$$\hat{f}(\lambda) = \sum_{n=0}^{\infty} c_n \, (n+1)! \lambda^n. \tag{2}$$

The second step is to derive a linear *recurrence relation* satisfied by the coefficients c_n. For this, one uses the fact that $\hat{f}(\lambda)$ is solution to a LDE. Indeed, starting from the definition of $\hat{f}$ and taking logarithmic derivatives, one has

$$\hat{f}'(\lambda) = \varphi(\lambda)\hat{f}(\lambda), \quad \hat{f}(0) = \pi R^2 \rho(0,0), \tag{3}$$

$$\begin{aligned} \varphi(\lambda) &= \omega_y R^2 + \frac{p\phi R^2}{2(1-p\phi R^2\lambda)} + \frac{pR^2}{1-pR^2\lambda} + \frac{\omega_x R^2}{(1-p\phi R^2\lambda)^2} \\ &= \frac{P(\lambda)}{Q(\lambda)} \quad \text{with } Q(\lambda) = (1-p\phi R^2\lambda)^2(1-pR^2\lambda). \end{aligned} \tag{4}$$

The coefficients of the polynomials P and Q alternate in sign: we write $P(\lambda) = P_0 - P_1\lambda + P_2\lambda^2 - P_3\lambda^3$ and $Q(\lambda) = 1 - Q_1\lambda + Q_2\lambda^2 - Q_3\lambda^3$ where $P_i, Q_i \geqslant 0$.

Lemma 1. *The sequence (c_n) satisfies the linear recurrence*

$$\begin{aligned} &nc_n - \frac{Q_1(n-1) + P_0}{n+1} c_{n-1} + \frac{Q_2(n-2) + P_1}{(n+1)n} c_{n-2} \\ &- \frac{Q_3(n-3) + P_2}{(n+1)n(n-1)} c_{n-3} + \frac{P_3}{(n+1)n(n-1)(n-2)} c_{n-4} = 0, \end{aligned} \tag{5}$$

for $n \geqslant 4$, with initial terms $c_0, \ldots, c_3$ given in Algorithm 1.

Proof sketch. The LDE (3) is equivalent to

$$Q(\lambda)\lambda\hat{f}'(\lambda) - P(\lambda)\lambda\hat{f}(\lambda) = 0. \tag{6}$$

One can check that, for any series $\hat{f}$ satisfying (2), one has

$$\begin{aligned}\lambda^k \hat{f}(\lambda) &= \sum_{n=k}^{+\infty} \frac{c_{n-k}}{(n+1)\dots(n-k+2)}(n+1)!\lambda^n,\\ \lambda(\hat{f})'(\lambda) &= \sum_{n=0}^{+\infty}(nc_n)(n+1)!\lambda^n.\end{aligned} \tag{7}$$

Using these identities repeatedly, one obtains

$$Q(\lambda)\lambda\hat{f}'(\lambda) - P(\lambda)\lambda\hat{f}(\lambda) = \sum_{n=0}^{\infty} F_n(c)(n+1)!\lambda^n,$$

where $F_n(c)$ is exactly the left-hand side of (5), with the additional convention that $c_{-1} = c_{-2} = c_{-3} = 0$ and terms whose denominator vanishes are ignored. It follows using (6) that $F_n(c) = 0$ for all $n \geqslant 0$. For $n \geqslant 4$, this gives the desired recurrence. For $n = 1, 2, 3$, one obtains the expressions for c_1, c_2, c_3 appearing in the algorithm, and similarly for c_0 since $c_0 = \hat{f}(0) = \pi R^2 \rho(0,0)$. □

Summarizing, we have

$$\mathcal{P}(R) = g(1) = \exp\left(-pR^2\right) \sum_{n=0}^{\infty} c_n,$$

where the coefficients c_n are given by Lemma 1. Algorithm 1 is a procedure for evaluating this expression. Our next goal is to analyse the effect of rounding errors on this procedure.

3 Local Rounding Error Bounds

In this section, we describe the employed FP setting and provide the local error analysis.

3.1 FP arithmetic setting

We assume that Algorithm 1 is implemented in radix-2, precision-t, round-to-nearest FP arithmetic, with unbounded exponent range. This means that, whenever an expression `a*b`, with a basic operation $* \in \{+, -, \cdot, /\}$, appears in the algorithm, what is effectively computed is $\mathrm{RN}(a * b)$, where $\mathrm{RN}(x)$ denotes the FP number closest to a real number x (with some arbitrary

Algorithm 1 Computation of the Probability of Collision.

Input: Parameters: σ_x, σ_y, x_m, y_m; combined object radius: R; number of terms: N.

Output: $\mathcal{P}_{0:N}$ – truncated series approximation of $\mathcal{P}$.

$p = \frac{1}{2\sigma_y^2}$; $\phi = 1 - \left(\frac{\sigma_y}{\sigma_x}\right)^2$; $\omega_x = \frac{x_m^2}{4\sigma_x^4}$; $\omega_y = \frac{y_m^2}{4\sigma_y^4}$;

$Q_1 = pR^2(2\phi + 1)$; $Q_2 = p^2R^4\phi(\phi + 2)$; $Q_3 = p^3R^6\phi^2$;

$P_0 = \left(p\left(\frac{\phi}{2} + 1\right) + \omega_x + \omega_y\right)R^2$;

$P_1 = \left(\frac{p\phi(\phi+5)}{2} + \omega_x + \omega_y(2\phi + 1)\right)pR^4$;

$P_2 = \left(\frac{3}{2}p + \omega_y(\phi + 2)\right)p^2R^6\phi$;

$P_3 = p^3\omega_yR^8\phi^2$;

$c_0 = \frac{1}{2\sigma_x\sigma_y}\exp\left(-\frac{1}{2}\left(\frac{x_m^2}{\sigma_x^2} + \frac{y_m^2}{\sigma_y^2}\right)\right)R^2$;

$c_1 = \frac{P_0}{2}c_0$; $\quad c_2 = \frac{Q_1+P_0}{6}c_1 - \frac{P_1}{12}c_0$;

$c_3 = \frac{2Q_1+P_0}{12}c_2 - \frac{Q_2+P_1}{36}c_1 + \frac{P_2}{72}c_0$;

$s = c_0 + c_1 + c_2 + c_3$

for $n = 4$ to $N - 1$ **do**

$c_n = \frac{Q_1(n-1)+P_0}{(n+1)n}c_{n-1} - \frac{Q_2(n-2)+P_1}{(n+1)n^2}c_{n-2}$
$+ \frac{Q_3(n-3)+P_2}{(n+1)n^2(n-1)}c_{n-3} - \frac{P_3}{(n+1)n^2(n-1)(n-2)}c_{n-4}$;

$s = s + c_n$;

end for

return $\mathcal{P}_{0:N} = \exp\left(-pR^2\right)s$.

tie-breaking rule). In particular, our arithmetic obeys the *standard error models* [8, Chap. 2.2.]

$$\mathrm{RN}(x) = x(1+r_1) = x/(1+r_2), \text{with } |r_1|, |r_2| \leqslant u, \tag{8}$$

with rounding unit $u = 2^{-t}$. In addition, multiplications by powers of two are exact.

This setting correctly models "real-life" IEEE-754 arithmetic provided that no overflows or underflows occur. It turns out that, when implemented in binary64 arithmetic, Algorithm 1 can easily encounter overflows for realistic values of the input. We comment in Section 5 on how the implementation by the authors of [12] simulates an increased exponent range, making the assumption of an unbounded exponent range legitimate for the error analysis.

For definiteness, we also assume that

- composite expressions are evaluated from left to right: for instance, $a + bcd$ is computed as $\mathrm{RN}(a + \mathrm{RN}(\mathrm{RN}(bc)d))$,
- the power operation is implemented according to the formulas $x^2 = x \cdot x$, $x^3 = x \cdot x^2$, $x^4 = (x^2)^2$, $x^6 = (x^2)^3$, and $x^8 = (x^4)^2$,
- the exponential function used at steps 7 and 15 is faithfully rounded, implying that the corresponding relative error is bounded by $2u$.

These assumptions are not critical and our bounds easily adapt to slightly different implementations.

We denote by $\tilde{x}$ the computed value of a quantity x. To express the relation between $\tilde{x}$ and x, we use the θ_k and γ_k notation of [8, Chap. 3]. In short, each occurrence of the symbol θ_k denotes a potentially different quantity of the form $\theta_k = \prod_{i=1}^{k}(1+r_i)^{\pm 1} - 1$ with $|r_i| \leqslant u$ for all i. Assuming $ku < 1$, one has $|\theta_k| \leqslant \gamma_k$ where γ_k is defined as $ku/(1-ku)$ and satisfies in particular $\gamma_k = ku + O(u^2)$ as $u \to 0$.

3.2 Bounds for loop-independent parameters

Let us first bound the rounding errors occurring in lines 1–7 of Algorithm 1. For instance, using Equation (8), one has

$$\tilde{p} = \mathrm{RN}(1/\mathrm{RN}(\sigma_y \cdot \sigma_y))/2 = p(1+\theta_2), \tag{9}$$

which gives an absolute error bound of $|p - \tilde{p}| \leqslant \gamma_2 p$. Similar bounds for the other parameters are summarized below.

Param.	p	ω_x	ω_y	ϕ	Q_1	Q_2	Q_3
Abs. Err.	$\gamma_2 p$	$\gamma_5\omega_x$	$\gamma_5\omega_y$	γ_4	$\gamma_9 Q_1^\sharp$	$\gamma_{18} Q_2^\sharp$	$\gamma_{24} Q_3^\sharp$
Param.	P_0	P_1	P_2	P_3	c_0	e^{-pR^2}	
Abs. Err.	$\gamma_{10} P_0^\sharp$	$\gamma_{18} P_1^\sharp$	$\gamma_{27} P_2^\sharp$	$\gamma_{32} P_3^\sharp$	$e_0 c_0$	τe^{-pR^2}	

Table 1: Absolute rounding error bounds for the parameters.

Proposition 1. *The absolute FP rounding error for the parameters appearing in lines 1–7 and the preconditionner* $\exp(-pR^2)$ *in line 15 in Algorithm 1 is bounded as indicated in Table 1, where* $P_i^\sharp := P_i\{\phi \leftarrow 1\}$, $Q_i^\sharp := Q_i\{\phi \leftarrow 1\}$ *denote the values for* $\phi = 1$ *of the* P_i *and* Q_i*:*

$$\begin{aligned}
&Q_1^\sharp = 3pR^2, && Q_2^\sharp = 3p^2R^4, && Q_3^\sharp = p^3R^6,\\
&P_0^\sharp = \left(\tfrac{3}{2}p + \omega_x + \omega_y\right) R^2, && P_1^\sharp = \left(3p + \omega_x + 3\omega_y\right) pR^4,\\
&P_2^\sharp = \left(\tfrac{3}{2}p + 3\omega_y\right) p^2R^6, && P_3^\sharp = p^3\omega_y R^8,
\end{aligned}$$

$$\begin{aligned}
e_0 &= \exp\left[\frac{1}{2}\left(\frac{x_m^2}{\sigma_x^2} + \frac{y_m^2}{\sigma_y^2}\right)\gamma_4\right](1+\gamma_6) - 1,\\
\tau &= \exp\left[pR^2\gamma_2\right](1+\gamma_2) - 1.
\end{aligned} \tag{10}$$

Proof. Similarly to Equation (9) one obtains $\tilde{\omega}_x = \omega_x(1+\theta_5)$ and $\tilde{\omega}_y = \omega_y(1+\theta_5)$.

Concerning ϕ, firstly observe that since $0 < \sigma_y \leqslant \sigma_x$, one has $0 < \sigma_y/\sigma_x \leqslant 1$ and because RN preserves inequalities, $0 < \tilde{a} := \mathrm{RN}(\mathrm{RN}(\sigma_y/\sigma_x) \cdot \mathrm{RN}(\sigma_y/\sigma_x)) \leqslant 1$, implying that $0 \leqslant b := 1 - \tilde{a} < 1$. Now, $\tilde{\phi} = \mathrm{RN}(b)$, so that $0 \leqslant \tilde{\phi} \leqslant 1$ and

$$|\phi - \tilde{\phi}| \leqslant |\phi - b| + |b - \tilde{\phi}| \leqslant |\sigma_y^2/\sigma_x^2 - \tilde{a}| + u \leqslant \gamma_3 + \gamma_1 \leqslant \gamma_4.$$

Regarding Q_1, one has $\tilde{Q}_1 = pR^2(2\tilde{\phi}+1)(1+\theta_6)$, and hence, using the previous bounds on ϕ and $|\phi - \tilde{\phi}|$,

$$\begin{aligned}
|Q_1 - \tilde{Q}_1| &= |pR^2(2\phi+1) - pR^2(2\tilde{\phi}+1)(1+\theta_6)|\\
&\leqslant 2pR^2|\phi - \tilde{\phi}| + pR^2|(2\tilde{\phi}+1)|\gamma_6\\
&\leqslant 2pR^2\gamma_4 + 3pR^2\gamma_6 \leqslant 3pR^2\gamma_9.
\end{aligned}$$

The last inequality is readily obtained by applying the rules given in Lemma 3.3 of the reference [8, Chap. 3.4]. The case of the other parameters P_i and Q_i is similar; see Appendix A.1 for detailed proofs.

For c_0, denote $z = -\frac{1}{2}\left(\frac{x_m^2}{\sigma_x^2} + \frac{y_m^2}{\sigma_y^2}\right)$. Then $\tilde{z} = z(1+\theta_4)$, and $\tilde{c}_0 = \frac{R^2}{2\sigma_x\sigma_y}\left(e^{\tilde{z}}(1+\theta_2)\right)(1+\theta_4) = c_0 e^{z\theta_4}(1+\theta_6)$. □

3.3 Local error analysis

Let us now turn to the computation of c_n (steps 7 –12 of Algorithm 1). We denote by $\tilde{c}_n$ the computed value of c_n, and we call *local* absolute error on c_n the absolute rounding error ε_n generated at the corresponding step of the algorithm. In other words, for $n \geqslant 4$, we set

$$\begin{aligned}\varepsilon_n := \tilde{c}_n - \Bigg(&\frac{Q_1(n-1)+P_0}{(n+1)n}\tilde{c}_{n-1} - \frac{Q_2(n-2)+P_1}{(n+1)n^2}\tilde{c}_{n-2} \\ &+ \frac{Q_3(n-3)+P_2}{(n+1)n^2(n-1)}\tilde{c}_{n-3} - \frac{P_3}{(n+1)n^2(n-1)(n-2)}\tilde{c}_{n-4}\Bigg)\end{aligned} \tag{11}$$

(where all operations are mathematically exact). We then have the following bound on $|\varepsilon_n|$.

Proposition 2. *The local error introduced at iteration n at step 12 of Algorithm 1 satisfies*

$$\begin{aligned}|\varepsilon_n| \leqslant \gamma\Bigg(&\frac{Q_1^\sharp(n-1)+P_0^\sharp}{(n+1)n}|\tilde{c}_{n-1}| + \frac{Q_2^\sharp(n-2)+P_1^\sharp}{(n+1)n^2}|\tilde{c}_{n-2}| \\ &+ \frac{Q_3^\sharp(n-3)+P_2^\sharp}{(n+1)n^2(n-1)}|\tilde{c}_{n-3}| \\ &+ \frac{P_3^\sharp}{(n+1)n^2(n-1)(n-2)}|\tilde{c}_{n-4}|\Bigg),\end{aligned}$$

where $\gamma = \gamma_{40}$.

Proof. The coefficient c_n is computed as $\tilde{c}_n = (((t_1 + t_2)(1+\theta_1) + t_3)(1+\theta_1) + t_4)(1+\theta_1)$ with

$$t_i = (-1)^{i+1}\frac{(\tilde{Q}_i(n-i)(1+\theta_1) + \tilde{P}_{i-1})(1+\theta_1)}{d_i(n)(1+\theta_i)^{-1}}\tilde{c}_{n-i}(1+\theta_2),$$

where $d_1(n) = (n+1)n, d_2(n) = (n+1)n^2, \ldots$ are the denominators appearing in (11), and $Q_4 = 0$. We thus have

$$\tilde{c}_n = \sum_{i=1}^{4}\frac{(\tilde{Q}_i(n-i)(1+\theta_9) + \tilde{P}_{i-1}(1+\theta_8)}{d_i(n)}.$$

Substituting into (11), we obtain

$$\begin{aligned}\varepsilon_n = \sum_{i=1}^{4}\frac{(-1)^{i+1}\tilde{c}_{n-i}}{d_i(n)}\big((&\tilde{Q}_i - Q_i + \tilde{Q}_i\theta_9)(n-i) \\ &+ (\tilde{P}_{i-1} - P_{i-1} + \tilde{P}_{i-1}\theta_8)\big).\end{aligned}$$

According to Table 1, we have $|\tilde{Q}_i - Q_i| \leqslant Q_i^\sharp\gamma_{24}$ (with the convention that $Q_4^\sharp = 0$) and $|\tilde{Q}_i\theta_9| \leqslant (1+\gamma_{24})Q_i^\sharp\gamma_9$, so that $|\tilde{Q}_i - Q_i + \tilde{Q}_i\theta_9| \leqslant Q_i^\sharp\gamma_{33}$. Similarly, we have $|\tilde{P}_{i-1} - P_{i-1} + \tilde{P}_{i-1}\theta_8| \leqslant P_{i-1}^\sharp\gamma_{40}$, and the result follows. □

Since the formulas used for computing c_1, c_2, c_3 correspond to truncated instances of the recurrence, (11) also applies for $\varepsilon_1, \varepsilon_2, \varepsilon_3$, if terms with a zero denominator are ignored. With this convention the bound from Prop. 2 holds for all $n \geqslant 1$.

4 Global Rounding Error Bounds

Let us apply the generating series approach of [11] to the rounding error analysis of the main loop.

4.1 Global error modeling

The local errors ε_n build up and lead to a *global* (absolute) error

$$\delta_n := c_n - \tilde{c}_n,$$

that is the main quantity we need to control. For doing so it is convenient to encode the sequences (δ_n) and (ε_n) as coefficients in the generating series $\hat{\delta}(\lambda) = \sum_{n=0}^{+\infty}(n+1)!\delta_n\lambda^n$ and $\hat{\varepsilon}(\lambda) = \sum_{n=0}^{+\infty}(n+1)!\varepsilon_n\lambda^n$ in the same manner as (2).

From Equations (5) and (11), we have

$$\begin{aligned} n\delta_n = &\frac{Q_1(n-1)+P_0}{(n+1)}\delta_{n-1} - \frac{Q_2(n-2)+P_1}{(n+1)n}\delta_{n-2} \\ &+ \frac{Q_3(n-3)+P_2}{(n+1)n(n-1)}\delta_{n-3} - \frac{P_3}{(n+1)n(n-1)(n-2)}\delta_{n-4} - n\varepsilon_n. \end{aligned}$$

After multiplying this relation by λ^n and summing over n, we obtain, using the identities(7), a LDE satisfied by the series $\hat{\delta}$:

$$Q(\lambda)(\hat{\delta})'(\lambda) - P(\lambda)\hat{\delta}(\lambda) = \hat{\varepsilon}'(\lambda). \tag{12}$$

Comparing with Equation (6), we see that $\hat{\delta}$ satisfies the same first-order LDE as the generating series $\hat{f}$ of the (exact) coefficients c_n, except for the right hand side which now depends on the local errors ε_n. Only bounds are available for these, so we need to work with differential inequalities.

Given two series $a(\lambda) = \sum_{n=0}^{+\infty} a_n\lambda^n$ and $b(\lambda) = \sum_{n=0}^{+\infty} b_n\lambda^n$, denote by $a(\lambda) \ll b(\lambda)$ the fact that $|a_n| \leqslant b_n$ for all $n \geqslant 0$. In particular, this implies that the b_n coefficients are nonnegative real numbers. We denote by $|a|(\lambda) = \sum_{n=0}^{+\infty} |a_n|\lambda^n$ the series of absolute values of coefficients.

Proposition 3 (Corollary of Proposition 2)**.** *The generating series of local errors satisfies this differential inequality:*

$$\begin{aligned}\hat{\varepsilon}'(\lambda) \ll \gamma \Big(\big(Q^\sharp(\lambda)\varphi(\lambda) + P^\sharp(\lambda)\big)\, \hat{f}(\lambda) + \\ Q^\sharp(\lambda)|\hat{\delta}|'(\lambda) + P^\sharp(\lambda)|\hat{\delta}|(\lambda)\Big),\end{aligned} \tag{13}$$

with $Q^\sharp(\lambda) = Q_1^\sharp\lambda + Q_2^\sharp\lambda^2 + Q_3^\sharp\lambda^3$ *and* $P^\sharp(\lambda) = P_0^\sharp + P_1^\sharp\lambda + P_2^\sharp\lambda^2 + P_3^\sharp\lambda^3 + P_4^\sharp\lambda^4$.

Proof. It follows by using the inequality $|\tilde{c}_n| \leqslant c_n + |\delta_n|$ in the bound on ε_n obtained from Proposition 2, which is to be multiplied by $n\lambda^n$, and summed over n. □

Located in the Laplace plane, Eqs. (12) and (13) allow for deriving bounds on $\hat{\delta}(\lambda)$. They are obtained as solutions of order-1 LDE. But there is still a need to bound the inverse Laplace transform $\delta(\xi) = \sum_{n=0}^{\infty} \delta_n \xi^{n+1}$. In particular, we need to bound the total sum of absolute rounding errors $|\delta|(1)$. This is done by an *ad-hoc* majorization of convolution terms. For clarity, we provide in Table 2 a synthesis of forthcoming notations in this twofold view.

4.2 A simplified bound

For the sake of exposition, we first prove a simplified, not fully rigorous error bound obtained by neglecting the terms involving $\gamma\hat{\delta}(\lambda)$ in (13), which are of order $O(u^2)$:

$$Q(\lambda)\hat{\delta}'(\lambda) - P(\lambda)\hat{\delta}(\lambda) \ll \gamma \big(Q^\sharp(\lambda)\varphi(\lambda) + P^\sharp(\lambda)\big)\, \hat{f}(\lambda).$$

To further simplify this equation, we denote $\varphi^\sharp(\lambda) := \varphi(\lambda)\{\phi \leftarrow 1\}$ and use $0 \ll \varphi(\lambda) \ll \varphi^\sharp(\lambda)$ and $0 \ll Q(\lambda)^{-1} \ll Q(\lambda)^{-1}\{\phi \leftarrow 1\} = (1 - pR^2\lambda)^{-3}$ (this follows directly from Equation (4)) to obtain

$$\begin{aligned}|\hat{\delta}|'(\lambda) &\ll \varphi(\lambda)|\hat{\delta}|(\lambda) + \gamma\hat{\psi}(\lambda)\hat{f}(\lambda), \\ \text{with } \hat{\psi}(\lambda) &:= \frac{Q^\sharp(\lambda)\varphi^\sharp(\lambda) + P^\sharp(\lambda)}{(1 - pR^2\lambda)^3} \gg 0.\end{aligned} \tag{14}$$

Solving this differential inequality gives rise to the following simplified bound.

Proposition 4. *Under the simplified model above, the total rounding error accumulated while computing* $f(1)$ *satisfies*

$$\sum_{n=0}^{+\infty} |\tilde{c}_n - c_n| = |\delta|(1) \leqslant (e_0 + \gamma C) f(1),$$

'Real' (ξ)	'Laplace at ∞' (λ)
$f(\xi) = \sum_{n=0}^{\infty} c_n \xi^{n+1}$	$\hat{f}(\lambda) = \sum_{n=0}^{\infty} c_n (n+1)! \lambda^n$ $\hat{f}(\lambda) = \lambda^{-2} \mathcal{L}_f(\lambda^{-1})$ $Q\hat{f}' - P\hat{f} = 0$
$\delta(\xi) = \sum_{n=0}^{\infty} \delta_n \xi^{n+1}$ $\lvert\delta\rvert(1) = \sum_{n=0}^{\infty} \lvert c_n - \tilde{c}_n \rvert$	$\hat{\delta}(\lambda) = \sum_{n=0}^{\infty} \delta_n (n+1)! \lambda^n$ $\hat{\delta}(\lambda) = \lambda^{-2} \mathcal{L}_\delta(\lambda^{-1})$ $Q\hat{\delta}' - P\hat{\delta} = \hat{\varepsilon}'$
$\varepsilon(\xi) = \sum_{n=0}^{\infty} \varepsilon_n \xi^{n+1}$	$\hat{\varepsilon}(\lambda) = \sum_{n=0}^{\infty} \varepsilon_n (n+1)! \lambda^n$ $\hat{\varepsilon}' \ll \gamma \left(\left(Q^\sharp \hat{f}' + P^\sharp \hat{f} \right) + \left(Q^\sharp \hat{\delta}' + P^\sharp \hat{\delta} \right) \right)$
$\delta(\xi) \ll \Delta(\xi)$	$\hat{\delta} \ll \hat{\Delta}$ $\hat{\Delta}' = \varphi \hat{\Delta} + \frac{\gamma}{Q} \left(Q^\sharp \left(\hat{f}' + \hat{\Delta}' \right) + P^\sharp \left(\hat{f} + \hat{\Delta} \right) \right)$
$\Delta = e_0 f + \sum_{k=0}^{\infty} \frac{\gamma^k}{k!} \underbrace{\Psi * \ldots * \Psi}_{k \text{ times}} * f$ $\Psi(\xi) \leqslant W(\xi) e^{pR^2 \xi}$	$\hat{\Delta} = \hat{e} \hat{f}$ $\hat{e} = e_0 + (1 + e_0) \sum_{k=0}^{\infty} \frac{\gamma^k}{k!} \hat{\Psi}^k$
$\Delta = e_0 f + \gamma \Psi * f$ $\lvert\delta\rvert(1) \leqslant \Delta(1)$	Order 1 approx of $\hat{e}$ $\hat{e} = e_0 + \gamma \hat{\Psi}$

Table 2: Main properties of real and Laplace plane series.

with e_0 given in (10), $\gamma = \gamma_{40}$ *and*

$$C := \frac{7}{96}p^3\omega_x R^8 + \left(\frac{7}{12}p + \frac{1}{2}\omega_x\right)p^2R^6 + \left(\frac{9}{4}p + \frac{5}{4}\omega_x + \frac{15}{4}\omega_y\right)pR^4 + \left(\frac{3}{2}p + \omega_x + 3\omega_y\right)R^2. \tag{15}$$

Proof. Since all the series on the right-hand side of (14) have nonnegative coefficients, Lemma 6.5 in [11] implies that $|\hat{\delta}|(\lambda) \ll \hat{\Delta}(\lambda)$ where $\hat{\Delta}(\lambda)$ is the solution with $\hat{\Delta}(0) = e_0\hat{f}(0) \geqslant |\delta(0)|$ of the LDE

$$\hat{\Delta}'(\lambda) = \varphi(\lambda)\hat{\Delta}(\lambda) + \gamma\hat{\psi}(\lambda)\hat{f}(\lambda). \tag{16}$$

Using $\hat{f}$ as a solution of the homogeneous part of (16),

$$\hat{\Delta}(\lambda) = \left(e_0 + \gamma\hat{\Psi}(\lambda)\right)\hat{f}(\lambda), \quad \hat{\Psi}(\lambda) := \int_0^\lambda \hat{\psi}(\sigma)\mathrm{d}\sigma.$$

This is a bound on $|\hat{\delta}|$, in the Laplace plane. To go back to $|\delta|$ and obtain an inequality $|\delta|(\lambda) \ll \Delta(\lambda)$, consider the series Δ and Ψ defined by $\hat{\Delta}(\lambda) = \lambda^{-2}\mathcal{L}_\Delta(\lambda^{-1})$ and $\hat{\Psi}(\lambda) = \mathcal{L}_\Psi(\lambda^{-1})$ (with no λ^{-2} factor in the latter). Standard Laplace transform theory, see [17, Chap. 5, §8], gives

$$\Delta(\xi) = e_0 f(\xi) + \gamma\,(\Psi * f)(\xi),$$

with $(\Psi * f)(\xi) = \int_0^\xi \Psi(\tau)f(\xi-\tau)\mathrm{d}\tau$ the convolution of Ψ and f.

A technical but straightforward computation[1] (see Lemma 3 in Appendix A.2) shows that the series $\Psi(\xi)$ can be bounded as $\Psi(\xi) \ll W(\xi)e^{pR^2\xi}$, where $W(\xi)$ is an explicit polynomial of degree 3 in ξ with nonnegative coefficients and $\int_0^1 W(\tau)\mathrm{d}\tau$ is equal to the constant C defined in (15). It follows that

$$\begin{aligned}(\Psi * f)(1) &\leqslant \int_0^1 W(\tau)e^{pR^2\tau}e^{pR^2(1-\tau)}g(1-\tau)\mathrm{d}\tau \\ &\leqslant e^{pR^2}g(1)\int_0^1 W(\tau)\mathrm{d}\tau = Cf(1),\end{aligned}$$

(where the second inequality uses the fact that g is nondecreasing), and therefore

$$|\delta|(1) \leqslant \Delta(1) = e_0 f(1) + \gamma\,(\Psi * f)(1) \leqslant (e_0 + \gamma C)f(1). \qquad \square$$

[1] Maple™ worksheet used for these computations given in Appendix A.3.

4.3 A rigorous bound

For a fully rigorous bound on $\hat{\delta}(\lambda)$, we consider again the differential inequalities (12) and (13). Reasoning as in the previous section, we have

$$|\hat{\delta}|'(\lambda) \ll \gamma \frac{Q^\sharp(\lambda)}{Q(\lambda)} |\hat{\delta}|'(\lambda) + \left(\varphi(\lambda) + \gamma \frac{P^\sharp(\lambda)}{Q(\lambda)} \right) |\hat{\delta}|(\lambda) + \gamma \frac{Q^\sharp(\lambda)\varphi(\lambda) + P^\sharp(\lambda)}{Q(\lambda)} \hat{f}(\lambda),$$

where the coefficients of $|\delta|$ and $|\delta|'$ as well as the inhomogeneous term are series with nonnegative coefficients. Since $Q^\sharp(0) = 0$, Lemma 6.5 in [11] applies again and shows that $\hat{\delta}(\lambda) \ll \hat{\Delta}(\lambda)$, with $\hat{\Delta}(\lambda) \gg 0$ satisfying the LDE

$$\hat{\Delta}'(\lambda) = \varphi(\lambda)\hat{\Delta}(\lambda) + \frac{\gamma}{Q(\lambda)} \Big(Q^\sharp(\lambda) \left(\hat{f}'(\lambda) + \hat{\Delta}'(\lambda) \right) + P^\sharp(\lambda) \left(\hat{f}(\lambda) + \hat{\Delta}(\lambda) \right) \Big).$$

Let us write $\hat{\Delta}(\lambda) = \hat{e}(\lambda)\hat{f}(\lambda)$ and use (3) to obtain a LDE satisfied by $\hat{e}$, where the right-hand side is a positive series:

$$\left(1 - \gamma \frac{Q^\sharp(\lambda)}{Q(\lambda)} \right) \hat{e}'(\lambda) = \gamma \left(\frac{Q^\sharp(\lambda)}{Q(\lambda)} \varphi(\lambda) + \frac{P^\sharp(\lambda)}{Q(\lambda)} \right) (1 + \hat{e}(\lambda)).$$

Since $a(\lambda) := \gamma Q^\sharp(\lambda)/Q(\lambda)$ satisfies $a(0) = 0$ and $a(\lambda) \gg 0$,

$$\frac{1}{1 - a(\lambda)} = \frac{Q(\lambda)}{Q(\lambda) - \gamma Q^\sharp(\lambda)} \gg 0,$$

by composition of two series with nonnegative coefficients. After multiplication by this series, we obtain

$$\hat{e}'(\lambda) = \gamma \frac{Q^\sharp(\lambda)\varphi(\lambda) + P^\sharp(\lambda)}{Q(\lambda) - \gamma Q^\sharp(\lambda)} (1 + \hat{e}(\lambda)). \tag{17}$$

This LDE has several poles due to the perturbation $\gamma Q^\sharp(\lambda)$ of the denominator in the right-hand side. To overcome this additional difficulty, we use the following lemma, proved in Appendix A.2, to obtain a unique pole, at the price of a slight increase in the parameter p. This is a key point for adapting the proof of Proposition 4 to the current setting.

Lemma 2. *Assuming $7\gamma < 1$, we have*

$$\frac{1}{Q(\lambda) - \gamma Q^\sharp(\lambda)} \ll \frac{1}{(1 - p^+ R^2 \lambda)^3} \text{ with } p^+ := \frac{p}{1 - \sqrt[3]{7\gamma}}.$$

This result allows us to bound the solution of LDE (17) by the solution of the simpler LDE

$$\hat{e}'(\lambda) = \gamma\hat{\psi}^+(\lambda)(1+\hat{e}(\lambda)), \qquad \hat{e}(0) = e_0, \tag{18}$$

where $\hat{\psi}^+(\lambda) := \hat{\psi}(\lambda)\{p \leftarrow p^+\}$ with $\hat{\psi}(\lambda)$ defined in (14).

Proposition 5. *The terms $\tilde{c}_n$ computed in FP arithmetic in the main loop of Algorithm 1 satisfy:*

$$\sum_{n=0}^{+\infty} |\tilde{c}_n - c_n| = |\delta|(1) \leqslant \Big(e_0 + (1+e_0)e^{\eta p R^2}(e^{\gamma C^+} - 1)\Big) f(1),$$

with $\gamma = \gamma_{40}$, e_0 as in (10), $\eta := \dfrac{\sqrt[3]{7\gamma}}{1-\sqrt[3]{7\gamma}}$ and $C^+ := C\{p \leftarrow p^+\}$.

Proof. Denoting $\hat{\Psi}^+(\lambda) := \displaystyle\int_0^\lambda \hat{\psi}^+(\sigma)\mathrm{d}\sigma$, LDE (18) gives

$$\hat{e}(\lambda) = (1+e_0)e^{\gamma\hat{\Psi}^+(\lambda)} - 1 = e_0 + (1+e_0)\sum_{k=1}^{+\infty} \frac{\gamma^k \hat{\Psi}^+(\lambda)^k}{k!}.$$

This gives an explicit expression for $\hat{\Delta}(\lambda) = \hat{e}(\lambda)\hat{f}(\lambda)$.

To obtain $\Delta(\xi)$ s.t. $\hat{\Delta}(\lambda) = \lambda^{-2}\mathcal{L}_\Delta(\lambda^{-1})$, let $\Psi^+(\xi)$ be the series such that $\hat{\Psi}^+(\lambda) = \mathcal{L}_{\Psi^+}(\lambda^{-1})$. Then Laplace transform rules give the following identity of formal power series:

$$\Delta(\xi) = e_0 f(\xi) + (1+e_0)\sum_{k=1}^{+\infty} \frac{\gamma^k}{k!}\big(\Psi^{+*k} * f\big)(\xi), \tag{19}$$

where $\Psi^{+*k} = \Psi^+ * \cdots * \Psi^+$ (k times). In Lemma 4, Appendix A.2, we prove the remaining inequality:

$$(\Psi^{+*k} * f)(1) \leqslant e^{\eta p R^2}(C^+)^k f(1). \qquad \square$$

4.4 The final rounding error bound

The truncated series approximation $\mathcal{P}_{0:N} = e^{-pR^2}s = e^{-pR^2}\sum_{n=0}^{N-1} c_n$ of $\mathcal{P}$ is obtained by evaluating the sum $\sum_{n=0}^{N-1}\tilde{c}_n$ in FP arithmetic and by multiplying the result $\tilde{s}$ with e^{-pR^2}. We call $\tilde{\mathcal{P}}_{0:N}$ the FP number returned by Algorithm 1.

The following theorem provides a relative rounding error bound w.r.t. $\mathcal{P}$. Adding to this bound a relative truncation error bound on $|\mathcal{P}_{0:N} - \mathcal{P}|/\mathcal{P}$ derived from [12, §III.C] would yield a total error bound on $|\tilde{\mathcal{P}}_{0:N} - \mathcal{P}|/\mathcal{P}$.

Theorem 1. *The total rounding error is bounded by*

$$\frac{|\tilde{\mathcal{P}}_{0:N} - \mathcal{P}_{0:N}|}{\mathcal{P}} \leqslant (1+\gamma_N)(1+\tau)(1+e_0)\left(1 + e^{\eta p R^2}(e^{\gamma C^+} - 1)\right) - 1,$$

with $\gamma = \gamma_{40}$, and where the quantities e_0, τ are defined in Proposition 1, and η, C^+ are defined in Proposition 5.

The first-order error approximation in the roundoff unit u for this bound on $|\tilde{\mathcal{P}}_{0:N} - \mathcal{P}_{0:N}|/\mathcal{P}$ is

$$\left(N + 8 + 2pR^2 + \frac{2x_m^2}{\sigma_x^2} + \frac{2y_m^2}{\sigma_y^2} + 40C\right) u, \tag{20}$$

where $40Cu$ is the dominant term for large p, R, x_m and y_m.

Proof. Denote $\overline{s} = \sum_{n=0}^{N-1} \tilde{c}_n$ and $\tilde{s}$ its FP evaluation using $N-1$ additions. By Proposition 5, $|\overline{s} - s| \leqslant \sum_{n=0}^{N-1} |\tilde{c}_n - c_n| \leqslant \nu f(1)$ where $\nu := e_0 + (1 + e_0)e^{\eta p R^2}(e^{\gamma C^+} - 1)$. Then $|\tilde{s} - \overline{s}| \leqslant \gamma_{N-1} \sum_{n=0}^{N-1} |\tilde{c}_n| \leqslant \gamma_{N-1}(1+\nu) f(1)$. Combining these two bounds yields $|\tilde{s} - s| \leqslant (\gamma_{N-1} + \nu + \gamma_{N-1}\nu) f(1)$.

Finally, the relative errors $|\tau'| \leqslant \tau$ and $|\theta_1| \leqslant u$ induced by the evaluation of $\exp(-pR^2)$ and the multiplication by $\tilde{s}$ give

$$\begin{aligned}
|\tilde{\mathcal{P}}_{0:N} - \mathcal{P}_{0:N}| &= |e^{-pR^2}\tilde{s}(1+\tau')(1+\theta_1) - e^{-pR^2}s| \\
&\leqslant e^{-pR^2}\Big(s(\tau + u + \tau u) + |\tilde{s} - s|(1+\tau)(1+u)\Big) \\
&\leqslant \mathcal{P}\Big(\tau + u + \tau u + (\gamma_{N-1} + \nu + \gamma_{N-1}\nu)(1+\tau)(1+u)\Big) \\
&\leqslant \mathcal{P}\Big((1+\gamma_N)(1+\tau)(1+\nu) - 1\Big),
\end{aligned}$$

which is exactly the bound claimed by Theorem 1. □

5 Preventing Overflows and Underflows

Algorithm 1 may be subject to *overflows* and *underflows*, depending on the problem parameters and the number N of terms to be computed. First, according to [12, §III.C, Prop. 4], this number N has to be at least $2e(p + \omega_x + \omega_y)R^2$ to obtain a reasonable approximation of $\mathcal{P}$. Since the c_n sum to $f(1) = e^{pR^2}\mathcal{P}$ and $\mathcal{P}$ may be close to 1, the use of plain IEEE 754-1985

binary64 FP arithmetic with maximum exponent 1023 may cause overflows for examples requiring more than $2e\ln(2^{1023}) \approx 4000$ terms.

In the C implementation of [12], a rescaling strategy is used to prevent overflows and underflows. At the end of each iteration, if the absolute value of the computed term $\tilde{c}_n$ is above A or below A^{-1}, for $A = 2^{800}$, then the values of $\tilde{c}_n, \tilde{c}_{n-1}, \tilde{c}_{n-2}, \tilde{c}_{n-3}$ are rescaled by 2^k for some k so that their absolute values belong to $[A^{-1}, A]$. This number k is added to a signed 64-bit integer used to "store" the current exponent, and the summation of the terms $\tilde{c}_n$ keeps track of these intermediate rescalings. Two additional rescalings are also used for c_0 and the final factor e^{-pR^2} to prevent underflows. Note that this rescaling strategy, where the exponent is stored separately in a 64-bit integer, does not modify the relative rounding error model used in the previous sections.

The following theorem guarantees the absence of overflows under reasonable assumptions on the size of input parameters.

Theorem 2. *We assume that the number N of terms required to approximate $\mathcal{P}$ is bounded by $N_* = 10^8$, and that this bound also holds for pR^2, $\omega_x R^2$ and $\omega_y R^2$. In addition, considering the size of a satellite and its distance to the space debris, we assume $1 \leqslant R \leqslant 10^3$ and $\sigma_x, \sigma_y, |x_m|, |y_m| \leqslant 10^6$ (all these quantities are expressed in meters).*

Then the execution of the C implementation of Algorithm 1 on $\sigma_x, \sigma_y, x_m, y_m, R, N$ is not subject to overflows.

Proof. We prove this property for all the steps of the algorithm, postponing to the end of the proof the additional effect of rounding errors.

• *Loop-independent parameters.* By combining the inequalities assumed for the parameters in the theorem, it is straightforward that all the subexpressions involved in the computation of p, ϕ, ω_x, ω_y, Q_i, P_i and c_0 are much smaller than 2^{1023} and do therefore not cause any overflow.

• *Evaluation of c_n for $n \geqslant 1$.* At the beginning of iteration n, the preceding terms c_{n-1}, c_{n-2}, c_{n-3}, c_{n-4} are bounded by $A = 2^{800}$ in absolute value thanks to the rescaling strategy. A quick analysis shows that each of the four coefficients in front of c_{n-i} is bounded by $2N_*^4$, and both their numerator and denominator are bounded by $2N_*^5$. Hence, their evaluation cannot produce overflows. Finally, $c_n \leqslant \left(\sum_{i=1}^{3} Q_i + \sum_{i=0}^{3} P_i\right) A \leqslant 7N_*^4 A \leqslant 2^{910} < 2^{1023}$, so that no overflow can occur.

• *No overflow of the 64-bit exponent.* The total sum is bounded by $f(1) = e^{pR^2}\mathcal{P} \leqslant e^{N_*} \leqslant 2^{2^{20.5}} < 2^{2^{63}-1}$, so that the exponents of all c_n and all partial sums fit in the 64-bit integer. The final multiplication by e^{-pR^2} cannot cause an overflow either since the argument is negative.

• *The effect of rounding errors.* The local rounding errors in the constants and in each iteration of the loop were bounded by small constants (Propositions 1 and 2), hence they do not modify significantly the overflow analysis of the first two items above. The worst-case relative error bound given by Theorem 1 is (crudely) bounded by $\exp\left(\gamma_{N+8} + (2 \cdot 10^{12} + \gamma_2 + \eta)N_* + \gamma C^+\right) \leqslant 2^{2^{61.4}}$. Although huge, this bound is sufficient to prove that the computed sum of the $\tilde{c}_n$ is smaller than $2^{2^{61.4}} \cdot 2^{2^{20.5}} < 2^{2^{63}-1}$. □

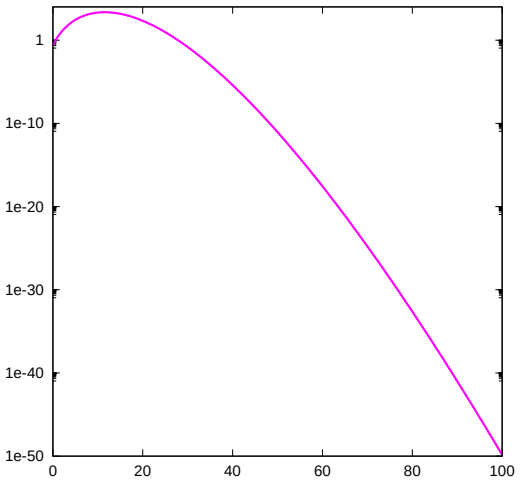
The rigorous underflow analysis is slightly more involved and postponed to future work. Roughly speaking, the rescaling strategy prevents underflows in the exponentials in c_0 and e^{pR^2} (which would cause the output to be zero). Underflows can however occur when unrolling the recurrence, but then it means that the neglected terms c_n are so small compared to the previous ones that this underflow error is smaller than the relative error already computed for the partial sum.

6 Examples and conclusion

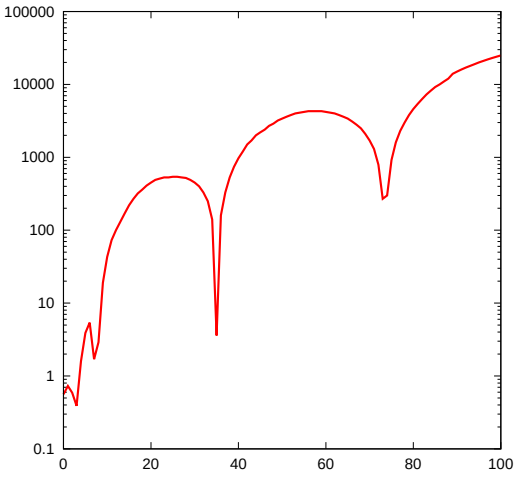
We exemplify the error bounds on the examples provided in [12], together with additional numerically challenging examples that we custom made for illustration purposes.

The numerical behavior of the algorithm is illustrated on Test 1, given in the first line of Table 3, for which 101 terms are computed. In Figure 1a, the magnitude of the coefficients $c_0, \ldots, c_{100}$ is plotted on a log-scale. This is a higher precision 106-bits FP arithmetic computation, using the MPFR library [5], in order to accurately approximate their *exact* values. Their magnitude increases up to c_{16} and then the convergent regime is observed. In Figure 1b the relative rounding error on each coefficient is plotted, when the loop is evaluated with a 53-bit FP arithmetic (this rounding error is estimated by comparing with the *shadow* 106-bit higher precision computation). These errors are plotted in terms of the roundoff unit $u = 2^{-53}$. The corresponding evaluated sum $\tilde{s}$ and probability $\tilde{\mathcal{P}}_{0:N}$ are recalled in Figure 1d. For comparison, we also tested an interval arithmetic implementation with a 53-bit precision interval format, using the MPFI library[2]. While these intervals provide enclosures of all the accumulated rounding errors, we observe in Figure 1c, where the radius of the intervals is plotted in terms of u, that they highly overestimate the actual rounding errors. This is confirmed in 1d: with interval arithmetic, the final absolute enclosure radius is 2.9439e12 $u \simeq$ 3.26e-4, while the *shadowed* absolute error bound is 2.5920e6 $u \simeq$ 2.87e-10.

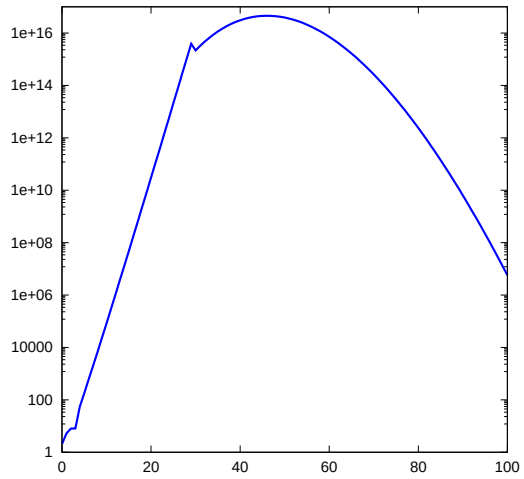
[2] `https://gitlab.inria.fr/mpfi/mpfi`



(a) Coeff. magnitude



(b) Coeff. rel. err. (in terms of u) for 53-bit precision



(c) Interval radius (in terms of u) for MPFI 53-bit precision

Sum $\tilde{s}$	Abs. err. (u)	Rel. err. (u)
2.0521e4	2.5920e6	1.2631e2
Proba $\tilde{\mathcal{P}}_{0:N}$	Abs. err. (u)	Rel. err. (u)
7.6474e-2	9.6505e0	1.2619e2
MPFI Sum mid	Abs. rad. (u)	Rel. rad. (u)
2.0521e4	7.8997e17	3.8496e13
MPFI Proba mid	Abs. rad. (u)	Rel. rad. (u)
7.6474e-2	2.9439e12	3.8496e13

(d) Computed values

Figure 1: Loop evaluation results for Test 1.

Case	Input parameters (m)					N	Relative Error			
#	σ_x	σ_y	R	x_m	y_m		*Exact*	MPFI	Lin. Bound (20)	Bound Thm. 1
Test 1	50	1	5	10	0	101	1.40e-14	4.27e-3	6.72e-12	6.72e-12
Chan 1	50	25	5	10	0	49	5.86e-17	5.86e-15	6.48e-15	6.48e-15
Chan 2	50	25	5	0	10	49	1.50e-16	6.23e-15	6.53e-15	6.53e-15
Chan 3	75	25	5	10	0	49	9.01e-18	4.55e-15	6.47e-15	6.47e-15
Chan 4	75	25	5	0	10	49	1.80e-16	4.88e-15	6.53e-15	6.53e-15
Chan 5	3,000	1,000	10	1,000	0	49	2.02e-16	7.41e-15	6.35e-15	6.35e-15
Chan 6	3,000	1,000	10	0	1,000	48	1.18e-16	5.61e-15	6.44e-15	6.44e-15
Chan 7	3,000	1,000	10	10,000	0	40	3.38e-16	5.45e-15	7.80e-15	7.80e-15
Chan 8	3,000	1,000	10	0	10,000	4	1.53e-14	4.45e-16	2.36e-14	2.36e-14
Chan 9	10,000	1,000	10	10,000	0	46	9.31e-17	4.46e-15	6.22e-15	6.22e-15
Chan 10	10,000	1,000	10	0	10,000	4	1.52e-14	5.57e-14	2.36e-14	2.36e-14
Chan 11	3,000	1,000	50	5,000	0	47	9.92e-17	4.38e-15	6.73e-15	6.73e-15
Chan 12	3,000	1,000	50	0	5,000	4	4.84e-17	1.98e-15	7.10e-15	7.10e-15
CSM 1	152.88	57.91	10.3	60.58	84.87	46	1.57e-17	4.28e-15	6.85e-15	6.85e-15
CSM 2	5,756.84	15.98	1.3	115.05	-81.61	20	6.15e-16	5.48e-15	9.50e-15	9.50e-15
CSM 3	643.40	94.23	5.3	693.40	102.17	45	6.24e-17	6.39e-15	6.43e-15	6.43e-15
Alfano 3	114.25	1.41	15	0.15	-3.88	1627	4.14e-12	1.15e54	7.07e-10	7.08e-10
Alfano 5	177.81	0.03	10	2.12	-1.22	>1e7	4.35e-4	4e69380	4.87e-01	3.60e+00
Custom 1	1	1	10	1	1	543	6.96e-16	1.78e-13	1.53e-09	1.53e-09
Custom 2	1	0.8	10	1	1	969	2.73e-14	4.7e23	5.59e-09	5.60e-09
Custom 3	1	0.5	10	1	1	3805	7.74e-14	4.4e174	8.95e-08	9.00e-08
Custom 4	1	0.2	10	1	1	95139	4.6e-12	2e1483	2.13e-05	2.22e-05
Custom 5	1	0.1	10	1	1	>1e7	3.63e-8	1e6155	1.36e-03	1.59e-03
Custom 6	0.5	0.1	10	1	1	>1e7	1.49e-11	2e5988	1.66e-02	1.95e-02
Custom 7	1	0.05	10	1	1	>1e7	3.0e-6	4e24841	8.68e-02	1.70e-01
Custom 8	0.2	0.05	10	1	1	>1e7	1.28e-9	2e23506	4.05e+01	7.40e+17

Table 3: Relative errors for test cases adapted from [12]: exact, obtained with MPFI, computed with our linearized bound w.r.t. u (20) and with our full bound of Theorem 1.

In Table 3, we computed both the fully rigorous relative error bound of Theorem 1 and its linearization (20) w.r.t. u. For easy examples requiring less than 50 terms, both MPFI and our bounds provide very sharp enclosures. Both of our bounds are almost identical. A rapid increase of interval widths with MPFI is observed when N is larger than 100 in most of the cases, whereas our bounds continue to guarantee at least one correct digit in very hard cases requiring about $N = 10^7$ terms. However, when the ratio σ_y/σ_x becomes very small, like in Case Alfano 5 and Custom 8, the bounds provided are very loose and the rigorous bound deviates from its linearization. As mentioned also in [12], such extremely degenerate cases are rarely occurring in practice (this roughly corresponds to the integration domain becoming uni-dimensional).

Therefore, we believe that the bound provided in this article can be of highly practical use. One can simply evaluate the provided closed-form bound and there is no need to overload the C code with shadowing computations or with an additional execution with interval arithmetic (which would encumber especially the on-board implementation). Our study also shows that the approach of [11] and the additional mathematical tools developed herein are applicable to an algorithm currently used in practice. While described by pen-and-paper, the formulas presented can be easily obtained by a computer algebra software (as shown in the jointed Maple code), so they can be at least partly automated. Future works, include possible refinements concerning underflow handling, an average case analysis in the framework of [9], a formal proof of these results and further generalizations to other implementations of similar mathematical functions like Chi-square densities (which have similar algebraic properties).

References

[1] A. Appel and A. Kellison. Vcfloat2: Floating-point error analysis in coq. In *Proceedings of the 13th ACM SIGPLAN International Conference on Certified Programs and Proofs*, CPP 2024, page 14–29, New York, NY, USA, 2024. Association for Computing Machinery.

[2] R. Barrio, B. Melendo, and S. Serrano. *Journal of computational and applied mathematics*, 150(1):71–86, 2003.

[3] F. Chan. *Spacecraft Collision Probability*. American Institute of Aeronautics and Astronautics, Reston, Virginia, USA, 2008.

[4] A. Das, I. Briggs, G. Gopalakrishnan, S. Krishnamoorthy, and P. Panchekha. Scalable yet rigorous floating-point error analysis. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–14. IEEE, 2020.

[5] L. Fousse, G. Hanrot, V. Lefèvre, P. Pélissier, and P. Zimmermann. Mpfr: A multiple-precision binary floating-point library with correct rounding. *ACM Transactions on Mathematical Software (TOMS)*, 33(2):13–es, 2007.

[6] 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.

[7] 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.

[8] N. Higham. *Accuracy and stability of numerical algorithms*. SIAM, 2002.

[9] N. J. Higham and T. Mary. A new approach to probabilistic rounding error analysis. *SIAM journal on scientific computing*, 41(5):A2815–A2835, 2019.

[10] A. Mathai and S. Provost. *Quadratic forms in random variable: Theory and Applications*. Marcel Dekker, New York, USA, 1992.

[11] M. Mezzarobba. Rounding error analysis of linear recurrences using generating series. *Electronic Transactions on Numerical Analysis*, 58:196–227, 2023.

[12] 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. Code link `https://homepages.laas.fr/mmjoldes/CollisionProba/`.

[13] A. Solovyev, M. S. Baranowski, I. Briggs, C. Jacobsen, Z. Rakamarić, and G. Gopalakrishnan. Rigorous estimation of floating-point round-off errors with symbolic Taylor expansions. *ACM Trans. Program. Lang. Syst.*, 41(1), Dec 2018.

[14] J. Thomassin, S. Laurens, and F. Toussaint. Asteria : Autonomous collision risks management. In *72nd International Astronautical Congress*

(IAC), Dubai, United Arab Emirates, 25-29 October 2021, number IAC-21/A6/7Paper 63266. International Astronautical Federation, 2021. `https://iafastro.directory/iac/paper/id/63266/summary/`.

[15] L. Titolo, M. A. Feliú, M. Moscato, and C. A. Muñoz. An abstract interpretation framework for the round-off error analysis of floating-point programs. In *Verification, Model Checking, and Abstract Interpretation: 19th International Conference, VMCAI 2018, Los Angeles, CA, USA, January 7-9, 2018, Proceedings 19*, pages 516–537. Springer, 2018.

[16] D. Widder. *The Laplace transform*. Princeton Univ. Press, 1946.

[17] D. Widder. *An introduction to transform theory*. Academic Press New York, 1971.

A Appendix

A.1 Complementary proofs for the rounding error analysis of the parameters of Algorithm 1

The following results complete the proof of Proposition 1. Note that we insist on bounding each P_i, Q_i in terms of the corresponding $P_i^\sharp$, $Q_i^\sharp$ for simplicity reasons only, and slightly better bounds can be obtained without this restriction.

Proposition 6. *At step 2 of Algorithm 1, the computed value $\tilde{Q}_2$ of Q_2 satisfies $|Q_2 - \tilde{Q}_2| \leqslant Q_2^\sharp \gamma_{18}$.*

Proof. The value of Q_2 is computed as

$$\tilde{Q}_2 = \mathrm{RN}(\mathrm{RN}(\mathrm{RN}(\mathrm{RN}(\tilde{p}^2) \cdot \mathrm{RN}(\mathrm{RN}(R^2)^2)) \cdot \tilde{\phi}) \cdot \mathrm{RN}(\tilde{\phi} + 2)).$$

By Equation (8), this implies

$$\begin{aligned}\tilde{Q}_2 &= (\tilde{p}R^2(1+\theta_2))^2\tilde{\phi}(\tilde{\phi}+2)(1+\theta_4)\\ &= (p(1+\theta_2)R^2(1+\theta_2))^2\tilde{\phi}(\tilde{\phi}+2)(1+\theta_4)\\ &= p^2R^4\tilde{\phi}(\tilde{\phi}+2)(1+\theta_{12}).\end{aligned}$$

It follows that

$$\tilde{Q}_2 - Q_2 = p^2R^4((2+\tilde{\phi}+\phi)(\tilde{\phi}-\phi) + \tilde{\phi}(\tilde{\phi}+2)\theta_{12})$$

and therefore

$$\begin{aligned}|\tilde{Q}_2 - Q_2| &\leqslant p^2R^4(4\gamma_4 + 3\gamma_{12})\\ &\leqslant Q_2^\sharp\left(\tfrac{4}{3}\gamma_4 + \gamma_{12}\right)\\ &\leqslant Q_2^\sharp\gamma_{18}.\end{aligned}$$

□

Proposition 7. *At step 2 of Algorithm 1, the computed value $\tilde{Q}_3$ of Q_3 satisfies $|Q_3 - \tilde{Q}_3| \leqslant Q_3^\sharp \gamma_{24}$.*

Proof. One has

$$\begin{aligned}\tilde{Q}_3 &= \tilde{p}^3(R^2(1+\theta_1))^3\tilde{\phi}^2(1+\theta_7)\\ &= (p(1+\theta_2))^3(R^2(1+\theta_1))^3\tilde{\phi}^2(1+\theta_7)\\ &= p^3R^6\tilde{\phi}^2(1+\theta_{16}),\end{aligned}$$

hence

$$\tilde{Q}_3 - Q_3 = p^3R^6((\tilde{\phi} - \phi)(\tilde{\phi} + \phi) + \tilde{\phi}^2\theta_{16})$$

and

$$\begin{aligned}|\tilde{Q}_3 - Q_3| &\leqslant p^3R^6[2\gamma_4 + \gamma_{16}]\\ &\leqslant p^3R^6 \cdot \gamma_{24}\\ &= Q_3^\sharp \cdot \gamma_{24}.\end{aligned}$$

□

Proposition 8. *At step 3 of Algorithm 1, the computed value $\tilde{P}_0$ of P_0 satisfies $|P_0 - \tilde{P}_0| \leqslant P_0^\sharp\gamma_{10}$.*

Proof. One has

$$\begin{aligned}\tilde{P}_0 &= ((\tilde{p}(\tfrac{1}{2}\tilde{\phi} + 1)(1 + \theta_2) + \tilde{\omega}_x)(1 + \theta_1) + \tilde{\omega}_y)R^2(1 + \theta_3)\\ &= \{[p(1 + \theta_2)(\tfrac{1}{2}\tilde{\phi} + 1)(1 + \theta_2)\\ &\quad + \omega_x(1 + \theta_5)](1 + \theta_1) + \omega_y(1 + \theta_5)\}R^2(1 + \theta_3)\\ &= \{p(\tfrac{1}{2}\tilde{\phi} + 1)(1 + \theta_8) + \omega_x(1 + \theta_9) + \omega_y(1 + \theta_8)\}R^2,\end{aligned}$$

hence

$$\tilde{P}_0 - P_0 = pR^2(\tfrac{1}{2}(\tilde{\phi} - \phi) + (\tfrac{1}{2}\tilde{\phi} + 1)\theta_8) + \omega_xR^2\theta_9 + \omega_yR^2\theta_8$$

and

$$\begin{aligned}|\tilde{P}_0 - P_0| &\leqslant pR^2(\tfrac{1}{2}\gamma_4 + \tfrac{3}{2}\gamma_8) + (\omega_x + \omega_y)R^2\gamma_9\\ &\leqslant R^2(\tfrac{3}{2}p(\tfrac{1}{3}\gamma_4 + \gamma_8) + (\omega_x + \omega_y)\gamma_9)\\ &\leqslant P_0^\sharp\gamma_{10}.\end{aligned}$$

□

Proposition 9. *At step 4 of Algorithm 1, the computed value $\tilde{P}_1$ of P_1 satisfies $|P_1 - \tilde{P}_1| \leqslant P_1^\sharp\gamma_{18}$.*

Proof. One has

$$\begin{aligned}\tilde{P}_1 &= \{[\tfrac{1}{2}\tilde{p}\tilde{\phi}(\tilde{\phi} + 5)(1 + \theta_3) + \tilde{\omega}_x](1 + \theta_1)\\ &\quad + \tilde{\omega}_y(2\tilde{\phi} + 1)(1 + \theta_2)\}\tilde{p}(R^2(1 + \theta_1))^2(1 + \theta_3)\\ &= \{[\tfrac{1}{2}p\tilde{\phi}(\tilde{\phi} + 5)(1 + \theta_5) + \omega_x(1 + \theta_5)](1 + \theta_1)\\ &\quad + \omega_y(2\tilde{\phi} + 1)(1 + \theta_7)\}pR^4(1 + \theta_7)\\ &= pR^4(\tfrac{1}{2}p\tilde{\phi}(\tilde{\phi} + 5)(1 + \theta_{13}) + \omega_x(1 + \theta_{13})\\ &\quad + \omega_y(2\tilde{\phi} + 1)(1 + \theta_{14}))\end{aligned}$$

hence

$$\begin{aligned}\tilde{P}_1 - P_1 = pR^4(&\tfrac{1}{2}p[\tilde{\phi}^2 - \phi^2 + 5(\tilde{\phi} - \phi) + \tilde{\phi}(\tilde{\phi} + 5)\theta_{13}] \\ &+ \omega_x\theta_{13} + \omega_y(2(\tilde{\phi} - \phi) + (2\tilde{\phi} + 1)\theta_{14}))\end{aligned}$$

and

$$\begin{aligned}|\tilde{P}_1 - P_1| &\leqslant pR^4(\tfrac{1}{2}p[2\gamma_4 + 5\gamma_4 + 6\gamma_{13}] \\ &\quad + \omega_x\gamma_{13} + \omega_y(2\gamma_4 + 3\gamma_{14})) \\ &\leqslant pR^4[3p(\gamma_{13} + \tfrac{7}{6}\gamma_4) + \omega_x\gamma_{13} + 3\omega_y(\gamma_{14} + \tfrac{2}{3}\gamma_4)] \\ &\leqslant P_1^\sharp\gamma_{18}.\end{aligned}$$

□

Proposition 10. *At step 5 of Algorithm 1, the computed value $\tilde{P}_2$ of P_2 satisfies $|P_2 - \tilde{P}_2| \leqslant P_2^\sharp\gamma_{27}$.*

Proof. One has

$$\begin{aligned}\tilde{P}_2 &= (\tfrac{3}{2}\tilde{p}(1 + \theta_1) + \tilde{\omega}_y(\tilde{\phi} + 2)(1 + \theta_2))\tilde{p}^2 \\ &\quad \cdot (R^2(1 + \theta_1))^3\tilde{\phi}(1 + \theta_7) \\ &= (\tfrac{3}{2}p(1 + \theta_3) + \omega_y(\tilde{\phi} + 2)(1 + \theta_7))p^2 \\ &\quad \cdot (R^2(1 + \theta_1))^3\tilde{\phi}(1 + \theta_{11}) \\ &= (\tfrac{3}{2}p(1 + \theta_3) + \omega_y(\tilde{\phi} + 2)(1 + \theta_7))p^2R^6\tilde{\phi}(1 + \theta_{14}) \\ &= p^2R^6(\tfrac{3}{2}p\tilde{\phi}(1 + \theta_{17}) + \omega_y(\tilde{\phi}^2 + 2\tilde{\phi})(1 + \theta_{21}))\end{aligned}$$

hence

$$\begin{aligned}\tilde{P}_2 - P_2 = p^2R^6(&\tfrac{3}{2}p(\tilde{\phi} - \phi) + \tfrac{3}{2}p\tilde{\phi}\theta_{17} \\ &+ \omega_y(\tilde{\phi}^2 - \phi^2 + 2\tilde{\phi} - 2\phi) + \omega_y(\tilde{\phi}^2 + 2\tilde{\phi})\theta_{21})\end{aligned}$$

and

$$\begin{aligned}|\tilde{P}_2 - P_2| &\leqslant p^2R^6(\tfrac{3}{2}p\gamma_4 + \tfrac{3}{2}p\gamma_{17} + 4\omega_y\gamma_4 + 3\omega_y\gamma_{21}) \\ &\leqslant p^2R^6(\tfrac{3}{2}p(\gamma_4 + \gamma_{17}) + 3\omega_y(\tfrac{4}{3}\gamma_4 + \gamma_{21})) \\ &\leqslant P_2^\sharp\gamma_{27}.\end{aligned}$$

□

Proposition 11. *At step 6 of Algorithm 1, the computed value $\tilde{P}_3$ of P_3 satisfies $|P_3 - \tilde{P}_3| \leqslant P_3^\sharp\gamma_{32}$.*

Proof. One has

$$\begin{aligned}\tilde{P}_3 &= \tilde{p}^3\tilde{\omega}_y((R^2(1+\theta_1))^2(1+\theta_1))^2\tilde{\phi}^2(1+\theta_7)\\ &= p^3\omega_y((R^2(1+\theta_1))^2(1+\theta_1))^2\tilde{\phi}^2(1+\theta_{18})\\ &= p^3\omega_y R^8\tilde{\phi}^2(1+\theta_{24}),\end{aligned}$$

hence

$$\tilde{P}_3 - P_3 = p^3\omega_y R^8(\tilde{\phi}^2 - \phi^2 + \tilde{\phi}^2\theta_{24})$$

and

$$\begin{aligned}|\tilde{P}_3 - P_3| &\leqslant p^3\omega_y R^8(2\gamma_4 + \gamma_{24})\\ &\leqslant P_3^\sharp\gamma_{32}.\end{aligned}$$

□

Proposition 12. *At step 15 of Algorithm 1, the computed value* $\tilde{a}$ *of* $\exp(-pR^2)$ *satisfies* $|\exp(-pR^2) - \tilde{a}| \leqslant \tau\exp(-pR^2)$ *with* $\tau = \exp(pR^2\gamma_2)(1+\gamma_2) - 1$.

Proof. Denote $z = -pR^2$. Then $\tilde{z} = z(1+\theta_2)$. This gives

$$\tilde{a} = \exp(\tilde{z})(1+\theta_2) = \exp(z)\exp(z\theta_2)(1+\theta_2).$$

□

A.2 Complementary proofs for the global error analysis

Lemma 3. *Let*

$$\hat{\Psi}(\lambda) = \int_0^\lambda \hat{\psi}(\sigma)\mathrm{d}\sigma$$

with $\hat{\psi}$ *defined in* (14). *Then for all* $\xi \geqslant 0$, *the series* $\Psi(\xi)$ *such that* $\hat{\Psi}(\lambda) = \mathcal{L}_\Psi(\lambda^{-1})$ *satisfies*

$$\Psi(\xi) \ll W(\xi)\, e^{pR^2\xi},$$

where

$$\begin{aligned}W(\xi) = \frac{7}{24}p^3\omega_x R^8\xi^3 + \Big(\frac{7}{4}p + \frac{3}{2}\omega_x\Big)p^2R^6\xi^2 + \Big(\frac{9}{2}p\\ + \frac{5}{2}\omega_x + \frac{15}{2}\omega_y\Big)pR^4\xi + \Big(\frac{3}{2}p + \omega_x + 3\omega_y\Big)R^2.\end{aligned}$$

Proof. Since

$$\hat{\Psi}(\lambda^{-1}) = \int_0^{\lambda^{-1}} \hat{\psi}(\sigma)\mathrm{d}\sigma = \int_\lambda^{+\infty} \frac{\hat{\psi}(\tau^{-1})}{\tau^2}\mathrm{d}\tau,$$

we have

$$\Psi(\xi) = \frac{1}{\xi}\mathcal{L}^{-1}\left\{\frac{\hat{\psi}(\lambda^{-1})}{\lambda^2}\right\}(\xi)$$

using Lemma 5.2 from [17, Chap. 5]. We compute the partial fraction decomposition

$$\begin{aligned}\frac{\hat{\psi}(\lambda^{-1})}{\lambda^2} &= \frac{7p^3\omega_x R^8}{(\lambda - pR^2)^5} + \frac{(21p + 18\omega_x)p^2R^6}{2(\lambda - pR^2)^4} \\ &\quad + \frac{(9p + 5\omega_x + 15\omega_y)pR^4}{(\lambda - pR^2)^3} + \frac{(3p + 2\omega_x - 18\omega_y)R^2}{2(\lambda - pR^2)^2} \\ &\quad + \frac{12\omega_y}{p(\lambda - pR^2)} - \frac{2\omega_y R^2}{\lambda^2} - \frac{12\omega_y}{p\lambda}.\end{aligned}$$

and take the inverse Laplace transform term-by-term, which yields

$$\begin{aligned}\Psi(\xi) &= \Big(\frac{7}{24}p^3\omega_x R^8\xi^3 + \Big(\frac{7}{4}p + \frac{3}{2}\omega_x\Big)p^2R^6\xi^2 + \Big(\frac{9}{2}p + \frac{5}{2}\omega_x \\ &\quad + \frac{15}{2}\omega_y\Big)pR^4\xi + \Big(\frac{3}{2}p + \omega_x - 9\omega_y\Big)R^2\Big)e^{pR^2\xi} \\ &\quad + 12\omega_y\frac{e^{pR^2\xi} - 1}{p\xi} - 2\omega_y R^2.\end{aligned}$$

We conclude by using the inequality $(e^x - 1)/x \ll e^x$ and dropping the final negative term. □

Proof of **Lemma 2**. The denominator $Q(\lambda) - \gamma Q^\sharp(\lambda)$ factors as $(1-\beta_1\lambda)(1-\beta_2\lambda)(1-\beta_3\lambda)$ with all $\beta_i \neq 0$. We prove that

$$|\beta_i| \leqslant p^+R^2 \quad \text{for each } i = 1, 2, 3, \tag{21}$$

which implies $\frac{1}{1-\beta_i\lambda} \ll \frac{1}{1-p^+R^2\lambda}$ and finally the desired result.

Let β denote one of the β_i. If $|\beta| \leqslant pR^2$, then clearly (21) holds. Now suppose $|\beta| > pR^2$. Since $\lambda = \beta^{-1}$ is a root of $Q(\lambda) - \gamma Q^\sharp(\lambda)$, we have $Q(\beta^{-1}) = \gamma Q^\sharp(\beta^{-1})$. We observe:

- $|Q(\beta^{-1})| = |1 - \phi pR^2/\beta|^2|1 - pR^2/\beta| \geqslant (1 - pR^2/|\beta|)^3$;
- $|Q^\sharp(\beta^{-1})| = |3pR^2/\beta + 3p^2R^4/\beta^2 + 3p^3R^6/\beta^3| \leqslant 7$.

We thus obtain $(1 - pR^2/|\beta|)^3 \leqslant 7\gamma$, and (21) holds. □

Lemma 4. *For all $k \geqslant 1$, we have*

$$(\Psi^{+*k} * f)(1) \ll e^{\eta pR^2}(C^+)^k f(1).$$

Proof. Replace the value of p by p^+ in the proof of Lemma 3 to have $\Psi^+(\xi) \ll W^+(\xi)\, e^{p^+R^2\xi}$ and $C^+ = \int_0^1 W^+(\tau)\mathrm{d}\tau$. Also bound $f(\xi)$ by $e^{p^+R^2}g(\xi)$ since $p^+ \geqslant p$. Then we have

$$(\Psi^{+*k} * f)(\xi) \ll ((W^+ e^{p^+R^2\xi})^{*k} * e^{p^+R^2\xi} g)(\xi).$$

For $k = 1$, we have as in the proof of Proposition 4

$$(W^+ e^{p^+R^2\xi} * e^{p^+R^2\xi} g)(\xi) = e^{p^+R^2\xi}(W^+ * g)(\xi).$$

Repeating this process k times to "push" the exponential $e^{p^+R^2\xi}$ out of the convolution product gives

$$\left(\left(W^+ e^{p^+R^2\xi}\right)^{*k} * \left(e^{p^+R^2\xi} g\right)\right)(\xi) = e^{p^+R^2\xi}(W^{+*k} * g)(\xi).$$

Evaluating the series at $\xi = 1$ gives:

$$(\Psi^{+*k} * f)(1) \leqslant e^{p^+R^2} g(1) \int_0^1 W^{+*k}(\tau)\mathrm{d}\tau \leqslant e^{\eta p R^2}(C^+)^k f(1),$$

using $C^+ = \int_0^1 W^+(\tau)\mathrm{d}\tau$ and the fact that for $h_1, h_2 \geqslant 0$, $\int_0^1 (h_1 * h_2)(\tau)\mathrm{d}\tau \leqslant (\int_0^1 h_1(\tau)\mathrm{d}\tau)(\int_0^1 h_2(\tau)\mathrm{d}\tau)$ by Fubini's theorem. □

A.3 Maple™ worksheet to verify the computations in Proposition 4 and Lemma 3

```
> restart:
  with(inttrans):
```

```
> varphi := omega[y]*R^2 + p*phi*R^2 / (2*(1-p*phi*R^2*lambda)) +
  p*R^2/(1-p*R^2*lambda) + omega[x]*R^2 / (1-p*phi*R^2*lambda)^2;
```

$$\varphi := \omega_y R^2 + \frac{p \phi R^2}{-2 p \phi R^2 \lambda + 2} + \frac{p R^2}{-p R^2 \lambda + 1} + \frac{\omega_x R^2}{\left(-p \phi R^2 \lambda + 1\right)^2} \tag{1}$$

```
> # denominator Q
  Q := (1-p*phi*R^2*lambda)^2*(1-p*R^2*lambda);
  Q1 := simplify(-coeff(Q, lambda, 1));
  Q2 := simplify(coeff(Q, lambda, 2));
  Q3 := simplify(-coeff(Q, lambda, 3));
  # must be 0
  simplify(Q - (1 - Q1*lambda + Q2*lambda^2 - Q3*lambda^3));
```

$$Q := \left(-p \phi R^2 \lambda + 1\right)^2 \left(-p R^2 \lambda + 1\right)$$

$$Q1 := R^2 p \left(2 \phi + 1\right)$$

$$Q2 := R^4 p^2 \phi \left(\phi + 2\right)$$

$$Q3 := R^6 p^3 \phi^2$$

$$0 \tag{2}$$

```
> P := collect(simplify(varphi*Q), lambda):
  P0 := simplify(coeff(P, lambda, 0));
  P1 := simplify(-coeff(P, lambda, 1));
  P2 := simplify(coeff(P, lambda, 2));
  P3 := simplify(-coeff(P, lambda, 3));
  # must be 0
  simplify(P - (P0 - P1*lambda + P2*lambda^2 - P3*lambda^3));
```

$$P0 := \frac{R^2 \left(\left(\phi + 2\right) p + 2 \omega_x + 2 \omega_y\right)}{2}$$

$$P1 := \frac{p R^4 \left(p \phi^2 + \left(5 p + 4 \omega_y\right) \phi + 2 \omega_x + 2 \omega_y\right)}{2}$$

$$P2 := \frac{3 p^2 R^6 \phi \left(\left(p + \frac{2 \omega_y}{3}\right) \phi + \frac{4 \omega_y}{3}\right)}{2}$$

$$P3 := R^8 \phi^2 \omega_y p^3$$

$$0 \tag{3}$$

```
> Qsharp := subs(phi=1, Q1*lambda+Q2*lambda^2+Q3*lambda^3):
  Psharp := subs(phi=1, P0+P1*lambda+P2*lambda^2+P3*lambda^3):
  psi := (Qsharp * subs(phi=1, varphi) + Psharp) / subs(phi=1, Q):
  convert(psi, parfrac, lambda);
```

(4)

$$-2\,\omega_y R^2+\frac{R^2\left(3p-2\omega_x-24\omega_y\right)}{\left(R^2p\lambda-1\right)^2}-\frac{12R^2\omega_y}{R^2p\lambda-1}+\frac{R^2\left(12p-8\omega_x-15\omega_y\right)}{\left(R^2p\lambda-1\right)^3}+\frac{3R^2\left(7p-8\omega_x\right)}{2\left(R^2p\lambda-1\right)^4}-\frac{7\omega_xR^2}{\left(R^2p\lambda-1\right)^5} \tag{4}$$

```
> psi_aux := convert(eval(psi, lambda=1/lambda) / lambda^2,
  parfrac, lambda);
```

$$psi\_aux := -\frac{2\omega_yR^2}{\lambda^2}+\frac{7R^8p^3\omega_x}{\left(-pR^2+\lambda\right)^5}+\frac{R^2\left(2\omega_x+3p-18\omega_y\right)}{2\left(-pR^2+\lambda\right)^2}-\frac{12\omega_y}{p\lambda}+\frac{3R^6p^2\left(6\omega_x+7p\right)}{2\left(-pR^2+\lambda\right)^4}+\frac{12\omega_y}{p\left(-pR^2+\lambda\right)}+\frac{R^4p\left(5\omega_x+9p+15\omega_y\right)}{\left(-pR^2+\lambda\right)^3} \tag{5}$$

```
> local Psi:
  Psi := 1/xi * invlaplace(psi_aux, lambda, xi):
  Psi := collect(expand(Psi), exp(p*R^2*xi));
```

$$\Psi := \left(\frac{7\xi^3p^3R^8\omega_x}{24}+\frac{7\xi^2p^3R^6}{4}+\frac{3\xi^2p^2R^6\omega_x}{2}+\frac{9\xi p^2R^4}{2}+\frac{5\xi pR^4\omega_x}{2}+\frac{15\xi pR^4\omega_y}{2}+\frac{3pR^2}{2}+R^2\omega_x-9\omega_yR^2+\frac{12\omega_y}{\xi p}\right)e^{pR^2\xi}-2\omega_yR^2-\frac{12\omega_y}{\xi p} \tag{6}$$

```
> # remove the negative term -2*omega[y]*R^2
  Psi := Psi + 2*omega[y]*R^2;
```

$$\Psi := \left(\frac{7\xi^3p^3R^8\omega_x}{24}+\frac{7\xi^2p^3R^6}{4}+\frac{3\xi^2p^2R^6\omega_x}{2}+\frac{9\xi p^2R^4}{2}+\frac{5\xi pR^4\omega_x}{2}+\frac{15\xi pR^4\omega_y}{2}+\frac{3pR^2}{2}+R^2\omega_x-9\omega_yR^2+\frac{12\omega_y}{\xi p}\right)e^{pR^2\xi}-\frac{12\omega_y}{\xi p} \tag{7}$$

```
> # use (exp(t)-1)/t << exp(t) with t=p*R^2*xi to bound Psi
  t := p*R^2*xi:
  Psi := Psi + 12*omega[y]*R^2*(exp(t) - (exp(t)-1)/t):
  Psi := collect(expand(Psi), exp(p*R^2*xi));
```

$$\Psi := \left(\frac{7}{24}\xi^3p^3R^8\omega_x+\frac{7}{4}\xi^2p^3R^6+\frac{3}{2}\xi^2p^2R^6\omega_x+\frac{9}{2}\xi p^2R^4+\frac{5}{2}\xi pR^4\omega_x+\frac{15}{2}\xi pR^4\omega_y+\frac{3}{2}pR^2+R^2\omega_x+3\omega_yR^2\right)e^{pR^2\xi} \tag{8}$$

```
> W := Psi / exp(p*R^2*xi):
  W := collect(W, xi);
```

$$W := \frac{7\xi^3p^3R^8\omega_x}{24}+\left(\frac{7}{4}p^3R^6+\frac{3}{2}p^2R^6\omega_x\right)\xi^2+\left(\frac{9}{2}p^2R^4+\frac{5}{2}pR^4\omega_x\right. \tag{9}$$

$$+\frac{15}{2}\,p\,R^4\,\omega_y\Big)\,\xi+\frac{3\,p\,R^2}{2}+R^2\,\omega_x+3\,\omega_y\,R^2$$

```
> C := int(W, xi=0..1):
  C := collect(C, R);
```

$$C := \frac{7\,R^8\,p^3\,\omega_x}{96}+\left(\frac{7}{12}\,p^3+\frac{1}{2}\,\omega_x\,p^2\right)R^6+\left(\frac{9}{4}\,p^2+\frac{5}{4}\,\omega_x\,p+\frac{15}{4}\,\omega_y\,p\right)R^4 \qquad \textbf{(10)}$$
$$+\left(\frac{3\,p}{2}+\omega_x+3\,\omega_y\right)R^2$$