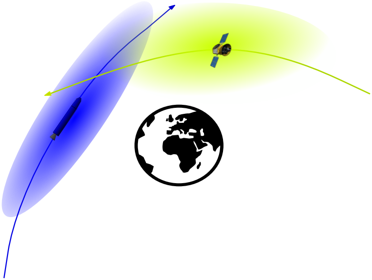
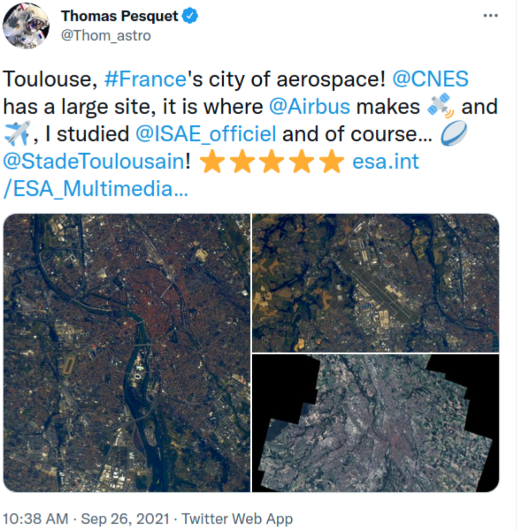
The Art of Avoiding Satellite Smash-Ups: Crunching Numbers but dodging Floating-Point Disasters with D-Finite Functions

Mioara Joldes

LAAS-CNRS, Toulouse, France



RAIM 2024: 15èmes Rencontres de l'Arithmétique en Informatique Mathématique
Perpignan, November 5, 2024



Joint works (2015–) with several collaborators, especially

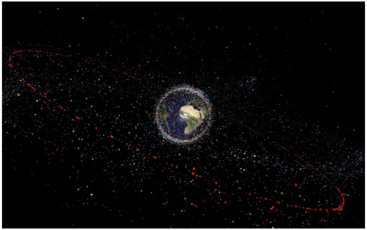
D. Arzelier, F. Bréhard, M. Mezzarobba, J.-B. Lasserre, B. Salvy, R. Serra

Joint Projects with CNES and Airbus Defence and Space, Thales

Outline

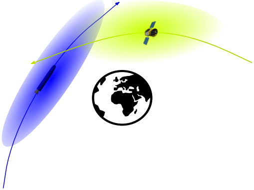
- The problem of orbital collision risk assessment and mitigation
- Evaluation of a *special function*:
  - Reliable double-precision implementation by power series
  - Roundoff error analysis
- Extensions & Conclusion

Orbital collisions



Space debris population model (source : ESA)

Cerise hit by a debris in 1996 (CNES/D. Ducros)

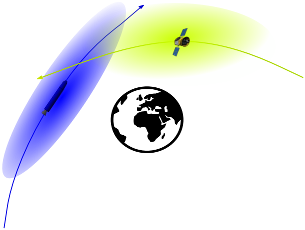


Conjunction illustration

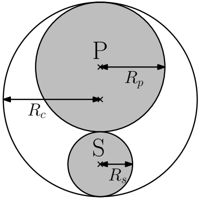
Thomas Pesquet, when a debris whizzes past the ISS:**

"Climb into an escape shuttle, wait and hope. This happened four times"

** http://www.chron.com/news/science-environment/article/Thousands-of-tiny-satellites-are-about-to-go-into-11088984.php



Primary P (operational) &
Secondary S (debris)



Combined spherical object

Classical assumptions

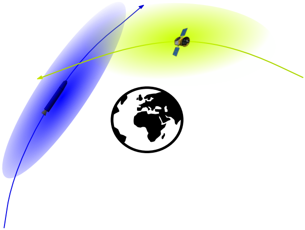
- Spherical objects
- Uncertain position & velocity
  $\rightsquigarrow$ Gaussian probability density functions
- Independent probability distribution laws

Space surveillance & tracking

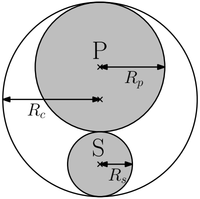
- Risk assessment
- Design of a collision avoidance strategy

  $\rightsquigarrow$ compute the probability of collision

On-orbit collision



Primary P (operational) &
Secondary S (debris)



Combined spherical object

Classical assumptions

- Spherical objects
- Uncertain position & velocity
  $\rightsquigarrow$ Gaussian probability density functions
- Independent probability distribution laws

Space surveillance & tracking

- Risk assessment
- Design of a collision avoidance strategy

  $\rightsquigarrow$ compute the probability of collision

Probability of collision

Generally: 12-dimensional, Gaussian integrand,
Complicated integration domain

Computation: Monte-Carlo trials and/or
simplified models

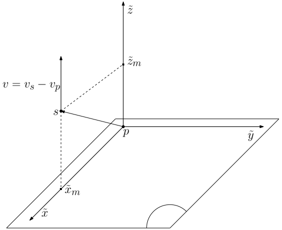
Short-term encounter model

- Framework: High relative velocity
- Assumptions:
  - Rectilinear relative motion
  - No velocity uncertainty
  - Infinite encounter time horizon

Short-term encounter model

- Framework: High relative velocity
- Assumptions:
  - Rectilinear relative motion
  - No velocity uncertainty
  - Infinite encounter time horizon

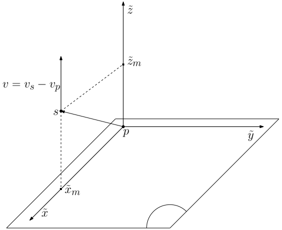
Encounter frame of reference

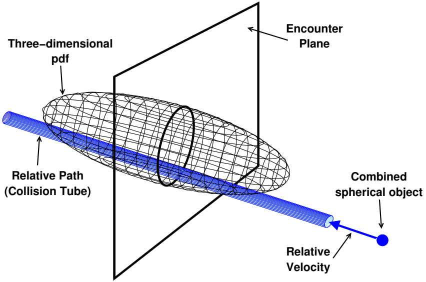


Short-term encounter model

- Framework: High relative velocity
- Assumptions:
  - Rectilinear relative motion
  - No velocity uncertainty
  - Infinite encounter time horizon

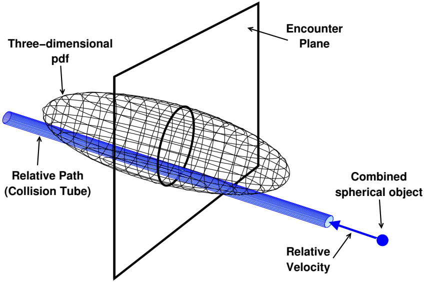
Encounter frame of reference



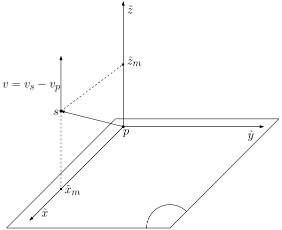


Short-term encounter model

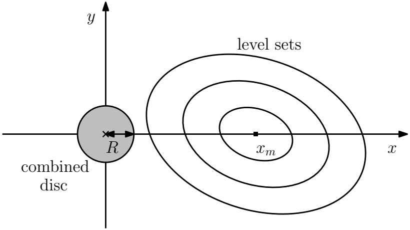
- Framework: High relative velocity
- Assumptions:
  - Rectilinear relative motion
  - No velocity uncertainty
  - Infinite encounter time horizon



Encounter frame of reference

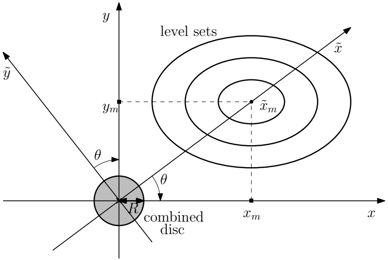


Encounter plane projection



Short-term encounter model and probability of collision

Rotation in encounter plane

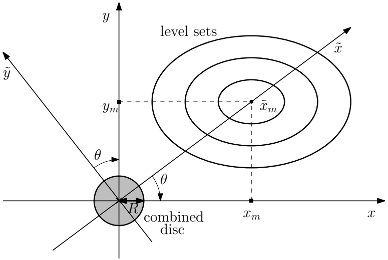


Probability of collision: bivariate normal law $\rightsquigarrow$ 2-D integral over a disk $\mathcal{B}(0, R)$.

$$\mathcal{P} = \frac{1}{2\pi\sigma_x\sigma_y} \int_{\mathcal{B}((0,0),R)} \exp\left(-\frac{(x-x_m)^2}{2\sigma_x{}^2} - \frac{(y-y_m)^2}{2\sigma_y{}^2}\right) \mathrm{d}x\mathrm{d}y$$

- R: radius of combined object
- x_m, y_m: mean relative coordinates
- σ_x, σ_y: standard deviations of relative coordinates

Rotation in encounter plane

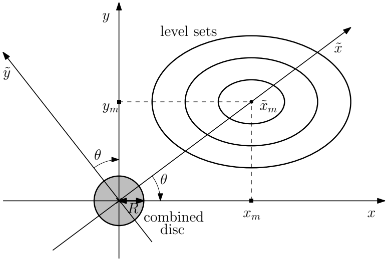


Probability of collision: bivariate normal law $\rightsquigarrow$ 2-D integral over a disk $\mathcal{B}(0, R)$.

$$\mathcal{P} = \frac{1}{2\pi\sigma_x\sigma_y} \int\limits_{\mathcal{B}((0,0),R)} \exp\left(-\frac{(x-x_m)^2}{2{\sigma_x}^2} - \frac{(y-y_m)^2}{2{\sigma_y}^2}\right) \mathrm{d}x\mathrm{d}y$$

- R: radius of combined object
- x_m, y_m: mean relative coordinates
- σ_x, σ_y: standard deviations of relative coordinates

Rotation in encounter plane



Goal: Give a "simple", "analytic" formula, suitable for double-precision evaluation and effective error bounds.

Existing Methods

Computer Algebra Systems (eg. Maple)$\rightsquigarrow$ EXACT

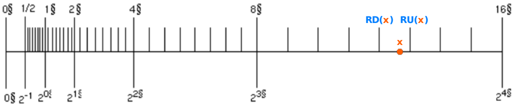
- Usually, pure symbolic methods are computationally expensive

Existing Methods

Computer Algebra Systems (eg. Maple) $\rightsquigarrow$ EXACT

- Usually, pure symbolic methods are computationally expensive

Numerics: floating-point arithmetic (eg. Matlab) $\rightsquigarrow$ FAST

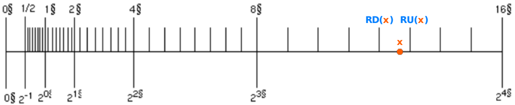


- Usually, solutions lack certification of the output accuracy

Computer Algebra Systems (eg. Maple)$\rightsquigarrow$ EXACT

- Usually, pure symbolic methods are computationally expensive

Numerics: floating-point arithmetic (eg. Matlab)$\rightsquigarrow$ FAST



- Usually, solutions lack certification of the output accuracy

e.g., methods based on numerical integration schemes: [Foster '92, Patera '01, Alfano '05] or truncated power series, with *trial and error* truncation:
Approximately 60,000 test cases were used to evaluate the numerical expression [...] The reference ("truth") probability was computed with MATHCAD 11 [...] - Alfano'05

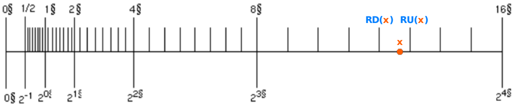
Existing Methods

Computer Algebra Systems (eg. Maple)$\leadsto$ EXACT

- Usually, pure symbolic methods are computationally expensive

e.g., closed-form power series coefficients [Chan '97] under simplifying assumption $\sigma_x = \sigma_y$

Numerics: floating-point arithmetic (eg. Matlab)$\leadsto$ FAST



- Usually, solutions lack certification of the output accuracy

e.g., methods based on numerical integration schemes: [Foster '92, Patera '01, Alfano '05] or truncated power series, with *trial and error* truncation:
Approximately 60,000 test cases were used to evaluate the numerical expression [...] The reference ("truth") probability was computed with MATHCAD 11 [...] - Alfano'05

Existing Methods

Computer Algebra Systems (eg. Maple)$\rightsquigarrow$ EXACT

- Usually, pure symbolic methods are computationally expensive

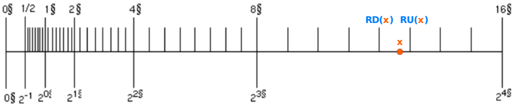
e.g., closed-form power series coefficients [Chan '97] under simplifying assumption $\sigma_x = \sigma_y$

Reliable/Validated Numerics:

Merge symbolic and numeric computations

efficient numerics + bounds on roundoff, discretization, truncation errors

Numerics: floating-point arithmetic (eg. Matlab)$\rightsquigarrow$ FAST



- Usually, solutions lack certification of the output accuracy

e.g., methods based on numerical integration schemes: [Foster '92, Patera '01, Alfano '05] or truncated power series, with *trial and error* truncation:

Approximately 60,000 test cases were used to evaluate the numerical expression [...] The reference ("truth") probability was computed with MATHCAD 11 [...] - Alfano'05

Outline

- The problem of orbital collision risk assessment and mitigation
- Evaluation of a *special function*:
  - Reliable double-precision implementation by power series
  - Roundoff error analysis
- Extensions & Conclusion

Orbital collision probability evaluation

Step 1: Symbolic representation

$$\boxed{g(\xi) = \frac{1}{2\pi\sigma_x\sigma_y} \int_{\mathcal{B}((0,0),R\sqrt{\xi})} \exp\left(-\frac{(x-x_m)^2}{2{\sigma_x}^2} - \frac{(y-y_m)^2}{2{\sigma_y}^2}\right) \mathrm{d}x\mathrm{d}y}$$

$$\boxed{g(\xi) = \frac{1}{2\pi\sigma_x\sigma_y} \int_{\mathcal{B}((0,0),R\sqrt{\xi})} \exp\left(-\frac{(x-x_m)^2}{2{\sigma_x}^2} - \frac{(y-y_m)^2}{2{\sigma_y}^2}\right) \mathrm{d}x\mathrm{d}y}$$

$$\boxed{= \sum_{i=0}^{\infty} g_i \xi^i}$$

$$\boxed{g(\xi) = \frac{1}{2\pi\sigma_x\sigma_y} \int_{\mathcal{B}((0,0),R\sqrt{\xi})} \exp\left(-\frac{(x-x_m)^2}{2{\sigma_x}^2} - \frac{(y-y_m)^2}{2{\sigma_y}^2}\right) \mathrm{d}x\mathrm{d}y} \quad \boxed{= \sum_{i=0}^{\infty} g_i \xi^i}$$

g is D-finite – solution of linear ODE with polynomial coefficients –

$$\boxed{g(\xi) = \frac{1}{2\pi\sigma_x\sigma_y} \int\limits_{\mathcal{B}((0,0),R\sqrt{\xi})} \exp\left(-\frac{(x-x_m)^2}{2{\sigma_x}^2} - \frac{(y-y_m)^2}{2{\sigma_y}^2}\right) \mathrm{d}x\mathrm{d}y}$$

$$\boxed{= \sum_{i=0}^{\infty} g_i \xi^i}$$

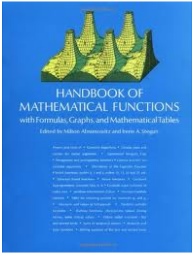
g is D-finite – solution of linear ODE with polynomial coefficients –

g_i can be efficiently symbolically described and computed
–solution of linear recurrence with polynomial coefficients–

D-finite Functions [Stanley 1980]

Differential equation + initial conditions = Data Structure

Fast algorithms for evaluation; Automatic proofs of identities



Examples

$f(x) = \exp(x) \quad \leftrightarrow \quad \{f' - f = 0,\ f(0) = 1\}.$

$\cos$, $\arccos$, Airy functions, Bessel functions, ...

About 60% of Abramowitz & Stegun

D-finite Functions [Stanley 1980]

Differential equation + initial conditions = Data Structure

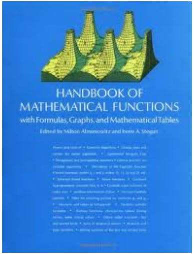
Fast algorithms for evaluation; Automatic proofs of identities

Examples

$f(x) = \exp(x) \quad \leftrightarrow \quad \{f' - f = 0,\ f(0) = 1\}.$

$\cos$, $\arccos$, Airy functions, Bessel functions, ...

About 60% of Abramowitz & Stegun

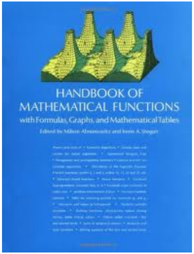


Thm. The series $\sum_{n=0}^{\infty} f_n x^n$ is D-finite iff the sequence f_n is solution of a linear recurrence with polynomial coefficients (is P-recursive).

D-finite Functions [Stanley 1980]

Differential equation + initial conditions = Data Structure

Fast algorithms for evaluation; Automatic proofs of identities



Examples

$f(x) = \exp(x) \quad \leftrightarrow \quad \{f' - f = 0,\ f(0) = 1\}.$

$\cos$, $\arccos$, Airy functions, Bessel functions, ...

About 60% of Abramowitz & Stegun

Thm. The series $\sum_{n=0}^{\infty} f_n x^n$ is D-finite iff the sequence f_n is solution of a linear recurrence with polynomial coefficients (is P-recursive).

$\rightsquigarrow$Gfun Maple package (Salvy, Zimmermann 1994)

Orbital collision probability evaluation

Step 1: Symbolic representation

$$\boxed{g(\xi) = \frac{1}{2\pi\sigma_x\sigma_y} \int_{\mathcal{B}((0,0),R\sqrt{\xi})} \exp\left(-\frac{(x-x_m)^2}{2{\sigma_x}^2} - \frac{(y-y_m)^2}{2{\sigma_y}^2}\right) \mathrm{d}x\mathrm{d}y}$$

$$\boxed{g(\xi) = \frac{1}{2\pi\sigma_x\sigma_y} \int_{\mathcal{B}((0,0),R\sqrt{\xi})} \exp\left(-\frac{(x-x_m)^2}{2\sigma_x{}^2} - \frac{(y-y_m)^2}{2\sigma_y{}^2}\right) \mathrm{d}x\mathrm{d}y}$$

Laplace Transform

- Lasserre, Zerron (2001)

$$\boxed{\mathcal{L}(g)(\lambda) = \frac{R^2 \exp\left(-\lambda\left(\frac{x_m{}^2}{2\lambda\sigma_x{}^2+R^2} + \frac{y_m{}^2}{2\lambda\sigma_y{}^2+R^2}\right)\right)}{\lambda\sqrt{(2\lambda\sigma_x{}^2+R^2)(2\lambda\sigma_y{}^2+R^2)}}}$$

Sketch of the proof - Laplace Transform [Lasserre, Zerron (2001)]

$\forall z \in \mathbb{R}^+$:

$$g(\xi) = \frac{1}{2\pi\sigma_x\sigma_y} \int_{\mathcal{B}((0,0),R\sqrt{\xi})} \exp\left(-\frac{(x-x_m)^2}{2{\sigma_x}^2} - \frac{(y-y_m)^2}{2{\sigma_y}^2}\right) \mathrm{d}x\mathrm{d}y,$$

Sketch of the proof - Laplace Transform [Lasserre, Zerron (2001)]

$\forall z \in \mathbb{R}^+$:

$$g(\xi) = \frac{1}{2\pi\sigma_x\sigma_y} \int_{\mathcal{B}((0,0),R\sqrt{\xi})} \exp\left(-\frac{(x-x_m)^2}{2{\sigma_x}^2} - \frac{(y-y_m)^2}{2{\sigma_y}^2}\right) \mathrm{d}x\mathrm{d}y,$$

$$\mathcal{L}(g)(\lambda) = \int_0^{+\infty} g(\xi)\exp(-\lambda\xi)\mathrm{d}\xi$$

$\forall z \in \mathbb{R}^+$:

$$g(\xi) = \frac{1}{2\pi\sigma_x\sigma_y} \int\limits_{\mathcal{B}((0,0),R\sqrt{\xi})} \exp\left(-\frac{(x-x_m)^2}{2{\sigma_x}^2} - \frac{(y-y_m)^2}{2{\sigma_y}^2}\right) \mathrm{d}x\mathrm{d}y,$$

$$\mathcal{L}(g)(\lambda) = \int_0^{+\infty} g(\xi)\exp(-\lambda\xi)\mathrm{d}\xi$$

$$= \frac{1}{2\pi\sigma_x\sigma_y} \int\limits_0^{+\infty}\int\limits_{\mathbb{R}^2} \mathbf{1}_{\mathcal{B}((0,0),R\sqrt{\xi})}(x,y) \exp\left(-\lambda\xi - \frac{(x-x_m)^2}{2{\sigma_x}^2} - \frac{(y-y_m)^2}{2{\sigma_y}^2}\right) \mathrm{d}x\mathrm{d}y\mathrm{d}\xi$$

$\forall z \in \mathbb{R}^+ :$

$$g(\xi) = \frac{1}{2\pi\sigma_x\sigma_y} \int\limits_{\mathcal{B}((0,0),R\sqrt{\xi})} \exp\left(-\frac{(x-x_m)^2}{2{\sigma_x}^2} - \frac{(y-y_m)^2}{2{\sigma_y}^2}\right) \mathrm{d}x\mathrm{d}y,$$

$$\mathcal{L}(g)(\lambda) = \int_0^{+\infty} g(\xi)\exp(-\lambda\xi)\mathrm{d}\xi$$

$$= \frac{1}{2\pi\sigma_x\sigma_y} \int\limits_0^{+\infty}\int\limits_{\mathbb{R}^2} \mathbf{1}_{\mathcal{B}((0,0),R\sqrt{\xi})}(x,y)\exp\left(-\lambda\xi - \frac{(x-x_m)^2}{2{\sigma_x}^2} - \frac{(y-y_m)^2}{2{\sigma_y}^2}\right) \mathrm{d}x\mathrm{d}y\mathrm{d}\xi$$

$$= \frac{1}{2\pi\sigma_x\sigma_y} \int\limits_{\mathbb{R}^2}\int\limits_0^{+\infty} \mathbf{1}_{\mathcal{B}((0,0),R\sqrt{\xi})}(x,y)\exp\left(-\lambda\xi - \frac{(x-x_m)^2}{2{\sigma_x}^2} - \frac{(y-y_m)^2}{2{\sigma_y}^2}\right) \mathrm{d}\xi\mathrm{d}x\mathrm{d}y$$

$\forall z \in \mathbb{R}^+$:

$$g(\xi) = \frac{1}{2\pi\sigma_x\sigma_y} \int\limits_{\mathcal{B}((0,0),R\sqrt{\xi})} \exp\left(-\frac{(x-x_m)^2}{2{\sigma_x}^2} - \frac{(y-y_m)^2}{2{\sigma_y}^2}\right) \mathrm{d}x\mathrm{d}y,$$

$$\begin{aligned}
\mathcal{L}(g)(\lambda) &= \int_0^{+\infty} g(\xi)\exp(-\lambda\xi)\mathrm{d}\xi \\
&= \frac{1}{2\pi\sigma_x\sigma_y} \int\limits_0^{+\infty} \int\limits_{\mathbb{R}^2} \mathbf{1}_{\mathcal{B}((0,0),R\sqrt{\xi})}(x,y) \exp\left(-\lambda\xi - \frac{(x-x_m)^2}{2{\sigma_x}^2} - \frac{(y-y_m)^2}{2{\sigma_y}^2}\right) \mathrm{d}x\mathrm{d}y\mathrm{d}\xi \\
&= \frac{1}{2\pi\sigma_x\sigma_y} \int\limits_{\mathbb{R}^2} \int\limits_0^{+\infty} \mathbf{1}_{\mathcal{B}((0,0),R\sqrt{\xi})}(x,y) \exp\left(-\lambda\xi - \frac{(x-x_m)^2}{2{\sigma_x}^2} - \frac{(y-y_m)^2}{2{\sigma_y}^2}\right) \mathrm{d}\xi\mathrm{d}x\mathrm{d}y \\
&= \frac{1}{2\pi\sigma_x\sigma_y} \int\limits_{\mathbb{R}^2} \int\limits_{(x^2+y^2)/R^2}^{+\infty} \exp\left(-\lambda\xi - \frac{(x-x_m)^2}{2{\sigma_x}^2} - \frac{(y-y_m)^2}{2{\sigma_y}^2}\right) \mathrm{d}\xi\mathrm{d}x\mathrm{d}y
\end{aligned}$$

$\forall z \in \mathbb{R}^+$:

$$g(\xi) = \frac{1}{2\pi\sigma_x\sigma_y} \int_{\mathcal{B}((0,0),R\sqrt{\xi})} \exp\left(-\frac{(x-x_m)^2}{2{\sigma_x}^2} - \frac{(y-y_m)^2}{2{\sigma_y}^2}\right) \mathrm{d}x\mathrm{d}y,$$

$$\begin{aligned}
\mathcal{L}(g)(\lambda) &= \int_0^{+\infty} g(\xi)\exp(-\lambda\xi)\mathrm{d}\xi \\
&= \frac{1}{2\pi\sigma_x\sigma_y} \int_0^{+\infty}\int_{\mathbb{R}^2} \mathbf{1}_{\mathcal{B}((0,0),R\sqrt{\xi})}(x,y)\exp\left(-\lambda\xi - \frac{(x-x_m)^2}{2{\sigma_x}^2} - \frac{(y-y_m)^2}{2{\sigma_y}^2}\right)\mathrm{d}x\mathrm{d}y\mathrm{d}\xi \\
&= \frac{1}{2\pi\sigma_x\sigma_y} \int_{\mathbb{R}^2}\int_0^{+\infty} \mathbf{1}_{\mathcal{B}((0,0),R\sqrt{\xi})}(x,y)\exp\left(-\lambda\xi - \frac{(x-x_m)^2}{2{\sigma_x}^2} - \frac{(y-y_m)^2}{2{\sigma_y}^2}\right)\mathrm{d}\xi\mathrm{d}x\mathrm{d}y \\
&= \frac{1}{2\pi\sigma_x\sigma_y} \int_{\mathbb{R}^2}\int_{(x^2+y^2)/R^2}^{+\infty} \exp\left(-\lambda\xi - \frac{(x-x_m)^2}{2{\sigma_x}^2} - \frac{(y-y_m)^2}{2{\sigma_y}^2}\right)\mathrm{d}\xi\mathrm{d}x\mathrm{d}y \\
&= \frac{1}{2\pi\sigma_x\sigma_y\lambda} \int_{\mathbb{R}^2} \exp\left(-\lambda\frac{x^2+y^2}{R^2} - \frac{1}{2}\left(\frac{(x-x_m)^2}{{\sigma_x}^2} + \frac{(y-y_m)^2}{{\sigma_y}^2}\right)\right)\mathrm{d}x\mathrm{d}y,
\end{aligned}$$

$\forall z \in \mathbb{R}^+$:

$$g(\xi) = \frac{1}{2\pi\sigma_x\sigma_y} \int_{\mathcal{B}((0,0),R\sqrt{\xi})} \exp\left(-\frac{(x-x_m)^2}{2\sigma_x{}^2} - \frac{(y-y_m)^2}{2\sigma_y{}^2}\right) \mathrm{d}x\mathrm{d}y,$$

$$\mathcal{L}(g)(\lambda) = \int_0^{+\infty} g(\xi)\exp(-\lambda\xi)\mathrm{d}\xi$$

$$= \frac{1}{2\pi\sigma_x\sigma_y} \int_0^{+\infty}\int_{\mathbb{R}^2} \mathbf{1}_{\mathcal{B}((0,0),R\sqrt{\xi})}(x,y)\exp\left(-\lambda\xi - \frac{(x-x_m)^2}{2\sigma_x{}^2} - \frac{(y-y_m)^2}{2\sigma_y{}^2}\right) \mathrm{d}x\mathrm{d}y\mathrm{d}\xi$$

$$= \frac{1}{2\pi\sigma_x\sigma_y} \int_{\mathbb{R}^2}\int_0^{+\infty} \mathbf{1}_{\mathcal{B}((0,0),R\sqrt{\xi})}(x,y)\exp\left(-\lambda\xi - \frac{(x-x_m)^2}{2\sigma_x{}^2} - \frac{(y-y_m)^2}{2\sigma_y{}^2}\right) \mathrm{d}\xi\mathrm{d}x\mathrm{d}y$$

$$= \frac{1}{2\pi\sigma_x\sigma_y} \int_{\mathbb{R}^2}\int_{(x^2+y^2)/R^2}^{+\infty} \exp\left(-\lambda\xi - \frac{(x-x_m)^2}{2\sigma_x{}^2} - \frac{(y-y_m)^2}{2\sigma_y{}^2}\right) \mathrm{d}\xi\mathrm{d}x\mathrm{d}y$$

$$= \frac{1}{2\pi\sigma_x\sigma_y\lambda} \int_{\mathbb{R}^2} \exp\left(-\lambda\frac{x^2+y^2}{R^2} - \frac{1}{2}\left(\frac{(x-x_m)^2}{\sigma_x{}^2} + \frac{(y-y_m)^2}{\sigma_y{}^2}\right)\right) \mathrm{d}x\mathrm{d}y,$$

$$\ldots = \frac{R^2\exp\left(-\lambda\left(\frac{x_m{}^2}{2\lambda\sigma_x{}^2+R^2} + \frac{y_m{}^2}{2\lambda\sigma_y{}^2+R^2}\right)\right)}{\lambda\sqrt{(2\lambda\sigma_x{}^2+R^2)(2\lambda\sigma_y{}^2+R^2)}}$$

$$g(\xi) = \frac{1}{2\pi\sigma_x\sigma_y} \int_{\mathcal{B}((0,0),R\sqrt{\xi})} \exp\left(-\frac{(x-x_m)^2}{2{\sigma_x}^2} - \frac{(y-y_m)^2}{2{\sigma_y}^2}\right) \mathrm{d}x\mathrm{d}y$$

Laplace Transform

- Lasserre, Zerron (2001)

$$\mathcal{L}(g)(\lambda) = \frac{R^2 \exp\left(-\lambda\left(\frac{{x_m}^2}{2\lambda{\sigma_x}^2+R^2} + \frac{{y_m}^2}{2\lambda{\sigma_y}^2+R^2}\right)\right)}{\lambda\sqrt{(2\lambda{\sigma_x}^2+R^2)(2\lambda{\sigma_y}^2+R^2)}}$$

expansion at ∞

$$= \sum_{i=0}^{\infty} \ell_i \left(\frac{1}{\lambda}\right)^i$$

Borel

$$= \sum_{i=0}^{\infty} \frac{\ell_i}{(i+1)!}\xi^{i+1}$$

$\mathcal{L}(g)$ is D-finite – solution of linear ODE with polynomial coefficients –

ℓ_i can be efficiently symbolically described and computed
–solution of linear recurrence with polynomial coefficients–

$\rightsquigarrow$ Gfun Maple package (Salvy, Zimmermann 1994)

$$g(\xi) = \frac{1}{2\pi\sigma_x\sigma_y} \int_{\mathcal{B}((0,0),R\sqrt{\xi})} \exp\left(-\frac{(x-x_m)^2}{2{\sigma_x}^2} - \frac{(y-y_m)^2}{2{\sigma_y}^2}\right) \mathrm{d}x\mathrm{d}y$$

Laplace Transform

- Lasserre, Zerron (2001)

$$\mathcal{L}(g)(\lambda) = \frac{R^2 \exp\left(-\lambda\left(\frac{{x_m}^2}{2\lambda{\sigma_x}^2+R^2} + \frac{{y_m}^2}{2\lambda{\sigma_y}^2+R^2}\right)\right)}{\lambda\sqrt{(2\lambda{\sigma_x}^2+R^2)(2\lambda{\sigma_y}^2+R^2)}}$$

expansion at ∞

$$= \sum_{i=0}^{\infty} \ell_i \left(\frac{1}{\lambda}\right)^i$$

Borel

$$\simeq \sum_{i=0}^{N} \frac{\ell_i}{(i+1)!} \xi^{i+1}$$

Numerical Issues

$\mathcal{L}(g)$ is D-finite – solution of linear ODE with polynomial coefficients –

ℓ_i can be efficiently symbolically described and computed
–solution of linear recurrence with polynomial coefficients–

$\rightsquigarrow$ Gfun Maple package (Salvy, Zimmermann 1994)

Cancellation in finite precision power series evaluation

Example: $\sigma_x = 115, \sigma_y = 1.41, x_m = 0.15, y_m = 3.88, R = 15$

$g(\xi) = \sum_{i=0}^{\infty} \frac{\ell_i}{(i+1)!} \xi^{i+1}$

Cancellation in finite precision power series evaluation

Example: $\sigma_x = 115, \sigma_y = 1.41, x_m = 0.15, y_m = 3.88, R = 15$

$g(\xi) = \sum\limits_{i=0}^{\infty} \frac{\ell_i}{(i+1)!}\xi^{i+1}$

$$g(1) = 0.16 \cdot 10^{-1} + 1.5 + 16.1 - 250 \ldots + 2.2 \cdot 10^{19} - 2.6 \cdot 10^{19} - \ldots + 4.3 - 0.14 - 0.60 \ldots$$

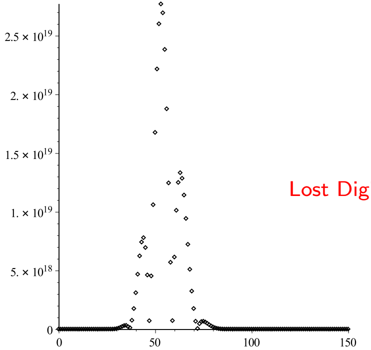
Cancellation in finite precision power series evaluation

Example: $\sigma_x = 115, \sigma_y = 1.41, x_m = 0.15, y_m = 3.88, R = 15$

$$g(\xi) = \sum_{i=0}^{\infty} \frac{\ell_i}{(i+1)!}\xi^{i+1}$$

$$g(1) = 0.16 \cdot 10^{-1} + 1.5 + 16.1 - 250 \ldots + 2.2 \cdot 10^{19} - 2.6 \cdot 10^{19} - \ldots + 4.3 - 0.14 - 0.60 \ldots$$

Values of $\left|\frac{\ell_i}{(i+1)!}\right|$, compared to $g(1) \simeq 0.1004$:



Lost Digits $\simeq \log \dfrac{\max_i \left|\frac{\ell_i}{(i+1)!}\right|}{|g(1)|}$

Cancellation in finite precision power series evaluation

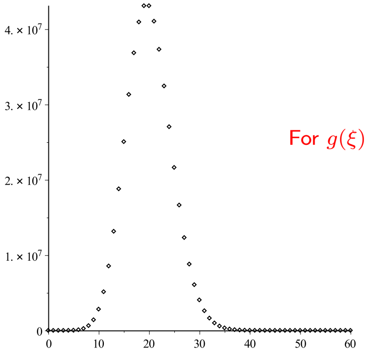
Example: $\exp(-x) = \sum_{i=0}^{\infty} \frac{(-1)^i x^i}{i!}$

$$\exp(-20) = 1 - 20 \ldots + 1.66 \cdot 10^7 - 1.23 \cdot 10^7 + \ldots + 1.19 \cdot 10^{-8} - 3.45 \cdot 10^{-9} \ldots$$

Example: $\exp(-x) = \sum_{i=0}^{\infty} \frac{(-1)^i x^i}{i!}$

$$\exp(-20) = 1 - 20 \ldots + 1.66 \cdot 10^7 - 1.23 \cdot 10^7 + \ldots + 1.19 \cdot 10^{-8} - 3.45 \cdot 10^{-9} \ldots$$

Values of $\left|\frac{(-1)^i 20^i}{i!}\right|$, compared to $\exp(-20) \simeq 2.06 \cdot 10^{-9}$:

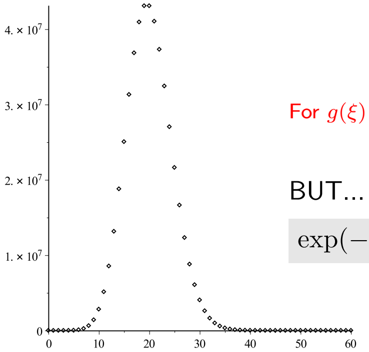


For $g(\xi) = \sum_{i=0}^{\infty} g_i \xi^i$, lost digits $\simeq \log \frac{\max_i |g_i \xi^i|}{|g(\xi)|}$

Example: $\exp(-x) = \sum_{i=0}^{\infty} \frac{(-1)^i x^i}{i!}$

$$\exp(-20) = 1 - 20 \ldots + 1.66 \cdot 10^7 - 1.23 \cdot 10^7 + \ldots + 1.19 \cdot 10^{-8} - 3.45 \cdot 10^{-9} \ldots$$

Values of $\left| \frac{(-1)^i 20^i}{i!} \right|$, compared to $\exp(-20) \simeq 2.06 \cdot 10^{-9}$:



For $g(\xi) = \sum_{i=0}^{\infty} g_i \xi^i$, lost digits $\simeq \log \frac{\max_i |g_i \xi^i|}{|g(\xi)|}$

BUT...

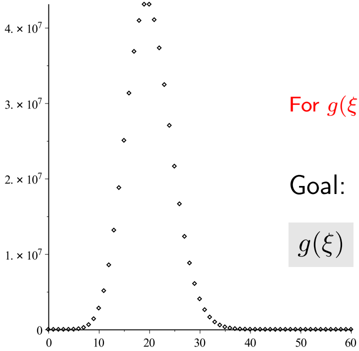
$$\boxed{\exp(-x)} = \frac{\boxed{1}}{\boxed{\exp(x)}}$$

No cancellation!

Example: $\exp(-x) = \sum_{i=0}^{\infty} \frac{(-1)^i x^i}{i!}$

$$\exp(-20) = 1 - 20 \ldots + 1.66 \cdot 10^7 - 1.23 \cdot 10^7 + \ldots + 1.19 \cdot 10^{-8} - 3.45 \cdot 10^{-9} \ldots$$

Values of $\left|\frac{(-1)^i 20^i}{i!}\right|$, compared to $\exp(-20) \simeq 2.06 \cdot 10^{-9}$:



For $g(\xi) = \sum_{i=0}^{\infty} g_i \xi^i$, lost digits $\simeq \log \frac{\max_i |g_i \xi^i|}{|g(\xi)|}$

Goal:

$$g(\xi) = \frac{f(\xi)}{h(\xi)}$$

No cancellation!

$$\underbrace{h(\xi)\,g(\xi)}_{f(\xi)} = h(\xi)\int \dots$$

Laplace Transform

$$\mathcal{L}(f)(\lambda) = \mathcal{L}(g)(\lambda - pR^2) = \dots$$

expansion at ∞

$$\hat{f}(\lambda) := \frac{\mathcal{L}(f)(\lambda^{-1})}{\lambda^2}$$

$$\hat{f}(\lambda) = \sum_{i=0}^{\infty}(i+1)!c_i\lambda^i$$

Borel

$$= \sum_{i=0}^{\infty} c_i \xi^{i+1}$$

Gawronski, Müller, Reinhard (2007): choose $h(\xi) = \exp(pR^2\,\xi)$, with $p \sim \dfrac{1}{2\sigma_y^2}$.

$\mathcal{L}(f)$ and $\hat{f}$ are D-finite

- c_i can be efficiently symbolically described and computed
  –solution of linear recurrence with polynomial coefficients–

$$\underbrace{h(\xi)\,g(\xi)}_{f(\xi)} = h(\xi) \int \ldots$$

Laplace Transform

$$\mathcal{L}(f)(\lambda) = \mathcal{L}(g)(\lambda - pR^2) = \ldots$$

expansion at ∞

$$\hat{f}(\lambda) := \frac{\mathcal{L}(f)(\lambda^{-1})}{\lambda^2}$$

$$\hat{f}(\lambda) = \sum_{i=0}^{\infty} (i+1)!c_i\lambda^i$$

Borel

Reduced Cancellation

$$\simeq \sum_{i=0}^{N} c_i \xi^{i+1}$$

Gawronski, Müller, Reinhard (2007): choose $h(\xi) = \exp(pR^2\,\xi)$, with $p \sim \dfrac{1}{2\sigma_y^2}$.

$\mathcal{L}(f)$ and $\hat{f}$ are D-finite

- c_i can be efficiently symbolically described and computed –solution of linear recurrence with polynomial coefficients–
- c_i are positive $\rightsquigarrow$ Reduced Cancellation

$$\underbrace{h(1)\,g(1)}_{f(1)} = h(1)\int \dots$$

Laplace Transform

$$\mathcal{L}(f)(\lambda) = \mathcal{L}(g)(\lambda - pR^2) = \dots$$

expansion at ∞

$$\hat{f}(\lambda) := \frac{\mathcal{L}(f)(\lambda^{-1})}{\lambda^2}$$

$$\hat{f}(\lambda) = \sum_{i=0}^{\infty}(i+1)!c_i\lambda^i$$

Borel

Reduced Cancellation

$$\simeq \sum_{i=0}^{N} c_i$$

Gawronski, Müller, Reinhard (2007): choose $h(\xi) = \exp(pR^2\,\xi)$, with $p \sim \dfrac{1}{2\sigma_y^2}$.

$\mathcal{L}(f)$ and $\hat{f}$ are D-finite

- c_i can be efficiently symbolically described and computed –solution of linear recurrence with polynomial coefficients–
- c_i are positive $\rightsquigarrow$ Reduced Cancellation

$$\hat{f}(\lambda) := \frac{\mathcal{L}(f)(\lambda^{-1})}{\lambda^2} = \frac{\pi R^2 \rho(0,0) \exp\left(\omega_y R^2 \lambda - \frac{\omega_x}{p\varphi} - \frac{\omega_x}{p\varphi(p\varphi R^2 \lambda - 1)}\right)}{(1 - pR^2\lambda)\sqrt{1 - p\varphi R^2 \lambda}}$$

$$\left\{ p = \frac{1}{2\sigma_y^2} \qquad \varphi = 1 - \frac{\sigma_y^2}{\sigma_x^2} \qquad \omega_x = \frac{x_m^2}{4\sigma_x^4} \qquad \omega_y = \frac{y_m^2}{4\sigma_y^4} \right\}$$

$$\hat{f}(\lambda) := \frac{\mathcal{L}(f)(\lambda^{-1})}{\lambda^2} = \frac{\pi R^2 \rho(0,0) \exp\left(\omega_y R^2 \lambda - \frac{\omega_x}{p\varphi} - \frac{\omega_x}{p\varphi(p\varphi R^2 \lambda - 1)}\right)}{(1 - pR^2\lambda)\sqrt{1 - p\varphi R^2 \lambda}}$$
$$= \sum_{n=0}^{+\infty} c_n (n+1)! \lambda^n$$

$$\left\{ p = \frac{1}{2\sigma_y^2} \qquad \varphi = 1 - \frac{\sigma_y^2}{\sigma_x^2} \qquad \omega_x = \frac{x_m^2}{4\sigma_x^4} \qquad \omega_y = \frac{y_m^2}{4\sigma_y^4} \right\}$$

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

$$\varphi(\lambda) = \omega_y R^2 + \frac{p\varphi R^2}{2(1 - p\varphi R^2 \lambda)} + \frac{pR^2}{1 - pR^2\lambda} + \frac{\omega_x R^2}{(1 - p\varphi R^2 \lambda)^2}$$

$$\hat{f}(\lambda) := \frac{\mathcal{L}(f)(\lambda^{-1})}{\lambda^2} = \frac{\pi R^2 \rho(0,0) \exp\left(\omega_y R^2 \lambda - \frac{\omega_x}{p\varphi} - \frac{\omega_x}{p\varphi(p\varphi R^2 \lambda - 1)}\right)}{(1 - pR^2\lambda)\sqrt{1 - p\varphi R^2 \lambda}}$$

$$= \sum_{n=0}^{+\infty} c_n (n+1)! \lambda^n$$

$$\left\{ p = \frac{1}{2\sigma_y^2} \qquad \varphi = 1 - \frac{\sigma_y^2}{\sigma_x^2} \qquad \omega_x = \frac{x_m^2}{4\sigma_x^4} \qquad \omega_y = \frac{y_m^2}{4\sigma_y^4} \right\}$$

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

$$\varphi(\lambda) = \omega_y R^2 + \frac{p\varphi R^2}{2(1 - p\varphi R^2 \lambda)} + \frac{pR^2}{1 - pR^2\lambda} + \frac{\omega_x R^2}{(1 - p\varphi R^2\lambda)^2}$$

$$\rightsquigarrow \quad \varphi(\lambda) = \frac{P(\lambda)}{Q(\lambda)} \qquad \begin{cases} Q(\lambda) = (1 - p\varphi R^2\lambda)^2 (1 - pR^2\lambda) \\ \qquad\quad = 1 - Q_1\lambda + Q_2\lambda^2 - Q_3\lambda^3 \\ P(\lambda) = P_0 - P_1\lambda + P_2\lambda^2 - P_3\lambda^3 \end{cases}$$

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

$$\lambda \hat{f}(\lambda) = \sum_{n=1}^{+\infty} c_{n-1} n! \lambda^n$$

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

$$\lambda \hat{f}(\lambda) = \sum_{n=1}^{+\infty} \frac{c_{n-1}}{n+1} (n+1)! \lambda^n$$

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

$$\lambda^{k} \hat{f}(\lambda) = \sum_{n=k}^{+\infty} \frac{c_{n-1}}{(n+1)\dots(n-k+2)} (n+1)! \lambda^n$$

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

$$\lambda^k \hat{f}(\lambda) = \sum_{n=k}^{+\infty} \frac{c_{n-1}}{(n+1)\dots(n-k+2)} (n+1)! \lambda^n \qquad \lambda \hat{f}'(\lambda) = \sum_{n=0}^{+\infty} (n c_n)(n+1)! \lambda^n$$

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

$$\lambda^k \hat{f}(\lambda) = \sum_{n=k}^{+\infty} \frac{c_{n-1}}{(n+1)\dots(n-k+2)} (n+1)! \lambda^n \qquad \lambda \hat{f}'(\lambda) = \sum_{n=0}^{+\infty} (n c_n)(n+1)! \lambda^n$$

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

Orbital collision probability evaluation

Step 3: Recurrence computation

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

$$\lambda^k \hat{f}(\lambda) = \sum_{n=k}^{+\infty} \frac{c_{n-1}}{(n+1)\dots(n-k+2)} (n+1)! \lambda^n \qquad \lambda \hat{f}'(\lambda) = \sum_{n=0}^{+\infty} (n c_n)(n+1)! \lambda^n$$

$$Q(\lambda)\lambda \hat{f}'(\lambda) - P(\lambda)\lambda \hat{f}(\lambda) = 0 \qquad \leftrightsquigarrow$$

$$\boxed{\begin{aligned} n c_n - \frac{Q_1(n-1)+P_0}{n+1} c_{n-1} + \frac{Q_2(n-2)+P_1}{(n+1)n} c_{n-2} \\ - \frac{Q_3(n-3)+P_2}{(n+1)n(n-1)} c_{n-3} + \frac{P_3}{(n+1)n(n-1)(n-2)} c_{n-4} = 0 \end{aligned}}$$

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

$$\hat{f}(\lambda) = \sum_{n=0}^{+\infty} \underbrace{c_n (n+1)!}_{\hat{f}_n} \lambda^n$$

$$\varphi(\lambda) = \omega_y R^2 + \frac{p\varphi R^2}{2(1 - p\varphi R^2 \lambda)} + \frac{pR^2}{1 - pR^2 \lambda} + \frac{\omega_x R^2}{(1 - p\varphi R^2 \lambda)^2}$$

$$\varphi(\lambda) := \sum_{n=0}^{\infty} \underbrace{\beta_n}_{\text{positive}} \lambda^n$$

Positivity (by induction):

$$(n+1)\hat{f}_{n+1} = \sum_{k=0}^{n} \beta_k \hat{f}_{n-k} > 0.$$

Orbital collision probability evaluation

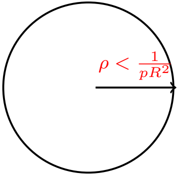
Step 2: Reliable numerics

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

$$\Big\uparrow \text{Borel}$$

$$\boxed{\hat{f}(\lambda) = \sum_{n=0}^{\infty} \underbrace{(n+1)!c_n}_{\hat{f}_n} \lambda^n}$$

Closed-form; Conv. radius $1/(pR^2)$



Tail bounds:

$$\sum_{k=0}^{\infty} \frac{\hat{f}_{k+N} \xi^{k+N+1}}{(k+N+1)!}$$

$$\leq \frac{\rho \left(\frac{\xi}{\rho}\right)^{N+1}}{(N+1)!} \hat{f}(\rho)$$

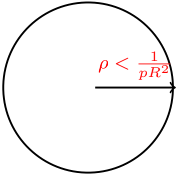
for any $\rho < \frac{1}{pR^2}$ and $n \geq n_0$, with $n_0 = \left\lceil \frac{\xi}{\rho} \right\rceil$.

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

Borel

$$\boxed{\hat{f}(\lambda) = \sum_{n=0}^{\infty} \underbrace{(n+1)! c_n}_{\hat{f}_n} \lambda^n}$$

Closed-form; Conv. radius $1/(pR^2)$



Tail bounds:

$$\sum_{k=0}^{\infty} \frac{\hat{f}_{k+N} \xi^{k+N+1}}{(k+N+1)!} \leq \frac{\left(\frac{\xi}{\rho}\right)^{N+1}}{(N+1)!} \sum_{k=0}^{\infty} \hat{f}_{k+N} \xi^k \rho^{N+1} \underbrace{\frac{(N+1)!}{(k+N+1)!}}_{\leq 1/(N+1)^k}$$

$$\leq \frac{\rho \left(\frac{\xi}{\rho}\right)^{N+1}}{(N+1)!} \underbrace{\sum_{k=0}^{\infty} \hat{f}_{k+N} \rho^{k+N} \underbrace{\frac{\xi^k}{\rho^k (n+1)^k}}_{\leq 1}}_{\bar{\varphi}(\rho)}$$

$$\leq \frac{\rho \left(\frac{\xi}{\rho}\right)^{N+1}}{(N+1)!} \hat{f}(\rho)$$

for any $\rho < \frac{1}{pR^2}$ and $n \geq n_0$, with $n_0 = \left\lceil \frac{\xi}{\rho} \right\rceil$.

In particular, set $\rho = \frac{1}{2pR^2}$ and $\xi = 1$:

$$\sum_{n=N}^{\infty} c_n \leq \frac{(2pR^2)^N}{(N+1)!} \hat{f}\left(\frac{1}{2pR^2}\right)$$

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

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

Algorithm for the Orbital Collision Evaluation

Sum-up

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 terms

Algorithm for the Orbital Collision Evaluation

Sum-up

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 for the Orbital Collision Evaluation

Sum-up

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 terms

unroll the recurrence

sum the terms

Algorithm for the Orbital Collision Evaluation

Sum-up

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

preconditioning
to ensure $c_n \geq 0$

initial terms

unroll the recurrence

sum the terms

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

explicit truncation error bounds

preconditioning to ensure $c_n \geq 0$

initial terms

unroll the recurrence

sum the terms

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 $c_n \geq 0$

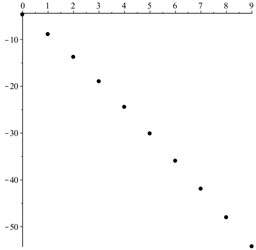
initial terms unroll the recurrence sum the terms

[JGCD, 2016] joint work with R. Serra, D. Arzelier, A. Rondepierre, J.-B. Lasserre, B. Salvy
2017: CNES impl+tests; 2021: CNES on-board impl
2023: https://lejournal.cnrs.fr/articles/un-algorithme-pour-eviter-les-debris-spatiaux

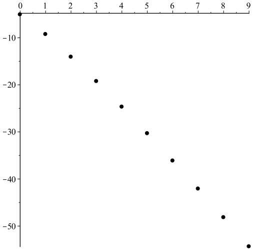
Case #	Input parameters (km)				
	σ_x	σ_y	R	x_m	y_m
1	0.05	0.025	0.005	0.01	0
2	0.05	0.025	0.005	0	0.01
3	0.075	0.025	0.005	0.01	0
4	0.075	0.025	0.005	0	0.01
5	3	1	0.01	1	0
6	3	1	0.01	0	1
7	3	1	0.01	10	0
8	3	1	0.01	0	10
9	10	1	0.01	10	0
10	10	1	0.01	0	10
11	3	1	0.05	5	0
12	3	1	0.05	0	5

Examples: **accuracy** η **and** $\log$ **plot of terms in the series**

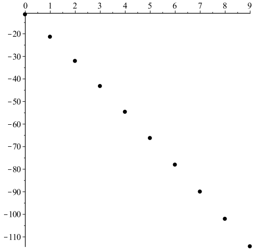
Case 1: $\eta = 23$



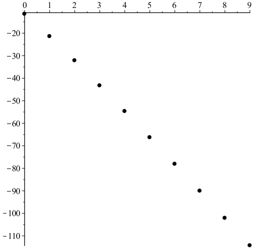
Case 2: $\eta = 22$



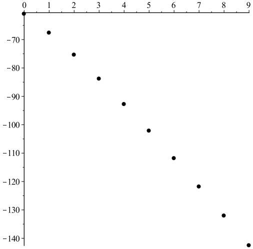
Case 4: $\eta = 22$



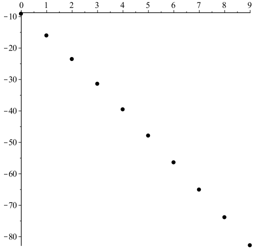
Case 6: $\eta = 47$



Case 8: $\eta = 33$



Case 11: $\eta = 34$



Outline

- The problem of orbital collision risk assessment and mitigation
- Evaluation of a *special function*:
  - Reliable double-precision implementation by power series
  - Roundoff error analysis
- Extensions & Conclusion

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**

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

$s = s + c_n$;

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 $\mathcal{P}$.

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

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

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

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

- standard binary64 (no interval arithmetic, no multiprecision)
  $\rightsquigarrow$ efficiency/architecture constraints
- no a priori small ranges for the parameters
- N can be large! $(N = 100,\ 1000,\ 10000, \dots)$

$\Rightarrow$ A priori bounds using majorizing series techniques for roundoff analysis of linear recurrences following Marc Mezzarobba's work

$\Rightarrow$ Initial and local errors using standard roundoff analysis

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 for arithmetic operations $\star \in \{+, -, \times, \div, \sqrt{}\}$

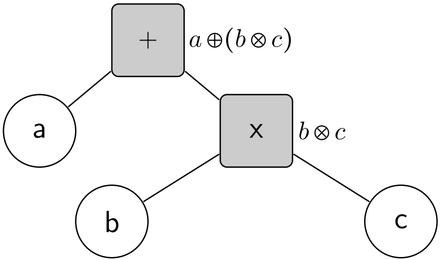
$$\widetilde{a \star b} = (a \star b)(1 + e) \qquad |e| \leq u := 2^{-p}$$

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 for arithmetic operations $\star \in \{+, -, \times, \div, \sqrt{}\}$

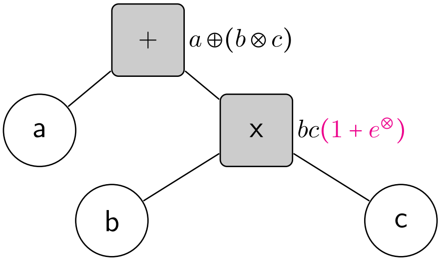


Error model for floating-point arithmetic

- Precision p binary floating-point arithmetic

(e.g., $p = 53$ for `binary64`)

- unbounded exponent range ($\Rightarrow$ no underflow, no overflow)
- round-to-nearest for arithmetic operations $\star \in \{+, -, \times, \div, \sqrt{}\}$

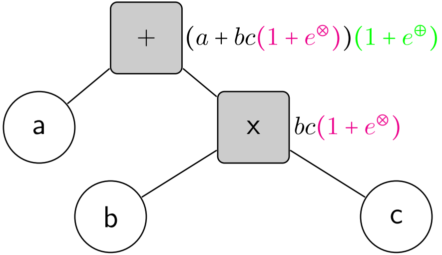


Error model for floating-point arithmetic

- Precision p binary floating-point arithmetic

(e.g., $p = 53$ for `binary64`)

- unbounded exponent range ($\Rightarrow$ no underflow, no overflow)
- round-to-nearest for arithmetic operations $\star \in \{+, -, \times, \div, \sqrt{}\}$

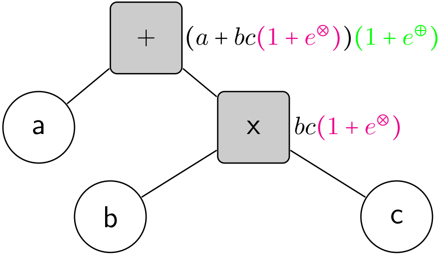


Error model for floating-point arithmetic

- Precision p binary floating-point arithmetic

(e.g., $p = 53$ for `binary64`)

- unbounded exponent range ($\Rightarrow$ no underflow, no overflow)
- round-to-nearest for arithmetic operations $\star \in \{+, -, \times, \div, \sqrt{}\}$



Example

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

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

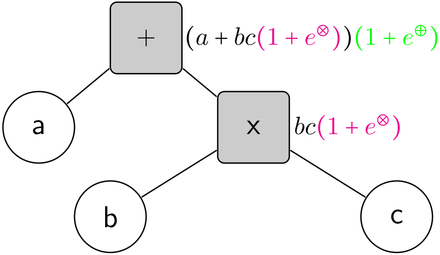
$$|e^{\oplus}|, |e^{\otimes}| \leq u := 2^{-p}$$

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 for arithmetic operations $\star \in \{+, -, \times, \div, \sqrt{}\}$



Example

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

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

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

Initial and local errors using standard roundoff analysis

- Initial errors (loop-independent parameters):
  $\omega_x = \dfrac{x_m^2}{4\sigma_x^4} \to |\widetilde{\omega}_x - \omega_x| \leq \gamma_5 \omega_x \quad \rightsquigarrow \quad$ Similar bounds for p, ω_y

Initial and local errors using standard roundoff analysis

- Initial errors (loop-independent parameters):

$\omega_x = \dfrac{x_m^2}{4\sigma_x^4} \to |\widetilde{\omega}_x - \omega_x| \le \gamma_5 \omega_x \quad \rightsquigarrow \quad$ Similar bounds for p, ω_y

$\varphi = 1 - \dfrac{\sigma_y^2}{\sigma_x^2} \to |\widetilde{\varphi} - \varphi| \le \gamma_4 \quad \rightsquigarrow \quad$ Relative bounds w.r.t. $\varphi = 1$

(for P_i, Q_i which depend on φ)

Initial and local errors using standard roundoff analysis

- Initial errors (loop-independent parameters):

$\omega_x = \dfrac{x_m^2}{4\sigma_x^4} \to |\widetilde{\omega}_x - \omega_x| \leq \gamma_5 \omega_x \qquad \rightsquigarrow \qquad$ Similar bounds for p, ω_y

$\varphi = 1 - \dfrac{\sigma_y^2}{\sigma_x^2} \to |\widetilde{\varphi} - \varphi| \leq \gamma_4 \qquad \rightsquigarrow \qquad$ Relative bounds w.r.t. $\varphi = 1$

(for P_i, Q_i which depend on φ)

$$Q_1 = pR^2(2\varphi + 1) \qquad \rightsquigarrow \qquad |\widetilde{Q}_1 - Q_1| \leq 3pR^2\gamma_9$$

Initial and local errors using standard roundoff analysis

- Initial errors (loop-independent parameters):

$\omega_x = \dfrac{x_m^2}{4\sigma_x^4} \to |\widetilde{\omega}_x - \omega_x| \leq \gamma_5 \omega_x \quad \rightsquigarrow \quad$ Similar bounds for p, ω_y

$\varphi = 1 - \dfrac{\sigma_y^2}{\sigma_x^2} \to |\widetilde{\varphi} - \varphi| \leq \gamma_4 \quad \rightsquigarrow \quad$ Relative bounds w.r.t. $\varphi = 1$

(for P_i, Q_i which depend on φ)

$$Q_1 = pR^2(2\varphi + 1) \quad \rightsquigarrow \quad |\widetilde{Q}_1 - Q_1| \leq \overbrace{3pR^2}^{Q_1^\sharp}\gamma_9 \qquad P_i^\sharp, Q_i^\sharp := P_i, Q_i\,\{\varphi \leftarrow 1\}$$

Initial and local errors using standard roundoff analysis

- Local errors ε_n (at each iteration):

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

- Local errors ε_n (at each iteration):

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

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

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

- Local errors ε_n (at each iteration):

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

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

$\gamma = \gamma_{40}$

$P_i^\sharp, Q_i^\sharp \stackrel{\text{def}}{=} P_i, Q_i \; \{\varphi \leftarrow 1\}$

Local Error $\rightarrow$ Global Error

$\widetilde{c}_n \quad \rightsquigarrow \quad$ FP evaluation of c_n

Global Error $\delta_n := \widetilde{c}_n - c_n$

Local Error $\rightarrow$ Global Error

$\widetilde{c}_n \quad \rightsquigarrow \quad$ FP evaluation of c_n Global Error $\delta_n := \widetilde{c}_n - c_n$

- The local error ε_n made at each iteration:

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

$\widetilde{c}_n \quad \rightsquigarrow \quad$ FP evaluation of c_n Global Error $\delta_n := \widetilde{c}_n - c_n$

- The local error ε_n made at each iteration:

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

$$0 = c_n - \left(\frac{Q_1(n-1)+P_0}{(n+1)n} c_{n-1} - \frac{Q_2(n-2)+P_1}{(n+1)n^2} c_{n-2} \right.$$
$$\left. + \frac{Q_3(n-3)+P_2}{(n+1)n^2(n-1)} c_{n-3} - \frac{P_3}{(n+1)n^2(n-1)(n-2)} c_{n-4} \right)$$

Local Error $\to$ Global Error

$\widetilde{c}_n \quad \rightsquigarrow \quad$ FP evaluation of c_n Global Error $\delta_n := \widetilde{c}_n - c_n$

- The local error ε_n made at each iteration:

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

$$0 = c_n - \left(\frac{Q_1(n-1)+P_0}{(n+1)n}c_{n-1} - \frac{Q_2(n-2)+P_1}{(n+1)n^2}c_{n-2} + \frac{Q_3(n-3)+P_2}{(n+1)n^2(n-1)}c_{n-3} - \frac{P_3}{(n+1)n^2(n-1)(n-2)}c_{n-4}\right)$$

- Obtain a linear recurrence satisfied by δ_n (involving ε_n)

$$\varepsilon_n = \delta_n - \left(\frac{Q_1(n-1)+P_0}{(n+1)n}\delta_{n-1} - \frac{Q_2(n-2)+P_1}{(n+1)n^2}\delta_{n-2} + \frac{Q_3(n-3)+P_2}{(n+1)n^2(n-1)}\delta_{n-3} - \frac{P_3}{(n+1)n^2(n-1)(n-2)}\delta_{n-4}\right)$$

following [Mezzarobba 2020]

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

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

following [Mezzarobba 2020]

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

$$\boxed{\begin{array}{c} c_n - \square\, c_{n-1} + \square\, c_{n-2} \\ - \square\, c_{n-3} + \square\, c_{n-4} = 0 \end{array}} \qquad \boxed{\begin{array}{c} \delta_n - \square\, \delta_{n-1} + \square\, \delta_{n-2} \\ - \square\, \delta_{n-3} + \square\, \delta_{n-4} = \varepsilon_n \end{array}}$$

$$f(1) = s = \sum_{n=0}^{+\infty} c_n \xrightarrow[\delta_n \overset{\text{def}}{=} \widetilde{c}_n - c_n]{} \delta(1) = \sum_{n=0}^{+\infty} \underbrace{(\widetilde{c}_n - c_n)}_{\delta_n}$$

Global error modeling using series

following [Mezzarobba 2020]

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

Diff. eq. on $f(\xi) = 0$

Diff. eq. on $\delta(\xi) = \varepsilon(\xi)$

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

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

bound ε_n

$$f(1) = s = \sum_{n=0}^{+\infty} c_n \xrightarrow[\delta_n \overset{\text{def}}{=} \widetilde{c}_n - c_n]{} \delta(1) = \sum_{n=0}^{+\infty} \underbrace{(\widetilde{c}_n - c_n)}_{\delta_n}$$

Global error modeling using series

following [Mezzarobba 2020]

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

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

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

$$\boxed{\begin{array}{c} c_n - \square\, c_{n-1} + \square\, c_{n-2} \\ - \square\, c_{n-3} + \square\, c_{n-4} = 0 \end{array}} \qquad \boxed{\begin{array}{c} \delta_n - \square\, \delta_{n-1} + \square\, \delta_{n-2} \\ - \square\, \delta_{n-3} + \square\, \delta_{n-4} = \varepsilon_n \end{array}}$$

bound ε_n

$$f(1) = s = \sum_{n=0}^{+\infty} c_n \xrightarrow[\delta_n \overset{\text{def}}{=} \widetilde{c}_n - c_n]{} \delta(1) = \sum_{n=0}^{+\infty} \underbrace{(\widetilde{c}_n - c_n)}_{\delta_n}$$

Global error modeling using series

following [Mezzarobba 2020]

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

$$\delta(\xi) = \widetilde{f}(\xi) - f(\xi) = \sum_{n=0}^{+\infty} \delta_n \xi^n$$

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

solve for $\delta(\xi)$

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

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

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

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

bound ε_n

$$f(1) = s = \sum_{n=0}^{+\infty} c_n \xrightarrow[\delta_n \overset{\text{def}}{=} \widetilde{c}_n - c_n]{} \delta(1) = \sum_{n=0}^{+\infty} \underbrace{(\widetilde{c}_n - c_n)}_{\delta_n}$$

Global error modeling using series

following [Mezzarobba 2020]

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

$\widehat{f}(\lambda) = \sum_{n=0}^{+\infty} n! c_n \xi^n$ $\qquad$ $\widehat{\delta}(\lambda) \rightsquigarrow \delta(\xi)$ solve for $\widehat{\delta}(\lambda)$

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

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

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

bound ε_n

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

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

Global error modeling using series

A priori closed-form relative error bound

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

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

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

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

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

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

bound ε_n

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

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

Global error modeling using series

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

$$\boxed{Q(\lambda)\hat{\delta}'(\lambda) - P(\lambda)\hat{\delta}(\lambda) = \hat{\varepsilon}'(\lambda)}$$

Global error modeling using series

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

$$\boxed{Q(\lambda)\hat{\delta}'(\lambda) - P(\lambda)\hat{\delta}(\lambda) = \hat{\varepsilon}'(\lambda)}$$

$$\boxed{\hat{\varepsilon}'(\lambda) \ll \gamma\left(Q^\sharp(\lambda)\hat{f}'(\lambda) + P^\sharp(\lambda)\hat{f}(\lambda) \quad + \quad Q^\sharp(\lambda)\hat{\delta}'(\lambda) + P^\sharp(\lambda)\hat{\delta}(\lambda)\right)}$$

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

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

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

"Linearized" differential inequality

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

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

Since $\dfrac{1}{Q(\lambda)} = \dfrac{1}{(1-p\varphi R^2\lambda)^2(1-pR^2\lambda)} \gg 0$:

“Linearized” differential inequality

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

Since $\dfrac{1}{Q(\lambda)} = \dfrac{1}{(1-p\varphi R^2\lambda)^2(1-pR^2\lambda)} \gg 0$:

$$\hat{\delta}'(\lambda) \ll \varphi(\lambda)\hat{\delta}(\lambda) + \gamma\hat{\psi}(\lambda)\hat{f}(\lambda)$$

$$\hat{\psi}(\lambda) := \frac{Q^\sharp(\lambda)\varphi^\sharp(\lambda) + P^\sharp(\lambda)}{(1-pR^2\lambda)^3} \gg 0$$

$$0 \ll \varphi(\lambda) \ll \varphi^\sharp(\lambda) := \varphi(\lambda)\,\{\varphi \leftarrow 1\}$$

Majorizing equation

$\hat{\delta}(\lambda) \ll \hat{\Delta}(\lambda)$ where:

$$\hat{\delta}'(\lambda) \ll \varphi(\lambda)\hat{\delta}(\lambda) + \gamma\hat{\psi}(\lambda)\hat{f}(\lambda)$$

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

Solving the differential inequality

Majorizing equation

$\hat{\delta}(\lambda) \ll \hat{\Delta}(\lambda)$ where:

$$\hat{\delta}'(\lambda) \ll \varphi(\lambda)\hat{\delta}(\lambda) + \gamma\hat{\psi}(\lambda)\hat{f}(\lambda)$$

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

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

From Laplace back to real world

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

$$\hat{\Delta}(\lambda) = \sum_{n=0}^{+\infty} \delta_n (n+1)!\lambda^n \leftrightsquigarrow \quad \Delta(\xi) = \sum_{n=0}^{+\infty} \delta_n \xi^{n+1}$$

$$\hat{\Delta}(\lambda) = \lambda^{-2}\mathcal{L}\{\Delta\}(\lambda^{-1}) \leftrightsquigarrow \quad \Delta(\xi) = \mathcal{L}^{-1}\{\lambda^{-2}\hat{\Delta}(\lambda^{-1})\}(\xi)$$

$$\hat{\Psi}(\lambda) \leftrightsquigarrow \quad \Psi(\xi) = \mathcal{L}^{-1}\{\hat{\Psi}(\lambda^{-1})\}(\xi)$$

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

$$(\Psi * f)(\xi) := \int_0^\xi \Psi(\tau) f(\xi - \tau)\mathrm{d}\tau \qquad \text{(convolution)}$$

Some explicit calculations

$$|\widetilde{f}(1) - f(1)| \leq \Delta(1) = e_0 f(1) + \gamma\,(\Psi * f)(1)$$

$$|\widetilde{f}(1) - f(1)| \leq \Delta(1) = e_0 f(1) + \gamma\,(\Psi * f)(1)$$

By explicit inverse Laplace transform (using Maple):

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

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

$$|\widetilde{f}(1) - f(1)| \le \Delta(1) = e_0 f(1) + \gamma\,(\Psi * f)(1)$$

By explicit inverse Laplace transform (using Maple):

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

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

Hence $\quad (\Psi * f)(1) \le f(1) \underbrace{\int_0^1 W(\tau)\mathrm{d}\tau}_{:=C}$

A simplified bound

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

$$C := \int_0^1 W(\tau)\mathrm{d}\tau = \frac{7}{96} p^3 \omega_x R^8 + \left(\frac{7}{12} p + \frac{1}{2}\omega_x\right) p^2 R^6 + \left(\frac{9}{4} p + \frac{5}{4}\omega_x + \frac{15}{4}\omega_y\right) p R^4 + \left(\frac{3}{2} p + \omega_x + 3\omega_y\right) R^2.$$

(Complete) differential inequality

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

(Complete) differential inequality

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

$\delta(\xi) \ll \Delta(\xi)$:

(linearized) $$\Delta(\xi) = \delta_0 f(\xi) + \gamma\, (\Psi * f)(\xi)$$

(rigorous) $$\Delta(\xi) = \delta_0 f(\xi) + (1 + \delta_0) \sum_{k=1}^{+\infty} \frac{\gamma^k}{k!} (\Psi^{*k} * f)(\xi)$$

Total roundoff error

Linearized bound

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

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

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

Total roundoff error

Linearized bound

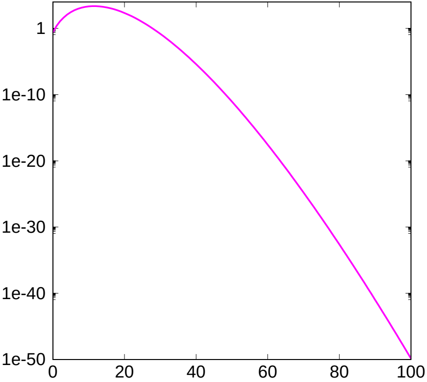
$$\frac{|\widetilde{\mathcal{P}}_{0:N} - \mathcal{P}_{0:N}|}{\mathcal{P}} \leq \left(\frac{2x_m^2}{\sigma_x^2} + \frac{2y_m^2}{\sigma_y^2} + 40C + N + 2pR^2 + 8 \right) u + o(u)$$

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

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

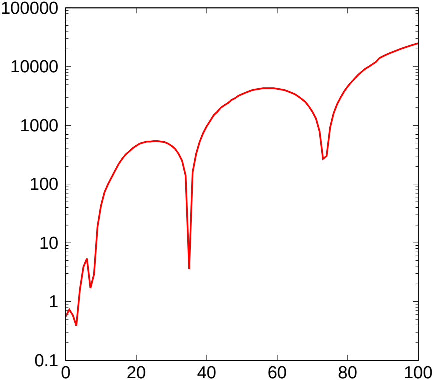
- The required number of terms N depends on the parameters ($\rightarrow$ truncation error bound), and $N \leq C$ in practice
- Proof that a 64-bit exponent emulation is sufficient to avoid overflows for realistic parameter ranges
- Rigorous (i.e., not linearized) total error bound available in ARITH'2024 article with D. Arzelier, F. Bréhard & M. Mezzarobba

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



c_n

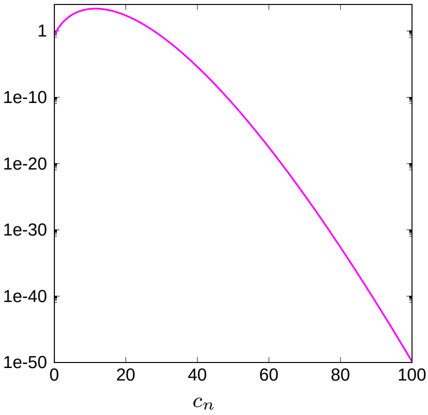
float binary64

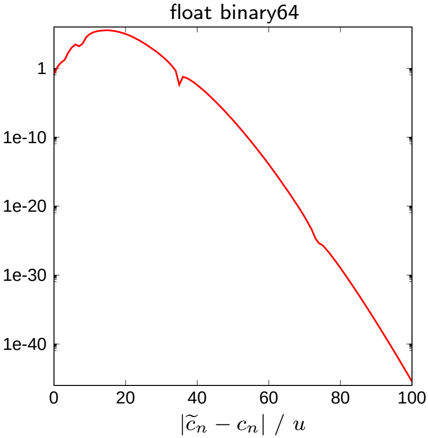


$|\widetilde{c}_n - c_n| \ / \ u c_n$

Analysis of a numerical example

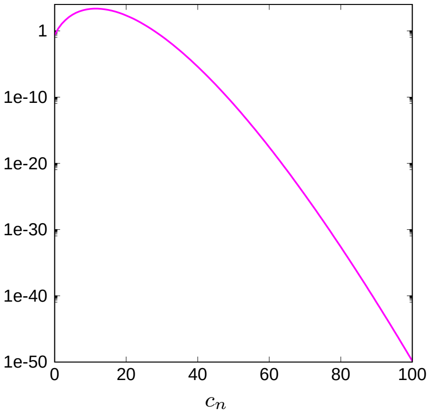
$$R = 5, \quad \sigma_x = 50, \quad \sigma_y = 1, \quad x_m = 10, \quad y_m = 0 \quad \Rightarrow \quad N = 101$$

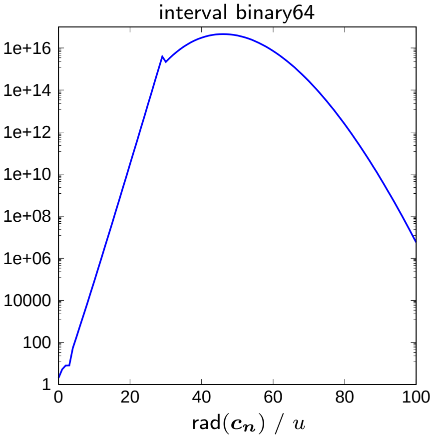




Analysis of a numerical example

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





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

Collision Probability – Extensions and Future Works

- Similar method for the 3D formula [MassonEtAl2024]

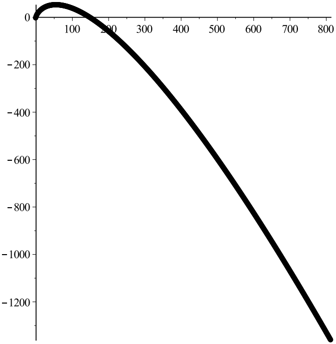
- Saddle-point method for "degenerate" cases

Trickier examples
(from [Alfano 2009])

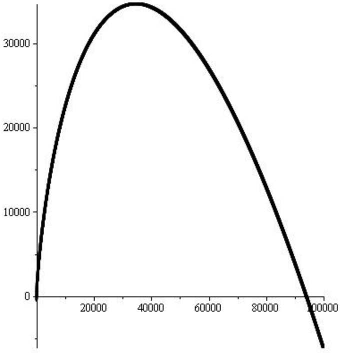
Case #	Input parameters (m)				
	σ_x	σ_y	R	x_m	y_m
3	114.25	1.41	15	0.15	3.88
5	177.8	0.038	10	2.12	-1.22

Alfano's test case number	Probability of collision (-)		
	Alfano	New method	Reference (MC)
3	0.10038	0.10038	0.10085
5	0.044712	0.045509	0.044499

Case 3 Alfano: $\eta_{800} = 30$



Case 5 Alfano: $\eta_{121000} = 20$



Case #	Input parameters (m)				
	σ_x	σ_y	R	x_m	y_m
3	114.25	1.41	15	0.15	3.88
5	177.8	0.038	10	2.12	-1.22

$\rightsquigarrow$ An alternative via Saddle-Point Method

- Similar method for the 3D formula [MassonEtAl2024]
- Saddle-point method for "degenerate" cases

- Sensitivity study to the parameters uncertainty
- Long-term/multiple encounters

- Collision risk assesment and mitigation strategies for the case of mega-constellations (collaboration with industrial partners)

Conclusion

- Symbolic-Numeric approach for floating-point implementation of *special functions*
- Benefit from algorithmic properties of D-finite functions

Conclusion

- Symbolic-Numeric approach for floating-point implementation of *special functions*
- Benefit from algorithmic properties of D-finite functions
- Reduced cancellation evaluation by *pre-condition*
  $\rightsquigarrow$ Gawronski, Müller, Reinhard, *Reduced Cancellation in the Evaluation of Entire Functions and Applications to the Error Function*, SIAM J. Num. Analysis, 2007
  $\rightsquigarrow$ S. Chevillard & M. Mezzarobba, *Multiple-Precision Evaluation of the Airy Ai Function with Reduced Cancellation*, ARITH 2013

Conclusion

- Symbolic-Numeric approach for floating-point implementation of *special functions*
- Benefit from algorithmic properties of D-finite functions
- Reduced cancellation evaluation by *pre-condition*
  $\rightsquigarrow$ Gawronski, Müller, Reinhard, *Reduced Cancellation in the Evaluation of Entire Functions and Applications to the Error Function*, SIAM J. Num. Analysis, 2007
  $\rightsquigarrow$ S. Chevillard & M. Mezzarobba, *Multiple-Precision Evaluation of the Airy Ai Function with Reduced Cancellation*, ARITH 2013
- Roundoff error of linear recurrences with polynomial coefficients
  $\rightsquigarrow$ M. Mezzarobba, *Rounding Error Analysis of Linear Recurrences Using Generating Series*, Electronic Trans. on Num. Analysis, 2023

Conclusion

- Symbolic-Numeric approach for floating-point implementation of *special functions*
- Benefit from algorithmic properties of D-finite functions
- Reduced cancellation evaluation by *pre-condition*
  $\rightsquigarrow$ Gawronski, Müller, Reinhard, *Reduced Cancellation in the Evaluation of Entire Functions and Applications to the Error Function*, SIAM J. Num. Analysis, 2007
  $\rightsquigarrow$ S. Chevillard & M. Mezzarobba, *Multiple-Precision Evaluation of the Airy Ai Function with Reduced Cancellation*, ARITH 2013
- Roundoff error of linear recurrences with polynomial coefficients
  $\rightsquigarrow$ M. Mezzarobba, *Rounding Error Analysis of Linear Recurrences Using Generating Series*, Electronic Trans. on Num. Analysis, 2023
- Approximations of D-finite functions + Sollya
  $\rightsquigarrow$ C. Lauter & M. Mezzarobba, *Semi-Automatic Floating-Point Implementation of Special Functions*, ARITH 2015

Conclusion

- Symbolic-Numeric approach for floating-point implementation of *special functions*
- Benefit from algorithmic properties of D-finite functions
- Reduced cancellation evaluation by *pre-condition*
  $\rightsquigarrow$ Gawronski, Müller, Reinhard, *Reduced Cancellation in the Evaluation of Entire Functions and Applications to the Error Function*, SIAM J. Num. Analysis, 2007
  $\rightsquigarrow$ S. Chevillard & M. Mezzarobba, *Multiple-Precision Evaluation of the Airy Ai Function with Reduced Cancellation*, ARITH 2013
- Roundoff error of linear recurrences with polynomial coefficients
  $\rightsquigarrow$ M. Mezzarobba, *Rounding Error Analysis of Linear Recurrences Using Generating Series*, Electronic Trans. on Num. Analysis, 2023
- Approximations of D-finite functions + Sollya
  $\rightsquigarrow$ C. Lauter & M. Mezzarobba, *Semi-Automatic Floating-Point Implementation of Special Functions*, ARITH 2015
- Fast & accurate evaluation algorithm for collision probability $\rightsquigarrow$ it took 10 years to "gather all the pieces"!

Thank you for your attention!

a D-finite function.

Also check
`http://wayfinder.privateer.com/`
to follow space debris in real time!

Example: constant coefficients recurrence

$$c_n = 2c_{n-1} - c_{n-2} \quad \rightsquigarrow \quad c_n = (n+1)c_0$$

Example: constant coefficients recurrence

$$
\begin{aligned}
c_n = 2c_{n-1} - c_{n-2} \quad &\rightsquigarrow \quad c_n = (n+1)c_0 \\
c(z) = \sum_{n=0}^{+\infty} c_n z^n \quad &\rightsquigarrow \quad c(z) = \frac{c_0}{(1-z)^2}
\end{aligned}
$$

Example: constant coefficients recurrence

$$c_n = 2c_{n-1} - c_{n-2} \quad \rightsquigarrow \quad c_n = (n+1)c_0$$

$$c(z) = \sum_{n=0}^{+\infty} c_n z^n \quad \rightsquigarrow \quad c(z) = \frac{c_0}{(1-z)^2}$$

- Local error: $\varepsilon_n = (2\widetilde{c}_{n-1} - \widetilde{c}_{n-2})e_n, \quad |e_n| \leq u$

Example: constant coefficients recurrence

$$c_n = 2c_{n-1} - c_{n-2} \quad \rightsquigarrow \quad c_n = (n+1)c_0$$

$$c(z) = \sum_{n=0}^{+\infty} c_n z^n \quad \rightsquigarrow \quad c(z) = \frac{c_0}{(1-z)^2}$$

- Local error: $\varepsilon_n = (2\widetilde{c}_{n-1} - \widetilde{c}_{n-2}) e_n, \quad |e_n| \leq u$
- Substitute $\widetilde{c}_n = c_n + \delta_n$:

$$\boxed{\widetilde{c}_n - 2\widetilde{c}_{n-1} + \widetilde{c}_{n-2} = \varepsilon_n}$$

Example: constant coefficients recurrence

$$c_n = 2c_{n-1} - c_{n-2} \quad \rightsquigarrow \quad c_n = (n+1)c_0$$

$$c(z) = \sum_{n=0}^{+\infty} c_n z^n \quad \rightsquigarrow \quad c(z) = \frac{c_0}{(1-z)^2}$$

- Local error: $\varepsilon_n = (2\widetilde{c}_{n-1} - \widetilde{c}_{n-2})e_n, \quad |e_n| \leq u$
- Substitute $\widetilde{c}_n = c_n + \delta_n$:

$$\boxed{\delta_n - 2\delta_{n-1} + \delta_{n-2} = e_n(c_n + 2\delta_{n-1} - \delta_{n-2})}$$

Example: constant coefficients recurrence

$$c_n = 2c_{n-1} - c_{n-2} \quad \rightsquigarrow \quad c_n = (n+1)c_0$$

$$c(z) = \sum_{n=0}^{+\infty} c_n z^n \quad \rightsquigarrow \quad c(z) = \frac{c_0}{(1-z)^2}$$

- Local error: $\varepsilon_n = (2\widetilde{c}_{n-1} - \widetilde{c}_{n-2})e_n, \quad |e_n| \leq u$
- Substitute $\widetilde{c}_n = c_n + \delta_n$:

$$\boxed{\delta_n - 2\delta_{n-1} + \delta_{n-2} = e_n(c_n + 2\delta_{n-1} - \delta_{n-2})}$$

- Convert to series inequality on $\delta(z) = \sum_{n=0}^{+\infty} \delta_n z^n$:

$$\boxed{(1 - 2z + z^2)\delta(z) \ll \frac{u|c_0|}{(1-z)^2} + u(2z + z^2)\delta(z)}$$

where $a(z) \ll b(z)$ means $|a_n| \leq b_n$ for all n

Example: constant coefficients recurrence

- Solve linearized inequality $(1-z)^2\delta(z) \ll \dfrac{u|c_0|}{(1-z)^2}$:

$$\delta(z) \ll \frac{u|c_0|}{(1-z)^4} \quad \rightsquigarrow \quad |\delta_n| \leq u|c_0|\frac{(n+1)(n+2)(n+3)}{6}$$

Example: constant coefficients recurrence

- Solve linearized inequality $(1-z)^2\delta(z) \ll \dfrac{u|c_0|}{(1-z)^2}$:

$$\delta(z) \ll \frac{u|c_0|}{(1-z)^4} \quad \rightsquigarrow \quad |\delta_n| \leq u|c_0|\frac{(n+1)(n+2)(n+3)}{6}$$

- Solve complete inequality $\delta(z) \ll \dfrac{u|c_0|}{(1-z)^4} + u\dfrac{2z+z^2}{(1-z)^2}\delta(z)$:

$$\rightsquigarrow \; \delta(z) \ll \Delta(z) \quad \text{where} \quad \Delta(z) = \frac{u|c_0|}{(1-z)^2[(1-z)^2 - u(2z+z^2)]}$$

Example: constant coefficients recurrence

- Solve linearized inequality $(1-z)^2\delta(z) \ll \dfrac{u|c_0|}{(1-z)^2}$:

$$\delta(z) \ll \frac{u|c_0|}{(1-z)^4} \quad \rightsquigarrow \quad |\delta_n| \leq u|c_0|\frac{(n+1)(n+2)(n+3)}{6}$$

- Solve complete inequality $\delta(z) \ll \dfrac{u|c_0|}{(1-z)^4} + u\dfrac{2z+z^2}{(1-z)^2}\delta(z)$:

$$\rightsquigarrow \ \delta(z) \ll \Delta(z) \quad \text{where} \quad \Delta(z) = \frac{u|c_0|}{(1-z)^2[(1-z)^2 - u(2z+z^2)]}$$

$\rightsquigarrow$ The roots of $(1-z)^2 - u(2z+z^2)$ bifurcate from 1 in $O(u^{\frac{1}{2}})$

$\rightsquigarrow$ $|\delta_n| \leq u|c_0|\dfrac{(n+1)(n+2)(n+3)}{6} + O(u^{\frac{3}{2}})$