Testing the Sharpness of Known Error Bounds of the Fast Fourier Transform

Nicolas Brisebarre[1] Jean-Michel Muller[1] **Joris Picot**[1]

[1]AriC Team, Laboratoire de l'Informatique du Parallélisme, ENS de Lyon

ARITH 2023

4 to 6 September, 2023

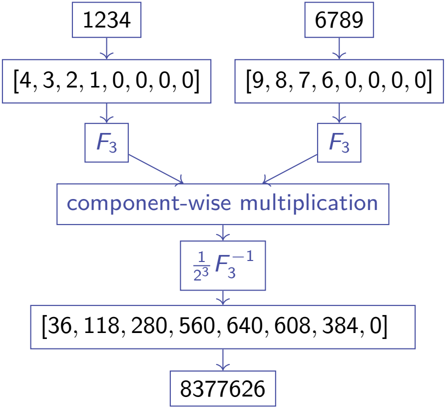
Portland, Oregon, USA

ENS DE LYON

PROJET FINANCÉ PAR L'ANR - PROJECT FUNDED BY THE ANR

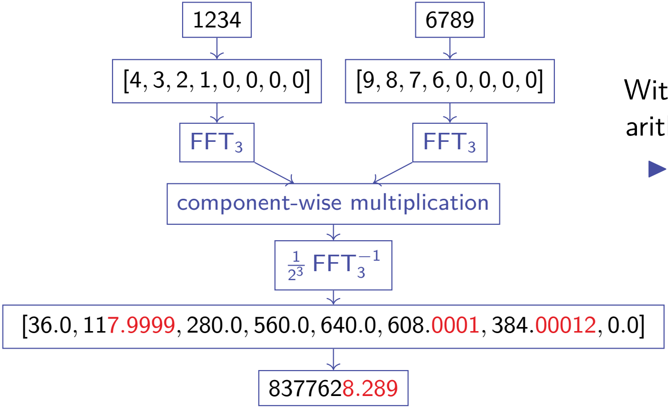
Application of FFT: fast multiplication of huge integers

Multiplication of 1234 by 6789:



Application of FFT: fast multiplication of huge integers

Multiplication of 1234 by 6789:

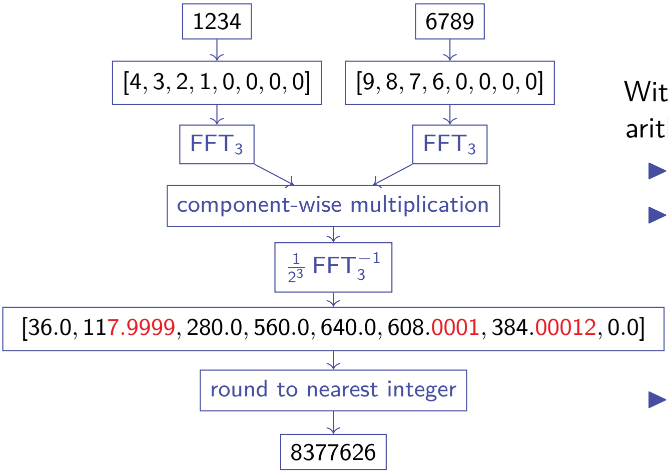


With floating-point arithmetic

- Numerical errors.

Application of FFT: fast multiplication of huge integers

Multiplication of 1234 by 6789:



With floating-point arithmetic

- Numerical errors.
- Here, we can recover the exact result because rounding errors < 0.5.
- But, as size grows, rounding error accumulates.

Motivation

We are interested in

- Absolute bounds on FFT numerical errors
- Tight bounds for improved certification

Two approaches

- Pick a global bound from the literature, valid for all possible inputs, and use conventional floating-point arithmetic
- Use *interval arithmetic* to compute a local bound on the fly, valid only for a reduced set of inputs

Motivation

Preceding work on global bound approach

N. Brisebarre, M. Joldeș, J.-M. Muller, A.-M. Naneș, J. Picot: "Error Analysis of Some Operations Involved in the Cooley-Tukey Fast Fourier Transform", *ACM Trans. Math. Softw. 46(2)*, 2020.

Two antagonistic effects of local bounds

- Local bounds are expected to be tighter than global bound
- Interval arithmetic is prone to overestimation[1]

Goal

Compare global and local bound approaches for the [Cooley & Tukey, 1965] with numerical tests

[1] Liu & Kreinovich [2010]

Notations

$X \in \mathbb{C}^{2^n}$ *input vector* of complex, *floating-point* coefficients

$F_n X$ is the *exact* transform

$\mathrm{FFT}_n(X)$ is the *computed* transform

Measuring errors: the choice of the metric

2-norm relative error

$$e_2(X) = \frac{\left\| F_n X - \mathrm{FFT}_n(X) \right\|_2}{\| F_n X \|_2}$$

where:

$$\|Z\|_2 = \sqrt{|z_0|^2 + \cdots + |z_{2^n-1}|^2}$$

- Natural norm for Fourier transforms
- Many results on the literature are based on it
- ☹ Ill-suited to bound individual coefficients

Measuring errors: the choice of the metric

Component-wise ∞-norm *input-relative* error

$$e_\infty^\perp(X) = \frac{\|F_n X - \mathrm{FFT}_n(X)\|_\infty^\perp}{\|X\|_\infty^\perp}$$

where:

$$\|Z\|_\infty^\perp = \max_{k=0}^{2^n-1} \left\{ |u|, |v| \;\middle|\; u + i\,v \in Z \right\}$$

☺ Represent the maximum error on every FP coefficient

- $\|X\|_\infty^\perp$ removes dependency on input magnitude
- We can deduce the latter from the former:

$$e_\infty^\perp(X) \leqslant 2^n \sqrt{2}\; e_2(X)$$

Sources of error

Excerpt of the FFT algorithm

Pseudo-code

$$\mathtt{y}_j := \mathtt{x}_j + \mathtt{x}_{j+2^{s-1}} \cdot \mathtt{w}_s^j$$
$$\mathtt{y}_{j+2^{s-1}} := \mathtt{x}_j - \mathtt{x}_{j+2^{s-1}} \cdot \mathtt{w}_s^j$$

FP rounding occurs in:

1. FP operations in $+, -, \times$
2. Rounded roots

1. Radix-2, precision-p, unbounded exponent range
   For any FP number $\mathtt{x}$ and $\mathtt{y}$ and $\triangle \in \{+, -, \times\}$:

$$\left|\,\mathsf{RN}(\mathtt{x} \triangle \mathtt{y}) - (\mathtt{x} \triangle \mathtt{y})\right| \leqslant u\,|\mathtt{x} \triangle \mathtt{y}|$$

   where $u = 2^{-p}$ is unit roundoff

2. For any order s: $\left|\mathtt{w}_s^j - \omega_s^j\right| \leqslant \delta_s \leqslant \frac{\sqrt{2}}{2} u$

s	δ_s
1	0
2	0
3	$0.616u$
4	$0.616u$
11	$0.641u$
15	$0.697u$

Global error bound [Brisebarre & al., 2020]

Error bound

Best known error bound

$$e_\infty^\perp(X) \leqslant b^{\text{global}} = 2^n\sqrt{2}\left((1+u)^n \prod_{s=1}^{n}(1+g_s) - 1\right)$$

where g_s is the relative error at step s

$$\begin{cases} g_1 = g_2 = 0 \\ g_s = \delta_s + \rho(1+\delta_s) \end{cases}$$

and ρ is the relative error of the complex multiplication

$$\begin{cases} \rho = u\sqrt{5} & \text{without FMA} \\ \rho = 2u & \text{with FMA} \end{cases}$$

Global error bound [Brisebarre & al., 2020]

Bad case

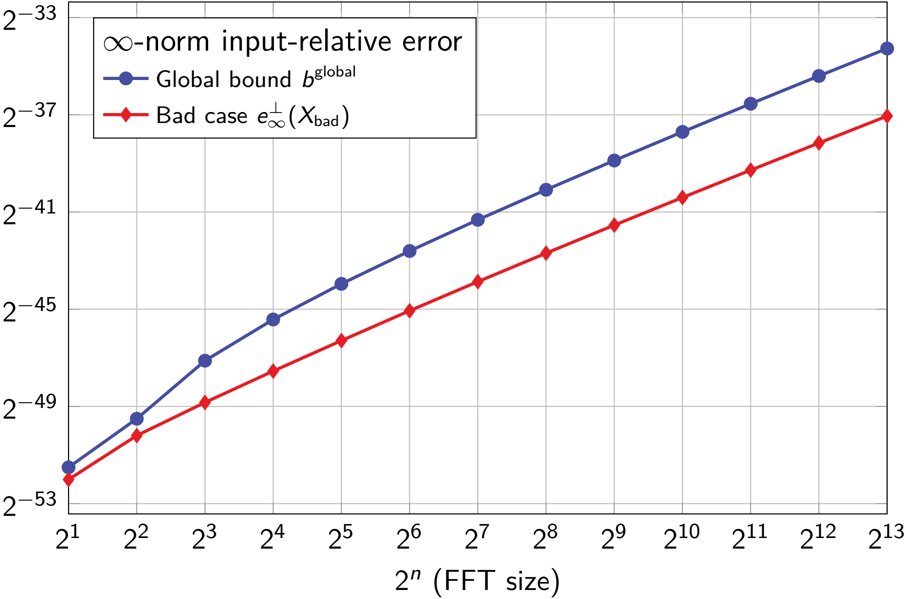
I.e. an input X_{bad} that generates large errors

- An explicit formula for the error can be computed with the help of B. Salvy and the GFUN package[2]:

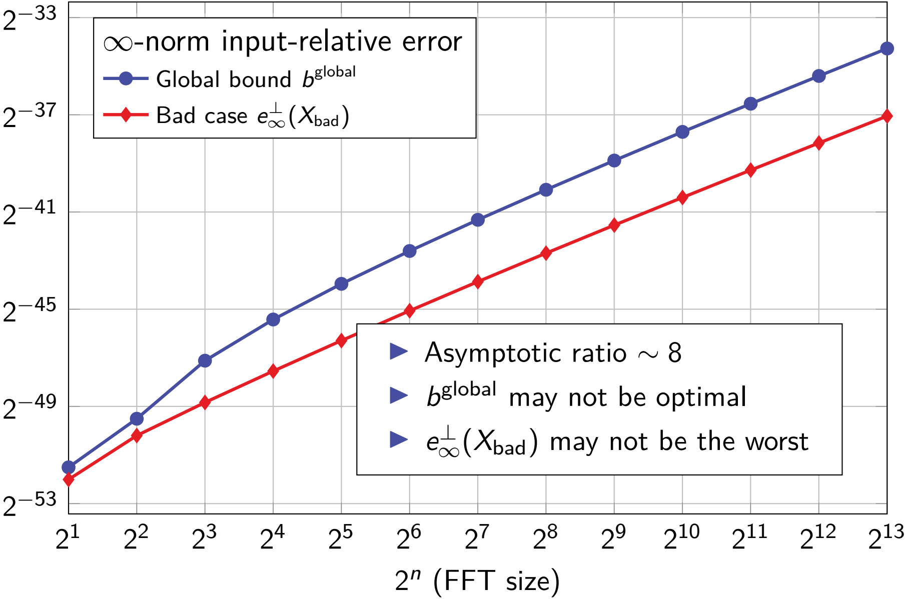
$$e_\infty^\perp(X_{\text{bad}}) = \left(\frac{2^n}{27} + (15n+14) - \frac{5}{9}\cos\left(\frac{n\pi}{3}\right) + \frac{\sqrt{3}}{9}\sin\left(\frac{n\pi}{3}\right) + \frac{(-1)^n}{27}\right)u$$

[2] Salvy & Zimmermann [1994]

Global error bound



Global error bound



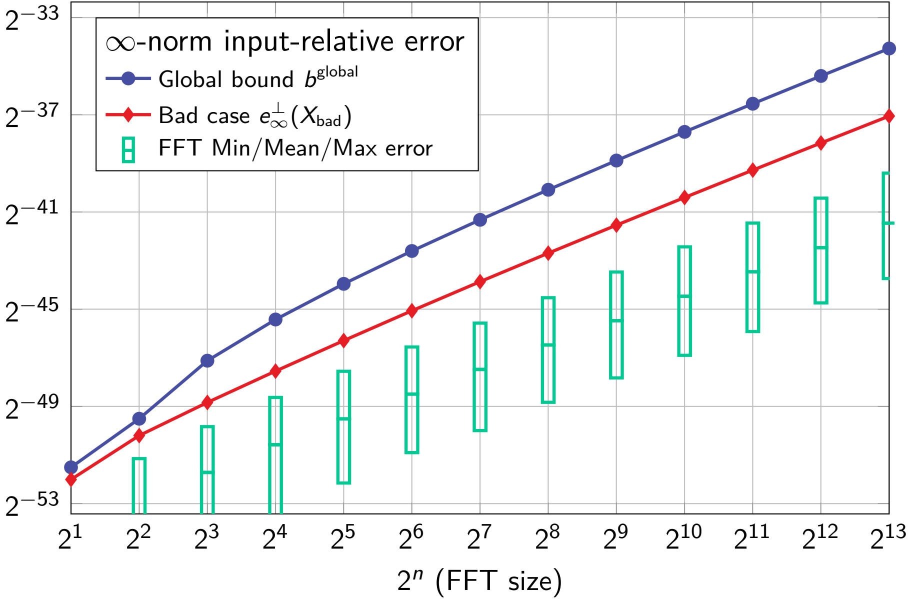
- Asymptotic ratio ~ 8
- b^{global} may not be optimal
- $e^{\perp}_{\infty}(X_{\text{bad}})$ may not be the worst

Numerical tests

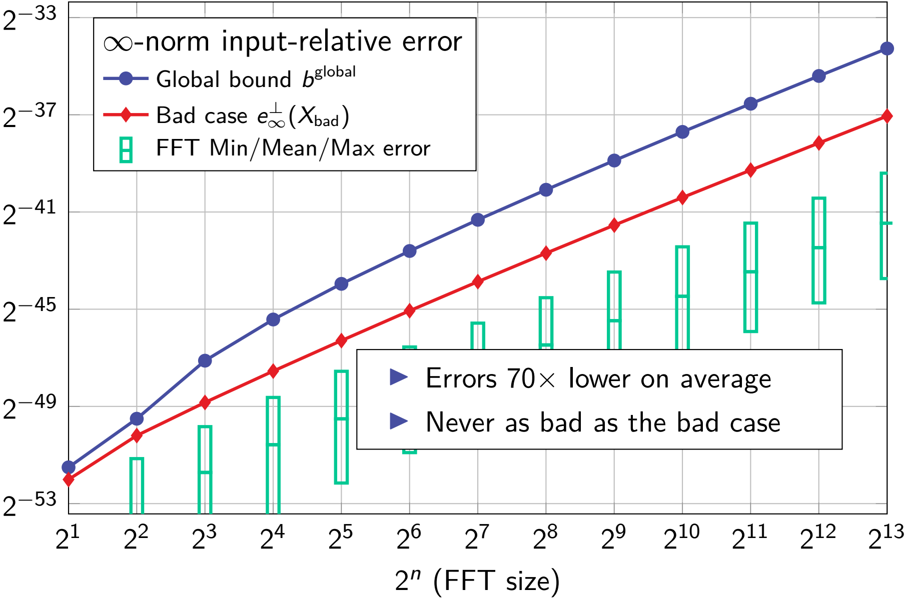
- Cooley & Tukey [1965] algorithm implemented with Julia (v1.7.2)
- FP format: binary64/double-precision arithmetic ($p = 53$)
- Random set of 65 536 input vectors for each n up to 13
- "Exact results" computed with Johansson's Arblib[3]
- For each size, we collect statistics of error values

[3] https://github.com/kalmarek/arblib.jl/tree/v0.8.1

Global error bound: numerical tests



Global error bound: numerical tests



- Errors 70× lower on average
- Never as bad as the bad case

- It seems that numerical errors are much smaller than b^{global} on average
- There is Interval Arithmetic that may produce *tighter local bounds*
- But Interval Arithmetic seems to have large overestimation on FFT
- Let's try it out

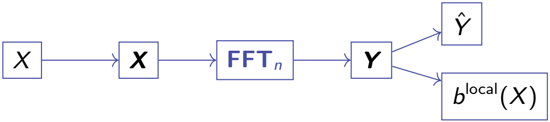
A posteriori or local error bound

Interval Arithmetic

- Each number z is replaced by an interval $\boldsymbol{z}$ of possible values
- Interval operations encloses exact and rounded values

$$x \triangle y \in \boldsymbol{x} \boldsymbol{\Delta} \boldsymbol{y} \qquad\qquad \mathrm{RN}(x \triangle y) \in \boldsymbol{x} \boldsymbol{\Delta} \boldsymbol{y}$$

- In **FFT**, operations are performed with Interval Arithmetic



where we can extract a local bound $b^{\text{local}}(X)$

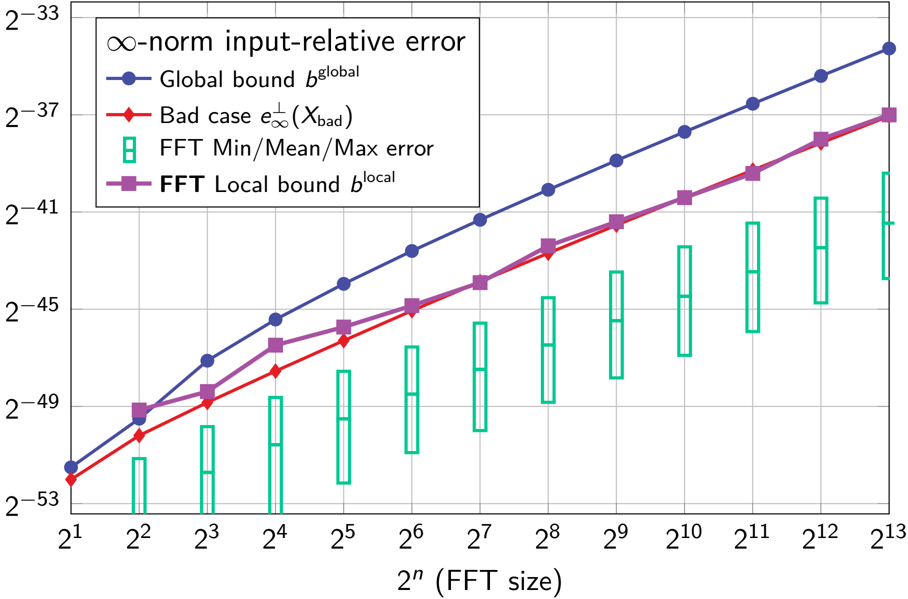
$$e_{\infty}^{\perp}(X) \leqslant b^{\text{local}}(X) = \frac{\max\left\{ \operatorname{diam}(\boldsymbol{u}), \operatorname{diam}(\boldsymbol{v}) \,\middle|\, \boldsymbol{u} + i\,\boldsymbol{v} \in \boldsymbol{Y} \right\}}{\|X\|_{\infty}^{\perp}}$$

Numerical tests

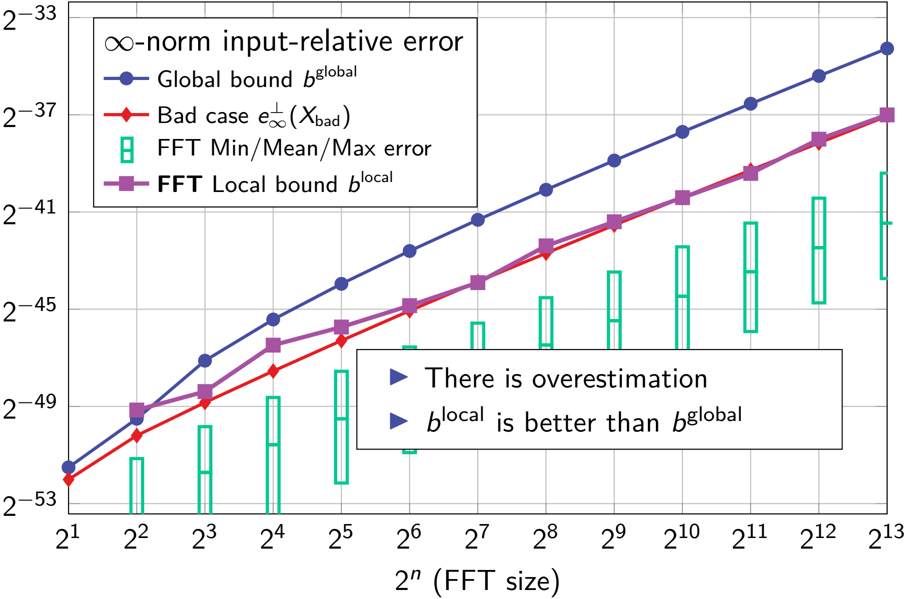
- We compute the local error bound for the same set of input
- We use the IntervalArithmetic library[4]
- Endpoints are encoded with binary64/double-precision arithmetic ($p = 53$)

[4] https://github.com/JuliaIntervals/IntervalArithmetic.jl/tree/v0.20.8

Local error bound: numerical tests

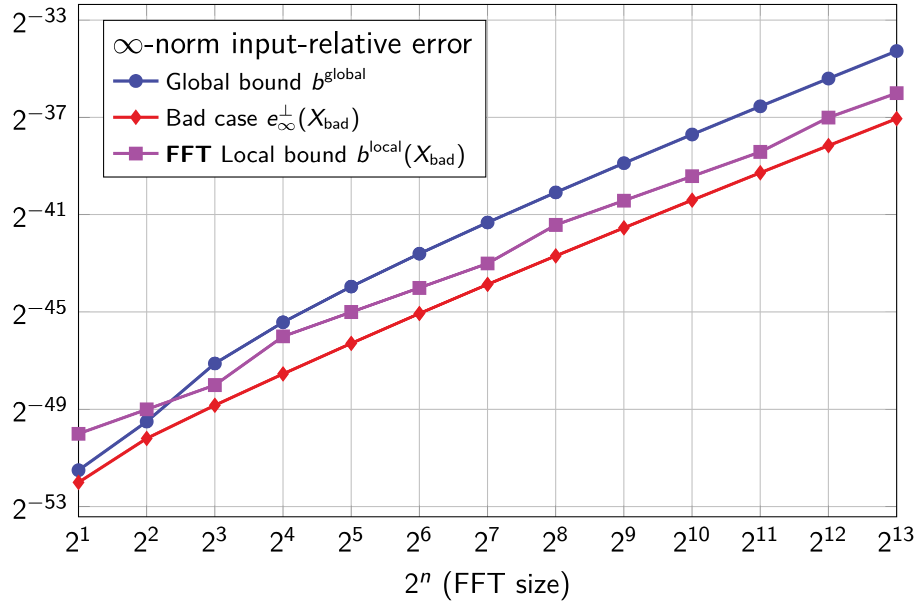


Local error bound: numerical tests

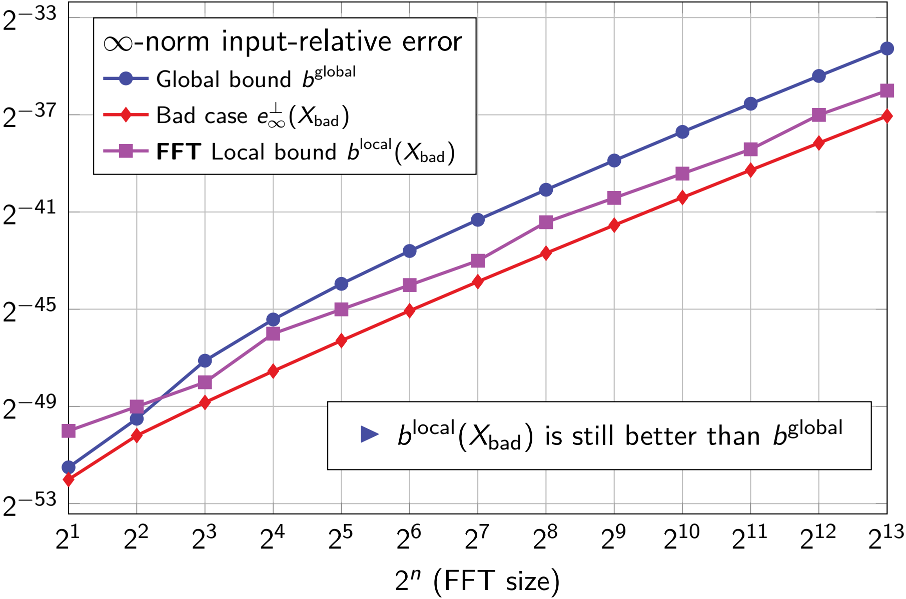


- There is overestimation
- b^{local} is better than b^{global}

Local bound applied to the bad case



Local bound applied to the bad case



▶ $b^{\text{local}}(X_{\text{bad}})$ is still better than b^{global}

Preliminary insights

Global bound approach

- We have a global bound, and a bad case [Brisebarre & al., 2020]
- Actual errors almost always much smaller

Local bound approach

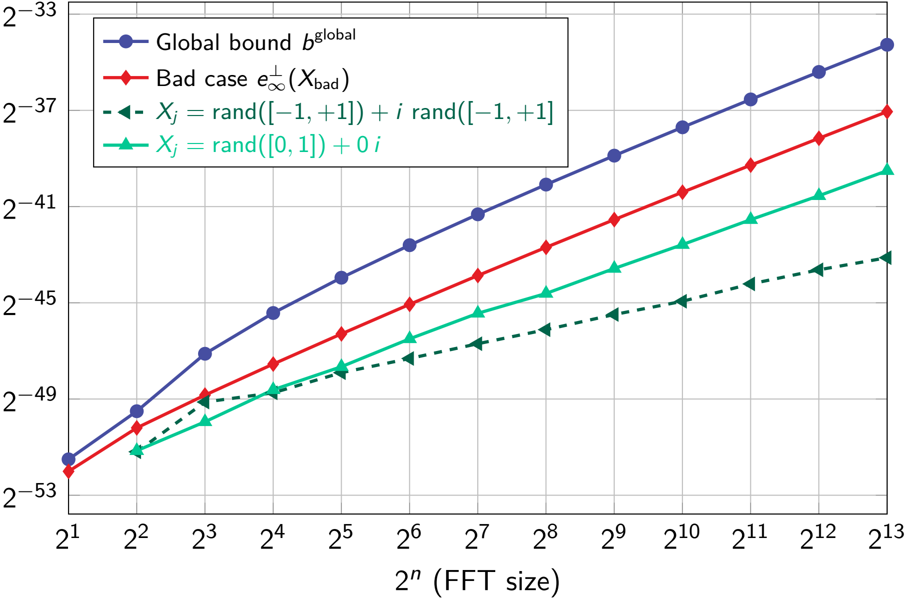
- Interval Arithmetic generate large overestimation on local bounds
- The local bound seems better for the average case

Future work

- Benchmarks should complete the choice between global and local

Thank you!

Generate random input vectors



Floating-point errors in complex numbers

- Complex numbers are implemented with real and imag. FP numbers:

$$x = \mathtt{a} + i\,\mathtt{b} \qquad\qquad y = \mathtt{c} + i\,\mathtt{d}$$

- Complex multiplication: $\hat{z} = x \otimes y = \mathtt{e} + i\,\mathtt{f}$ computed as:

Without FMA*

$$\begin{cases} \mathtt{e} = (\mathtt{a} \otimes \mathtt{c}) \ominus (\mathtt{b} \otimes \mathtt{d}) \\ \mathtt{f} = (\mathtt{a} \otimes \mathtt{b}) \oplus (\mathtt{c} \otimes \mathtt{d}) \end{cases}$$

With FMA*

$$\begin{cases} \mathtt{e} = \mathrm{fma}(\mathtt{a}, \mathtt{c}, -(\mathtt{b} \otimes \mathtt{d})) \\ \mathtt{f} = \mathrm{fma}(\mathtt{a}, \mathtt{b}, (\mathtt{c} \otimes \mathtt{d})) \end{cases}$$

*$\mathrm{fma}(\alpha, \beta, \gamma) = \circ(\alpha \times \beta + \gamma)$

- Error bounds for complex multiplication:

$$|z - \hat{z}| \leqslant u\sqrt{5}\,|z|$$

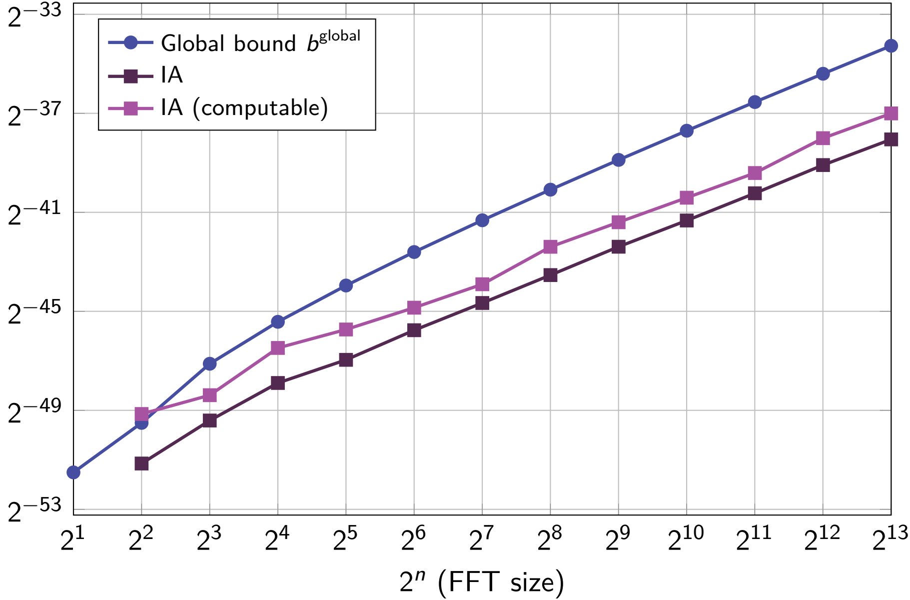
[Percival, 2002]

$$|z - \hat{z}| \leqslant 2u\,|z|$$

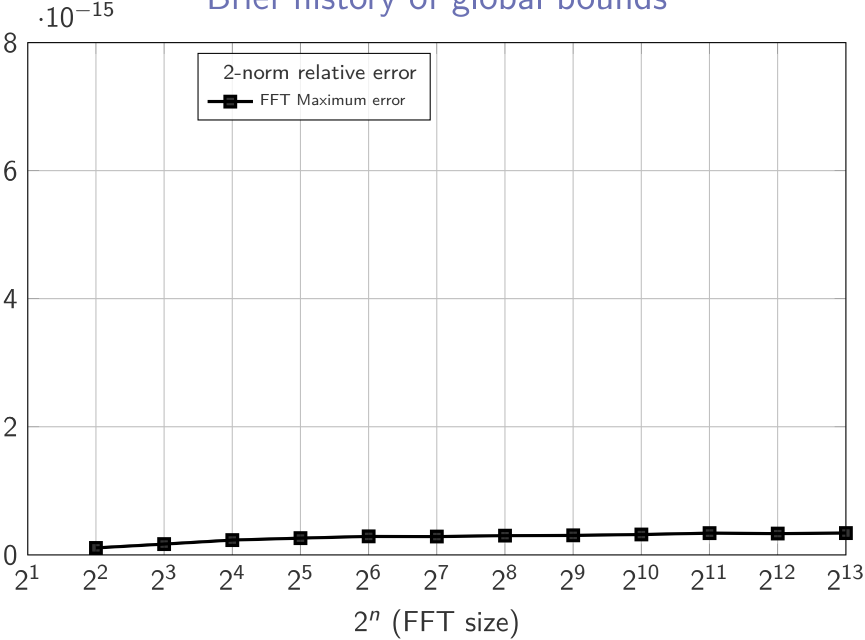
[Jeannerod & al., 2017]

- These bounds are asymptotically optimal.

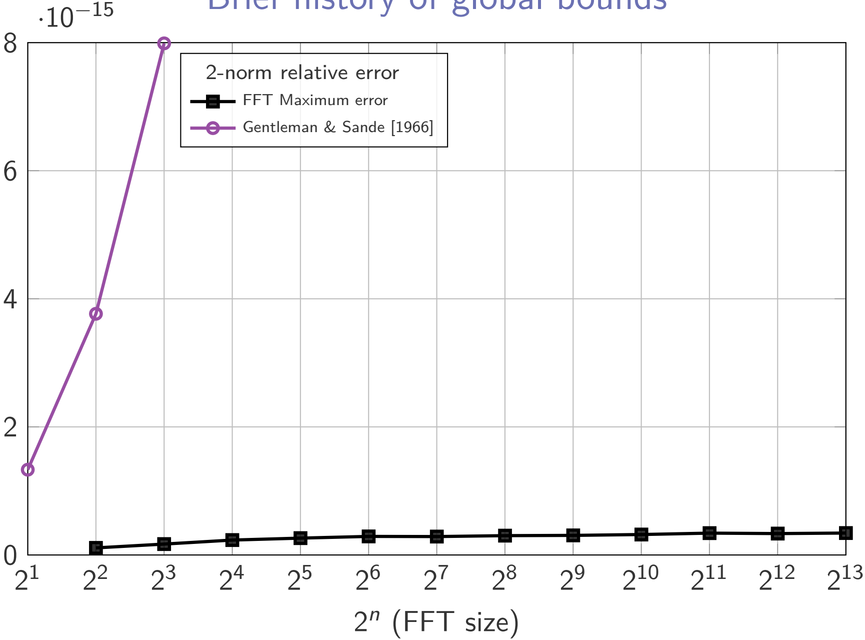
Comparison between error and computable error



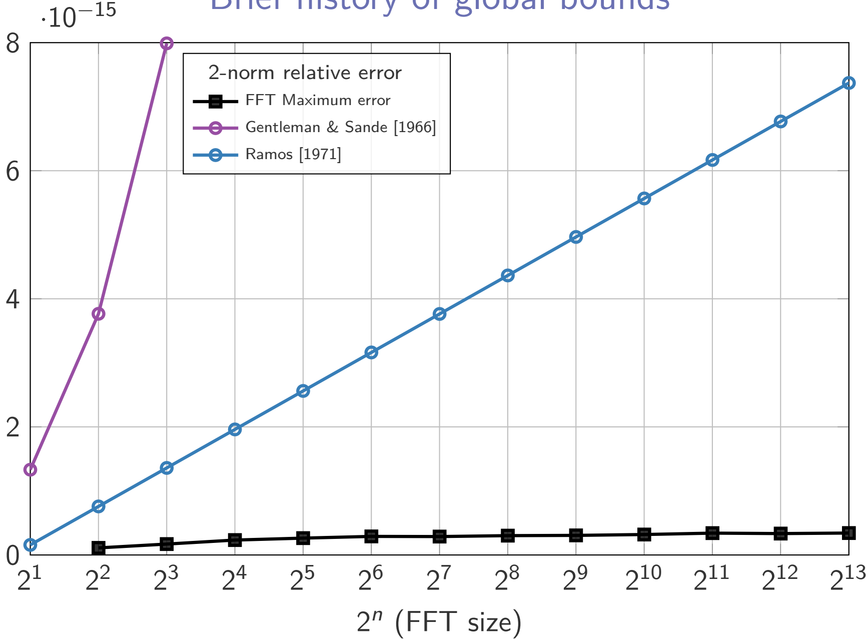
Brief history of global bounds



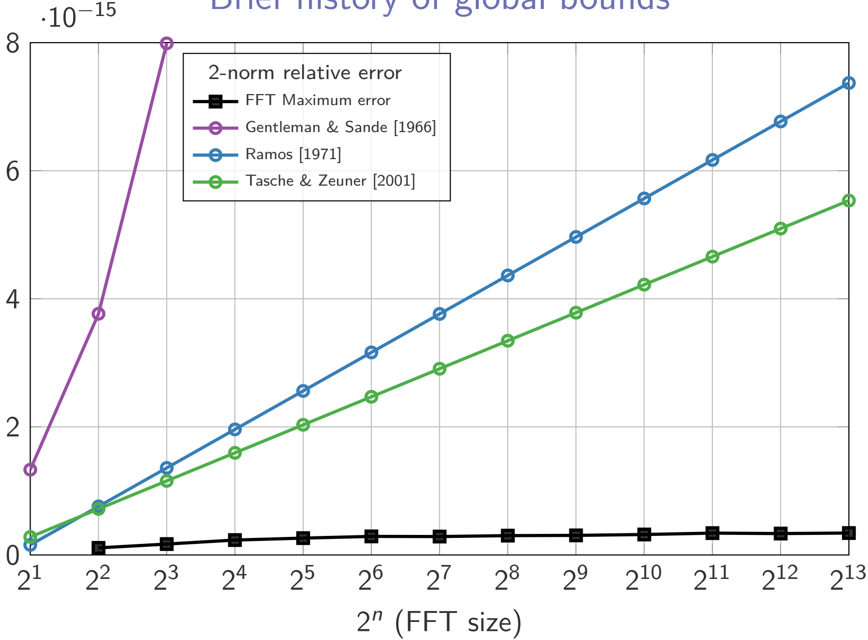
Brief history of global bounds



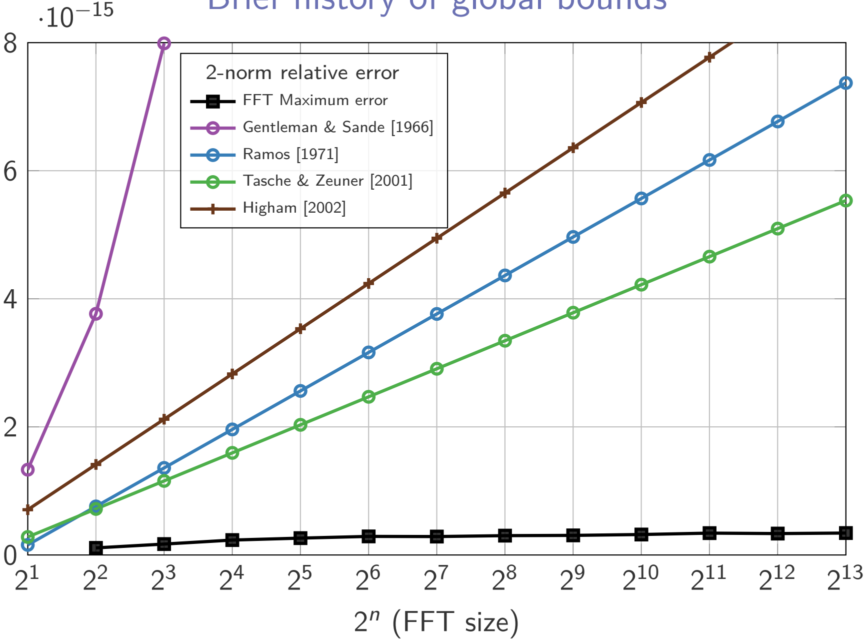
Brief history of global bounds



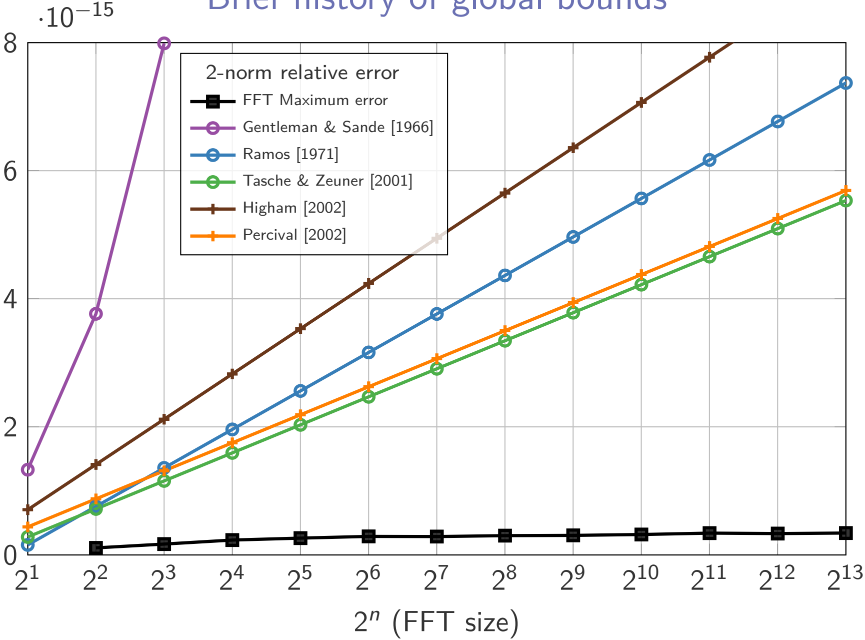
Brief history of global bounds



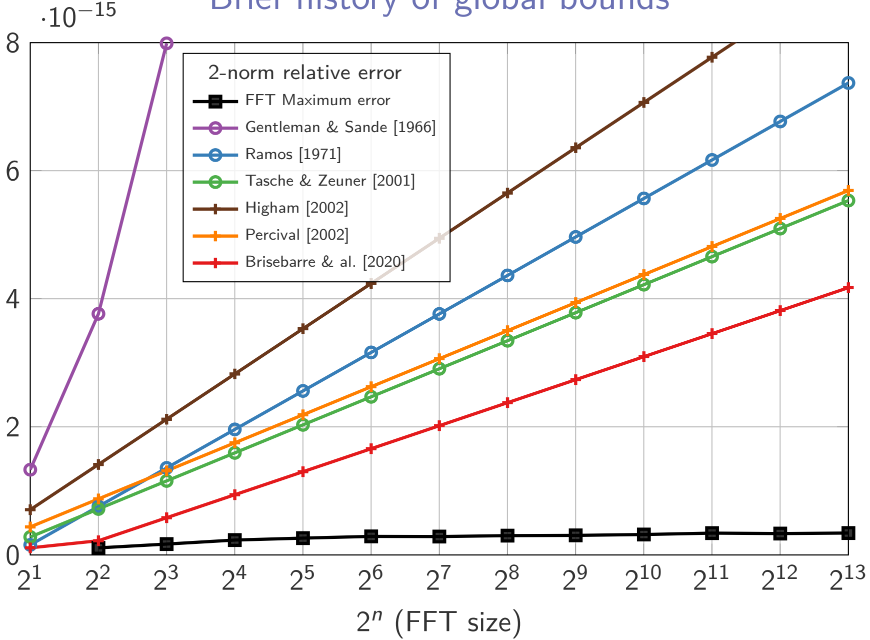
Brief history of global bounds



Brief history of global bounds



Brief history of global bounds



References

N. Brisebarre, M. Joldeș, J.-M. Muller, A.-M. Naneș, J. Picot: "Error Analysis of Some Operations Involved in the Cooley-Tukey Fast Fourier Transform", *ACM Trans. Math. Softw. 46(2)*, 2020.

J. Cooley and J. Tukey: "An Algorithm for the Machine Calculation of Complex Fourier Series", *Math. Comput. 19*, 1965.

W. M. Gentleman, G. Sande: "Fast Fourier Transforms: for fun and profit", *Fall Joint Computer Conference*, 1966.

N. J. Higham: "§ 23 Fast Matrix Multiplication" *Accuracy and stability of numerical algorithms*, 2002.

C.-P. Jeannerod, P. Kornerup, N. Louvet, J.-M. Muller: "Error bounds on complex floating-point multiplication with an FMA", *Math. Comp. 86(304)*, 2017.

G. Liu and V. Kreinovich: "Fast convolution and Fast Fourier Transform under interval and fuzzy uncertainty" *J. Comput. Syst. Sci, 76(1)*, 2010.

C. Percival: "Rapid multiplication modulo the sum and difference of highly composite numbers" *Math. Comp. 72(241)*, 2002.

G. Ramos: "Roundoff Error Analysis of the Fast Fourier Transform" *Math. Comp. 25(116)*, 1971.

B. Salvy and P. Zimmermann: "GFUN: a Maple package for the manipulation of generating and holonomic functions in one variable", *ACM Trans. Math. Softw. 20(2)*, 1994.

M. Tasche and H. Zeuner: "Worst and Average Case Roundoff Error Analysis for FFT", *Bit Numer. Math. 41(3)*, 2001.