Rounding Error Analysis of an Orbital Collision Probability Evaluation Algorithm

12^{e} Biennale Française des Mathématiques Appliquées et Industrielles
Carcans Maubuisson
June 2–6, 2025

Denis Arzelier[*], **Florent Bréhard**[†], **Mioara Joldes**[*], **Marc Mezzarobba**[○]

[*] LAAS–CNRS, Toulouse, France
[†] CNRS, Univ. Lille, CRIStAL, France
[○] CNRS, LIX, Palaiseau, France

arzelier@laas.fr; florent.brehard@univ-lille.fr; mmjoldes@laas.fr; marc@mezzarobba.net

Recurrences and Validated Numerics

- Recurrences are ubiquitous in computer-aided mathematics: spectral methods for functional equations, dynamical systems, combinatorics, asymptotics, ...

- They were extensively studied in numerical analysis: Olver, Miller, Wimp, Clenshaw, Henrici, Barrio, ...

- Challenges for validated numerics: numerical instability, wrapping effect
  $\Rightarrow$ Tight bounds are difficult to obtain

- M. Mezzarobba. Rounding Error Analysis of Linear Recurrences Using Generating Series. *ETNA - Electronic Transactions on Numerical Analysis*, 2023
  $\Rightarrow$ Connect the total error to the analytic properties of the initial problem
  $\Rightarrow$ Deduce a priori and realistic error bounds

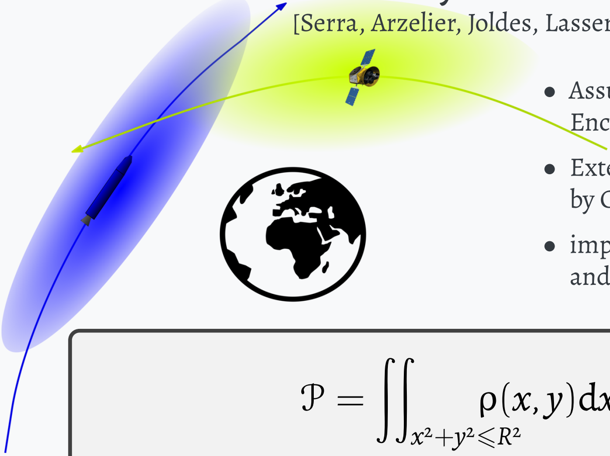
1. Orbital Collision Probability Evaluation Algorithm

2. Roundoff Error Analysis using Majorizing Series

3. Numerical Examples

Orbital Collision Probability

[Serra, Arzelier, Joldes, Lasserre, Rondepierre, Salvy – 2015]



- Assumptions for Short Term Encounter
- Extensively tested and approved by CNES
- implemented for both ground and onboard usage

$$\mathcal{P} = \iint_{x^2+y^2 \leqslant R^2} \rho(x,y)\,dx\,dy$$

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

Algorithm: Computation of the Probability of Collision

Require: FP Parameters R, σ_x, σ_y, x_m, y_m; number of terms N.

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

1: Evaluate p, ϕ, ω_x, ω_y, Q_1, Q_2, Q_3, P_0, P_1, P_2, P_3;

2: $c_0 = \ldots; c_1 = \ldots; c_2 = \ldots; c_3 = \ldots;$

3: $s = c_0 + c_1 + c_2 + c_3;$

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

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

6: $s = s + c_n;$

7: **end for**

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

[Serra, Arzelier, Joldes, Lasserre, Rondepierre, Salvy – 2015]

Algorithm: Computation of the Probability of Collision

Require: FP Parameters $R, \sigma_x, \sigma_y, x_m, y_m$; number of terms N.

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

Evaluate $p, \phi, \omega_x, \omega_y, Q_1, Q_2, Q_3, P_0, P_1, P_2, P_3$;

$c_0 = \ldots; c_1 = \ldots; c_2 = \ldots; c_3 = \ldots;$

$s = c_0 + c_1 + c_2 + c_3;$

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

initial terms

Algorithm: Computation of the Probability of Collision

Require: FP Parameters R, σ_x, σ_y, x_m, y_m; number of terms N.

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

Evaluate p, ϕ, ω_x, ω_y, Q_1, Q_2, Q_3, P_0, P_1, P_2, P_3;

$c_0 = \dots; c_1 = \dots; c_2 = \dots; c_3 = \dots;$

$s = c_0 + c_1 + c_2 + c_3;$

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

initial terms

unroll the recurrence

Algorithm: Computation of the Probability of Collision

Require: FP Parameters R, σ_x, σ_y, x_m, y_m; number of terms N.

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

Evaluate p, ϕ, ω_x, ω_y, Q_1, Q_2, Q_3, P_0, P_1, P_2, P_3;

$c_0 = \ldots; c_1 = \ldots; c_2 = \ldots; c_3 = \ldots;$

$s = c_0 + c_1 + c_2 + c_3;$

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

initial terms | unroll the recurrence | sum the terms

Algorithm: Computation of the Probability of Collision

Require: FP Parameters R, σ_x, σ_y, x_m, y_m; number of terms N.

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

1: Evaluate p, ϕ, ω_x, ω_y, Q_1, Q_2, Q_3, P_0, P_1, P_2, P_3;

2: $c_0 = \dots; c_1 = \dots; c_2 = \dots; c_3 = \dots;$

3: $s = c_0 + c_1 + c_2 + c_3;$

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

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

6: $s = s + c_n;$

7: **end for**

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

preconditioning to ensure $c_n \geqslant 0$

initial terms

unroll the recurrence

sum the terms

Algorithm: Computation of the Probability of Collision

Require: FP Parameters R, σ_x, σ_y, x_m, y_m; number of terms N.

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

1: Evaluate p, ϕ, ω_x, ω_y, Q_1, Q_2, Q_3, P_0, P_1, P_2, P_3;

2: $c_0 = \ldots; c_1 = \ldots; c_2 = \ldots; c_3 = \ldots;$

3: $s = c_0 + c_1 + c_2 + c_3;$

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

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

6: $s = s + c_n;$

7: **end for**

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

explicit truncation error bounds

preconditioning to ensure $c_n \geqslant 0$

initial terms

unroll the recurrence

sum the terms

How this algorithm works in a nutshell

$$\mathcal{P} = \exp(-pR^2\xi) f(\xi) \text{ at } \xi = 1$$

$$f(\xi) = c_0 + c_1\xi + c_2\xi^2 + \cdots + c_n\xi^n + \ldots$$

[$\Rightarrow$ ensures $c_n \geqslant 0$]

differential equation on $f(\xi)$

D-finite functions

Recurrence on the coefficients c_n:

$$c_n = \square c_{n-1} + \square c_{n-2} + \square c_{n-3} + \square c_{n-4}$$

$$\mathcal{P}_{0:N} \stackrel{\text{def}}{=} \exp(-pR^2) \sum_{n=1}^{N-1} c_n \approx \mathcal{P}$$

How this algorithm works in a nutshell

Laplace transform

$$\mathcal{P} = \exp(-pR^2\xi) f(\xi) \text{ at } \xi = 1$$

$$f(\xi) = c_0 + c_1\xi + c_2\xi^2 + \cdots + c_n\xi^n + \ldots$$

$$\widehat{f}(\lambda) = \left\{\text{explicit function of } \lambda\right\}$$

$$= c_0 + c_1\lambda + 2c_2\lambda^2 + \cdots + n!c_n\lambda^n + \ldots$$

[$\Rightarrow$ ensures $c_n \geqslant 0$]

differential equation on $f(\xi)$

D-finite functions

Recurrence on the coefficients c_n:

$$c_n = \square c_{n-1} + \square c_{n-2} + \square c_{n-3} + \square c_{n-4}$$

$$\mathcal{P}_{0:N} \stackrel{\text{def}}{=} \exp(-pR^2) \sum_{n=1}^{N-1} c_n \approx \mathcal{P}$$

How this algorithm works in a nutshell

Laplace transform

$$\mathcal{P} = \exp(-pR^2\xi) f(\xi) \text{ at } \xi = 1$$

$$f(\xi) = c_0 + c_1\xi + c_2\xi^2 + \cdots + c_n\xi^n + \dots$$

$$\widehat{f}(\lambda) = \left\{\text{explicit function of } \lambda\right\}$$

$$= c_0 + c_1\lambda + 2c_2\lambda^2 + \cdots + n!c_n\lambda^n + \dots$$

[$\Rightarrow$ ensures $c_n \geqslant 0$]

$$\widehat{f}'(\lambda) = \varphi(\lambda)\widehat{f}(\lambda)$$

$$\varphi(\lambda) = \frac{P_0 - P_1\lambda + P_2\lambda^2 - P_3\lambda^3}{1 - Q_1\lambda + Q_2\lambda^2 - Q_3\lambda^3}$$

Recurrence on the coefficients c_n:

$$c_n = \square c_{n-1} + \square c_{n-2} + \square c_{n-3} + \square c_{n-4}$$

$$\mathcal{P}_{0:N} \stackrel{\text{def}}{=} \exp(-pR^2) \sum_{n=1}^{N-1} c_n \approx \mathcal{P}$$

How this algorithm works in a nutshell

Laplace transform

$$\mathcal{P} = \exp(-pR^2\xi)\, f(\xi) \text{ at } \xi = 1$$

$$f(\xi) = c_0 + c_1\xi + c_2\xi^2 + \cdots + c_n\xi^n + \ldots$$

$$\widehat{f}(\lambda) = \left\{\text{explicit function of } \lambda\right\}$$

$$= c_0 + c_1\lambda + 2c_2\lambda^2 + \cdots + n!c_n\lambda^n + \ldots$$

[$\Rightarrow$ ensures $c_n \geqslant 0$]

$$\widehat{f}'(\lambda) = \varphi(\lambda)\widehat{f}(\lambda)$$

$$\varphi(\lambda) = \frac{P_0 - P_1\lambda + P_2\lambda^2 - P_3\lambda^3}{1 - Q_1\lambda + Q_2\lambda^2 - Q_3\lambda^3}$$

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

$$\mathcal{P}_{0:N} \stackrel{\text{def}}{=} \exp(-pR^2) \sum_{n=1}^{N-1} c_n \approx \mathcal{P}$$

How this algorithm works in a nutshell

Roundoff error bound for $|\widetilde{\mathcal{P}}_{0:N} - \mathcal{P}_{0:N}|$?

- standard binary64 (no interval arithmetic, no multiprecision)
  $\rightsquigarrow$ efficiency/architecture constraints
- N can be large! $(N = 100,\ 1000,\ 10000, \dots)$

$\Rightarrow$ A priori bounds using majorizing series techniques for round-off analysis of linear recurrences

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

$$\mathcal{P}_{0:N} \stackrel{\text{def}}{=} \exp(-pR^2) \sum_{n=1}^{N-1} c_n \approx \mathcal{P}$$

1. Orbital Collision Probability Evaluation Algorithm

2. Roundoff Error Analysis using Majorizing Series

3. Numerical Examples

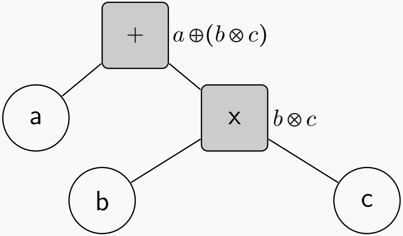
Error model for floating-point arithmetic

- Precision p binary floating-point arithmetic (e.g., $p = 53$ for `binary64`)
- unbounded exponent range ($\Rightarrow$ no underflow, no overflow)
- round-to-nearest mode for arithmetic operations $\star \in \{+, -, \times, \div, \sqrt{}\}$

$$\widetilde{a \star b} = (a \star b)(1 + e) \qquad |e| \leqslant u \stackrel{\text{def}}{=} 2^{-p}$$

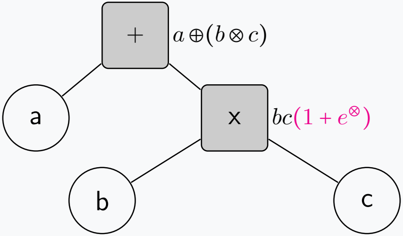
Error model for floating-point arithmetic

- Precision p binary floating-point arithmetic (e.g., $p = 53$ for `binary64`)
- unbounded exponent range ($\Rightarrow$ no underflow, no overflow)
- round-to-nearest mode for arithmetic operations $\star \in \{+, -, \times, \div, \sqrt{}\}$



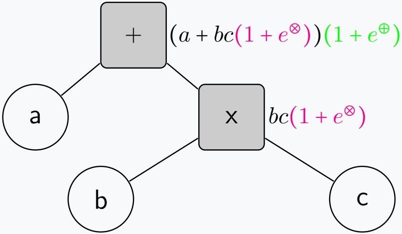
Error model for floating-point arithmetic

- Precision p binary floating-point arithmetic (e.g., $p = 53$ for `binary64`)
- unbounded exponent range ($\Rightarrow$ no underflow, no overflow)
- round-to-nearest mode for arithmetic operations $\star \in \{+, -, \times, \div, \sqrt{}\}$



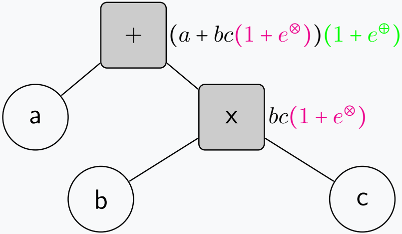
Error model for floating-point arithmetic

- Precision p binary floating-point arithmetic (e.g., $p = 53$ for `binary64`)
- unbounded exponent range ($\Rightarrow$ no underflow, no overflow)
- round-to-nearest mode for arithmetic operations $\star \in \{+, -, \times, \div, \sqrt{}\}$



Error model for floating-point arithmetic

- Precision p binary floating-point arithmetic (e.g., $p = 53$ for `binary64`)
- unbounded exponent range ($\Rightarrow$ no underflow, no overflow)
- round-to-nearest mode for arithmetic operations $\star \in \{+, -, \times, \div, \sqrt{}\}$



Example

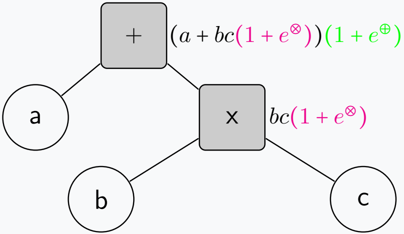
$$(a \oplus b \otimes c) - (a + bc) =$$

$$ae^{\oplus} + bc[(1 + e^{\oplus})(1 + e^{\otimes}) - 1]$$

$$|e^{\oplus}|, |e^{\otimes}| \leqslant u \stackrel{\text{def}}{=} 2^{-p}$$

Error model for floating-point arithmetic

- Precision p binary floating-point arithmetic (e.g., $p = 53$ for `binary64`)
- unbounded exponent range ($\Rightarrow$ no underflow, no overflow)
- round-to-nearest mode for arithmetic operations $\star \in \{+, -, \times, \div, \sqrt{}\}$



Example

$$(a \oplus b \otimes c) - (a + bc) =$$
$$ae^{\oplus} + bc[(1 + e^{\oplus})(1 + e^{\otimes}) - 1]$$
$$|e^{\oplus}|, |e^{\otimes}| \leqslant u \overset{\text{def}}{=} 2^{-p}$$

Notation ([Higham, *Accuracy and Stability of Numerical Algorithms* – 2002])

$$(1 + e_1) \ldots (1 + e_n) \overset{\text{def}}{=} (1 + \theta_n) \qquad |\theta_n| \leqslant \gamma_n \overset{\text{def}}{=} \frac{nu}{1 - nu}$$

Roundoff errors in the algorithm

Require: FP Parameters R, σ_x, σ_y, x_m, y_m; number of terms N.

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

Evaluate p, ϕ, ω_x, ω_y, Q_1, Q_2, Q_3, P_0, P_1, P_2, P_3;

$c_0 = \dots; c_1 = \dots; c_2 = \dots; c_3 = \dots;$

$s = c_0 + c_1 + c_2 + c_3;$

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

Roundoff errors in the algorithm

Require: FP Parameters R, σ_x, σ_y, x_m, y_m; number of terms N.

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

1: Evaluate p, ϕ, ω_x, ω_y, Q_1, Q_2, Q_3, P_0, P_1, P_2, P_3;

2: $c_0 = \ldots; c_1 = \ldots; c_2 = \ldots; c_3 = \ldots;$

3: $s = c_0 + c_1 + c_2 + c_3;$

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

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

6: $s = s + c_n;$

7: **end for**

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

Initial errors

Roundoff errors in the algorithm

Require: FP Parameters R, σ_x, σ_y, x_m, y_m; number of terms N.

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

Evaluate p, ϕ, ω_x, ω_y, Q_1, Q_2, Q_3, P_0, P_1, P_2, P_3;

$c_0 = \dots; c_1 = \dots; c_2 = \dots; c_3 = \dots;$

$s = c_0 + c_1 + c_2 + c_3;$

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

Initial errors   Local errors

Roundoff errors in the algorithm

Require: FP Parameters R, σ_x, σ_y, x_m, y_m; number of terms N.

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

Evaluate p, ϕ, ω_x, ω_y, Q_1, Q_2, Q_3, P_0, P_1, P_2, P_3;

$c_0 = \ldots; c_1 = \ldots; c_2 = \ldots; c_3 = \ldots;$

$s = c_0 + c_1 + c_2 + c_3;$

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

Initial errors　　Local errors $\rightarrow$ Global errors

Roundoff errors in the algorithm

Require: FP Parameters R, σ_x, σ_y, x_m, y_m; number of terms N.

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

Evaluate p, ϕ, ω_x, ω_y, Q_1, Q_2, Q_3, P_0, P_1, P_2, P_3;

$c_0 = \ldots; c_1 = \ldots; c_2 = \ldots; c_3 = \ldots;$

$s = c_0 + c_1 + c_2 + c_3;$

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

Initial errors Local errors $\rightarrow$ Global errors Summation errors

Roundoff errors in the algorithm

Require: FP Parameters R, σ_x, σ_y, x_m, y_m; number of terms N.

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

Evaluate $p, \phi, \omega_x, \omega_y, Q_1, Q_2, Q_3, P_0, P_1, P_2, P_3$;

$c_0 = \ldots; c_1 = \ldots; c_2 = \ldots; c_3 = \ldots;$

$s = c_0 + c_1 + c_2 + c_3;$

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

Initial errors + Local errors $\rightarrow$ Global errors + Summation errors

Initial and local errors using standard roundoff analysis

- Initial errors (loop-independent parameters):

$$\omega_x = \frac{x_m^2}{4\sigma_x^4} \quad \rightsquigarrow \quad \widetilde{\omega}_x = \frac{1}{4} \cdot (x_m \otimes x_m) \oslash ((\sigma_x \otimes \sigma_x) \otimes (\sigma_x \otimes \sigma_x))$$

$$\Rightarrow \quad |\widetilde{\omega}_x - \omega_x| \leqslant \gamma_5 \omega_x$$

$\rightsquigarrow$ Similar bounds for the other parameters $\rightsquigarrow$ P_i, Q_i

Initial and local errors using standard roundoff analysis

- Initial errors (loop-independent parameters):

$$\omega_x = \frac{x_m^2}{4\sigma_x^4} \quad \rightsquigarrow \quad \widetilde{\omega}_x = \frac{1}{4} \cdot (x_m \otimes x_m) \oslash ((\sigma_x \otimes \sigma_x) \otimes (\sigma_x \otimes \sigma_x))$$

$$\Rightarrow \quad |\widetilde{\omega}_x - \omega_x| \leqslant \gamma_5 \omega_x$$

$\rightsquigarrow$ Similar bounds for the other parameters $\rightsquigarrow$ P_i, Q_i

- Local errors ε_n (at each iteration):

$$\widetilde{c}_n = \frac{Q_1(n-1)+P_0}{(n+1)n}\widetilde{c}_{n-1} - \frac{Q_2(n-2)+P_1}{(n+1)n^2}\widetilde{c}_{n-2} + \frac{Q_3(n-3)+P_2}{(n+1)n^2(n-1)}\widetilde{c}_{n-3} - \frac{P_3}{(n+1)n^2(n-1)(n-2)}\widetilde{c}_{n-4} + \varepsilon_n$$

$$\boxed{|\varepsilon_n| \leqslant \gamma_{40}\left(\frac{Q_1(n-1)+P_0}{(n+1)n}|\tilde{c}_{n-1}| + \frac{Q_2(n-2)+P_1}{(n+1)n^2}|\tilde{c}_{n-2}| + \frac{Q_3(n-3)+P_2}{(n+1)n^2(n-1)}|\tilde{c}_{n-3}| + \frac{P_3}{(n+1)n^2(n-1)(n-2)}|\tilde{c}_{n-4}|\right)}$$

Global error: Outline

following [Mezzarobba 2020]

$$f(\xi) = \sum_{n=0}^{+\infty} c_n \xi^n$$

$$\widehat{f}(\lambda) = \sum_{n=0}^{+\infty} n! c_n \lambda^n$$

$$\boxed{\widehat{f}'(\lambda) - \varphi(\lambda)\widehat{f}(\lambda) = 0}$$

$$\boxed{\begin{aligned} c_n - \square\, c_{n-1} + \square\, c_{n-2} \\ - \square\, c_{n-3} + \square\, c_{n-4} = 0 \end{aligned}}$$

$$f(1) = s = \sum_{n=0}^{+\infty} c_n$$

$$\delta_n \overset{\text{def}}{=} \widetilde{c}_n - c_n$$

$$\delta(1) = \widetilde{s} - s = \sum_{n=0}^{+\infty} \widetilde{c}_n - \sum_{n=0}^{+\infty} c_n$$

[bound ε_n]

$$\boxed{\begin{aligned} \delta_n - \square\, \delta_{n-1} + \square\, \delta_{n-2} \\ - \square\, \delta_{n-3} + \square\, \delta_{n-4} = \varepsilon_n \end{aligned}}$$

$[\sum_{n=0}^{+\infty} (\cdot)\, n! \lambda^n]$

$$\boxed{\widehat{\delta}'(\lambda) - \varphi(\lambda)\widehat{\delta}(\lambda) = \widehat{\varepsilon}(\lambda)}$$

[solve for $\widehat{\delta}(\lambda)$]

$[\widehat{\delta}(\lambda) \rightsquigarrow \delta(\xi)]$

$$\delta(\xi) = \widetilde{f}(\xi) - f(\xi) = \sum_{n=0}^{+\infty} \underbrace{(\widetilde{c}_n - c_n)}_{\delta_n} \xi^n$$

Global error: Differential inequality

Recall that for $\widetilde{c}_n = c_n + \delta_n$,

$$\delta_n - \square\, \delta_{n-1} + \square\, \delta_{n-2} - \square\, \delta_{n-3} + \square\, \delta_{n-4} = \varepsilon_n$$

$$|\varepsilon_n| \leqslant \gamma_{40}\Big(\square\, |\tilde{c}_{n-1}| + \square\, |\tilde{c}_{n-2}| + \square\, |\tilde{c}_{n-3}| + \square\, |\tilde{c}_{n-4}|\Big) \qquad \gamma_{40} \approx 40u$$

Global error: Differential inequality

Recall that for $\tilde{c}_n = c_n + \delta_n$,

$$\delta_n - \square\,\delta_{n-1} + \square\,\delta_{n-2} - \square\,\delta_{n-3} + \square\,\delta_{n-4} = \varepsilon_n$$

$$|\varepsilon_n| \leqslant \gamma_{40}\Big(\square\,|\tilde{c}_{n-1}| + \square\,|\tilde{c}_{n-2}| + \square\,|\tilde{c}_{n-3}| + \square\,|\tilde{c}_{n-4}|\Big) \qquad \gamma_{40} \approx 40u$$

Hence we obtain a differential inequality:

$$\hat{\delta}'(\lambda) - \varphi(\lambda)\hat{\delta}(\lambda) \ll \gamma_{40}\left(\psi(\lambda)\hat{f}(\lambda) + \underbrace{(\dots)\hat{\delta}'(\lambda) + (\dots)\hat{\delta}(\lambda)}_{\text{small}}\right)$$

$$[\psi(\lambda) = \{\text{an explicit rational function in } \lambda\}]$$

$$a(\lambda) \ll b(\lambda) \quad \text{means} \quad |a_n| \leqslant |b_n| \quad \text{for all} \quad n \geqslant 0$$

Global error: Linearized bound

- Differential inequality:

$$\boxed{\widehat{f}'(\lambda) - \varphi(\lambda)\widehat{f}(\lambda) = 0}$$

$$\boxed{\widehat{\delta}'(\lambda) - \varphi(\lambda)\widehat{\delta}(\lambda) \ll 40u\,\psi(\lambda)\widehat{f}(\lambda) + o(u)}$$

[$\psi(\lambda)$ an explicit rational function]

- Solve to obtain an upper bound (neglecting $o(u)$ terms):

$$\widehat{\delta}(\lambda) \ll \Big(\delta_0 + 40u\,\Psi(\lambda)\Big)\,\widehat{f}(\lambda) \qquad \Psi(\lambda) \stackrel{\text{def}}{=} \int_0^{\lambda} \psi(\sigma)d\sigma$$

Global error: Linearized bound

- Differential inequality:

$$\boxed{\widehat{f}'(\lambda) - \varphi(\lambda)\widehat{f}(\lambda) = 0}$$

$$\boxed{\widehat{\delta}'(\lambda) - \varphi(\lambda)\widehat{\delta}(\lambda) \ll 40u\,\psi(\lambda)\widehat{f}(\lambda) + o(u)}$$

[$\psi(\lambda)$ an explicit rational function]

- Solve to obtain an upper bound (neglecting $o(u)$ terms):

$$\widehat{\delta}(\lambda) \ll \left(\delta_0 + 40u\,\Psi(\lambda)\right)\widehat{f}(\lambda) \qquad \Psi(\lambda) \overset{\text{def}}{=} \int_0^\lambda \psi(\sigma)d\sigma$$

relative error on $\widetilde{c}_0$

propagation of local errors

Global error: Linearized bound

- Differential inequality:

$$\widehat{f}'(\lambda) - \varphi(\lambda)\widehat{f}(\lambda) = 0$$

$$\widehat{\delta}'(\lambda) - \varphi(\lambda)\widehat{\delta}(\lambda) \ll 40u\,\psi(\lambda)\widehat{f}(\lambda) + o(u)$$

[$\psi(\lambda)$ an explicit rational function]

- Solve to obtain an upper bound (neglecting $o(u)$ terms):

$$\widehat{\delta}(\lambda) \ll \left(\delta_0 + 40u\,\Psi(\lambda)\right)\widehat{f}(\lambda) \qquad \Psi(\lambda) \stackrel{\text{def}}{=} \int_0^\lambda \psi(\sigma)d\sigma$$

relative error on $\widetilde{c}_0$

propagation of local errors

- Inverse Laplace transform using Maple

$$|\delta(1)| = \sum_{n=0}^{+\infty} |\widetilde{c}_n - c_n| \leqslant \left(\delta_0 + 40u\,C\right) f(1) + o(u)$$

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

Total roundoff error

Linearized bound

$$\frac{|\widetilde{\mathcal{P}}_{0:N} - \mathcal{P}_{0:N}|}{\mathcal{P}} \leqslant \left(\frac{2x_m^2}{\sigma_x^2} + \frac{2y_m^2}{\sigma_y^2} + 40C + N + 2pR^2 + 8 \right) u + o(u)$$

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

[initial error on $\widetilde{c}_0$ global error on $\widetilde{c}_n$ summation error final rescaling]

- The required number of terms N depends on the parameters (→ truncation error bound), and $N \leqslant C$ in practice
- We also derived a rigorous (i.e., not linearized) total error bound
- We proved that the 64-bit exponent emulation is sufficient to avoid overflows for realistic parameter ranges

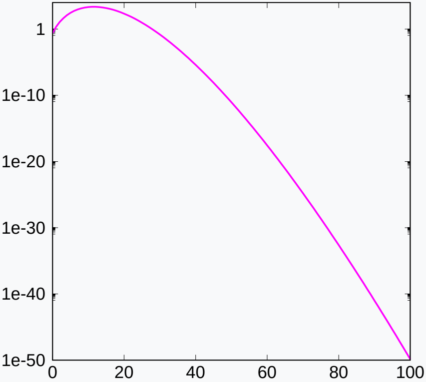
1. Orbital Collision Probability Evaluation Algorithm

2. Roundoff Error Analysis using Majorizing Series

3. Numerical Examples

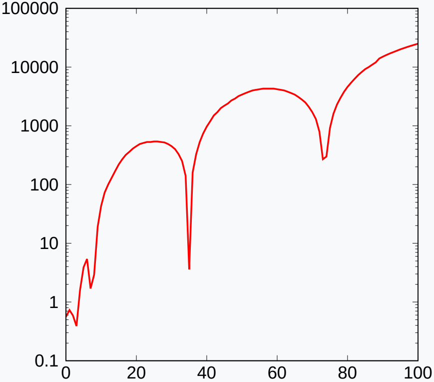
Analysis of a numerical example

$$R = 5, \quad \sigma_x = 50, \quad \sigma_y = 1, \quad x_m = 10, \quad y_m = 0 \quad \Rightarrow \quad N = 101$$



c_n

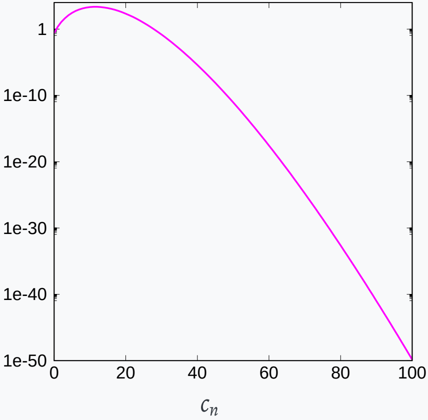
float binary64

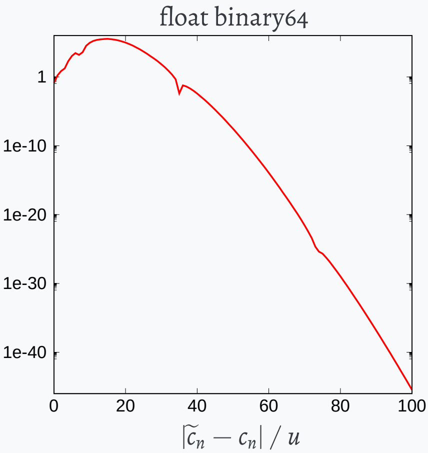


$|\widetilde{c}_n - c_n| \,/\, u c_n$

Analysis of a numerical example

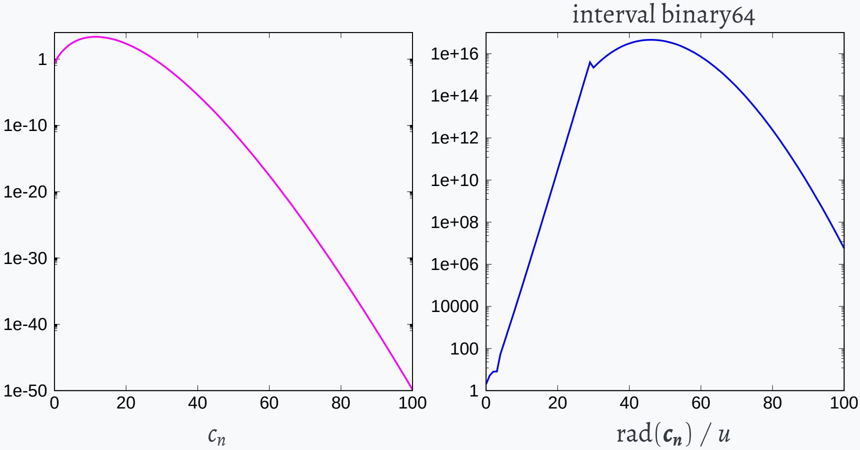
$$R = 5, \quad \sigma_x = 50, \quad \sigma_y = 1, \quad x_m = 10, \quad y_m = 0 \quad \Rightarrow \quad N = 101$$





Analysis of a numerical example

$$R = 5, \quad \sigma_x = 50, \quad \sigma_y = 1, \quad x_m = 10, \quad y_m = 0 \quad \Rightarrow \quad N = 101$$



Analysis of a numerical example

$$R = 5, \quad \sigma_x = 50, \quad \sigma_y = 1, \quad x_m = 10, \quad y_m = 0 \quad \Rightarrow \quad N = 101$$

float binary64	interval binary64	our bound
$1.26 \cdot 10^2\, u$ $= 1.40 \cdot 10^{-14}$	$3.85 \cdot 10^{13}\, u$ $= 4.27 \cdot 10^{-3}$	$6.05 \cdot 10^4\, u$ $= 6.72 \cdot 10^{-12}$

Obtained relative errors

Some more numerical examples

Case	Input parameters (m)						Relative Error			
#	σ_x	σ_y	R	x_m	y_m	N	*Exact*	MPFI	(Lin. Bound)	Full Bound
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 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
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	**3.00e-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

Take Away Message

- Successful application of Mezzarobba's FP error analysis to this POC algorithm $\Rightarrow$ a priori closed-form relative error bounds
- Provides rigorous and realistic error bounds that could not be obtained using plain interval arithmetic
- An interesting alternative to fixed-point based a posteriori validation: the rigorous computation part is "constant time" (i.e., not proportional to N)
- However, preliminary pen-and-paper work is still substantial, the class of treated recurrences is limited (linear?), and error bounds are "worst-case".