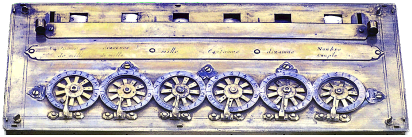
This not (really) a talk
a story of tables and chairs

Jean-Michel Muller

RAIM 2025

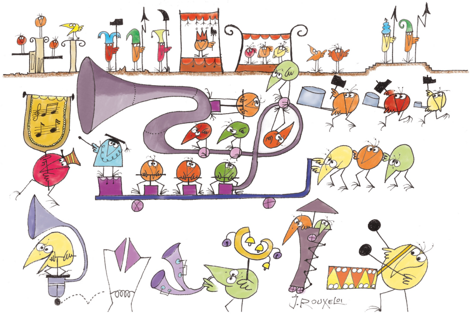


This talk is not (really) a talk… it's a story of tables and chairs

You'll quickly notice it's…
a little unstructured:

- Half French / moitié Anglais;
- not 100 percent serious;
- half scientific / moitié bavardage.

Plan de l'exposé:



Je promets, j'ai rien fumé pour faire ces slides même si parfois on dirait.

In a typical conference banquet, you have certainly observed that

Among the n tables:

- $n - 1$ tables where people talk of science, in English;
- and one table where they tell silly jokes, in Italian, Spanish, or French.

  (sau poate în român)

So I will follow tradition:

- most science in English;
- le reste en Français.

Nicolas asked me to prepare a forward-looking presentation…

To find inspiration…

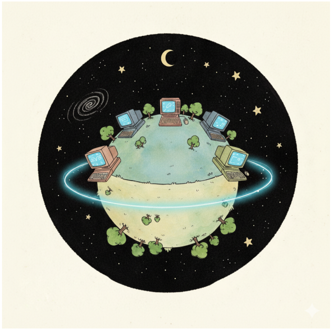
I looked at predictions made by big names…

Thomas J. Watson

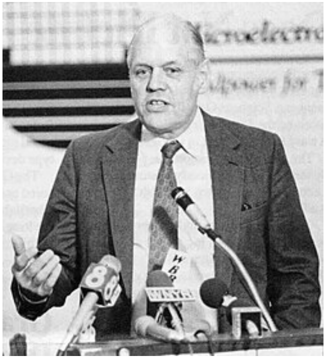
The great chairman of IBM, from 1924 to 1956.

I think there is a world market for maybe 5 computers

Je pense qu'il y a un marché mondial pour peut-être 5 ordinateurs



Ken Olsen (Founder and former chair of DEC)

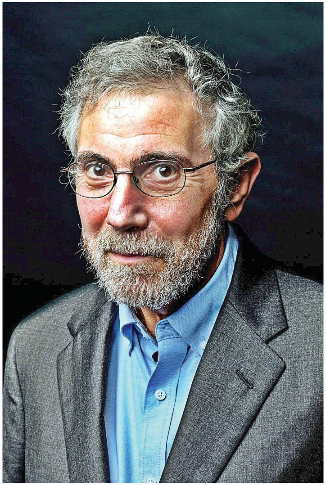


In 1977:

There is no reason for any individual to have a computer in their home

Il n’y a aucune raison pour qu’un particulier possède un ordinateur chez lui.

Paul Krugman (“Nobel prize” in economics in 2008)



In 1998:

By 2005 or so, it will become clear that the Internet’s impact on the economy has been no greater than the fax machine’s

D’ici 2005 environ, il apparaîtra clairement que l’impact d’Internet sur l’économie n’aura pas été plus important que celui du télécopieur

Steve Ballmer (Former CEO of Microsoft)

Talking about the IPhone in 2008:

Let’s look at the facts. Nobody uses those things

Regardons les faits. Personne n’utilise ces choses.

Nicolas Brisebarre

Entering my office, in 2002:

Within two years, the Table Maker's Dilemma will be solved.

Dans 2 ans, le dilemme du fabricant de tables sera résolu.

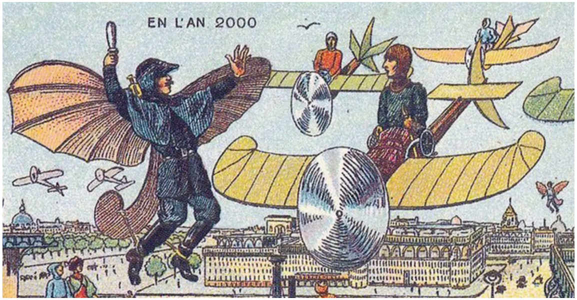
(bon…on a beaucoup avancé mais la solution complète on l'attend toujours)

Conclusion

Predicting the future is easy...

but getting it right is the hard part.

(attributed to Bill Gates)



french postcard, circa 1900

Getting some help from AI?

Computer arithmetic, AI and Surrealism

One of the most useful tools of Floating-Point arithmetic is:

Lemma 1 (Sterbenz Lemma)

Let $a, b \in \mathbb{F}$. *If*

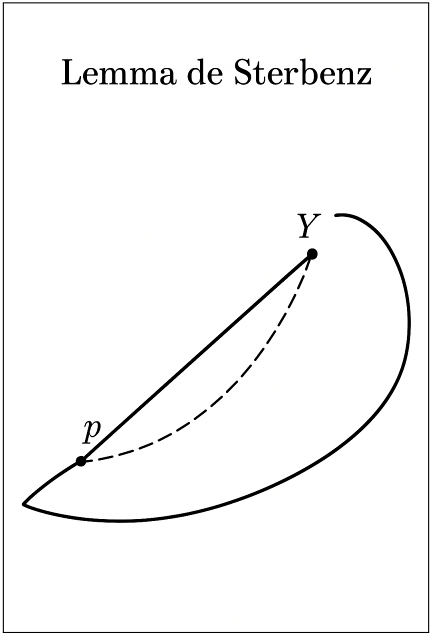
$$\frac{a}{2} \leq b \leq 2a,$$

then $a - b \in \mathbb{F}$.

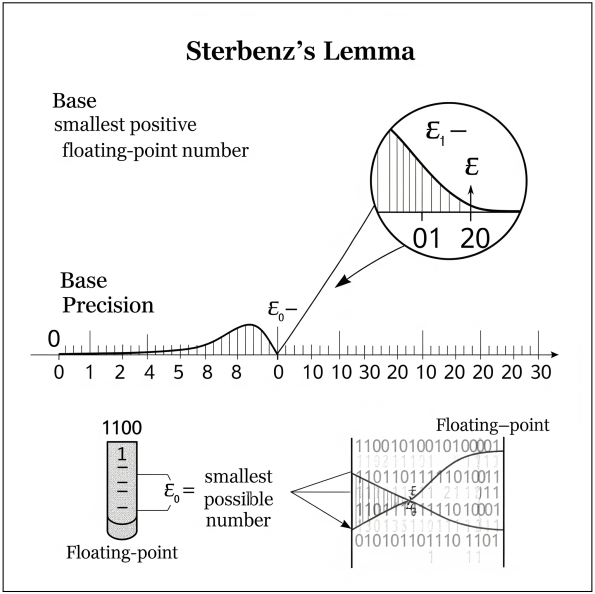
Implies that the subtraction $a - b$ is performed exactly in FP arithmetic.

I have told ChatGPT and Gemini "*Draw a figure that illustrates Sterbenz Lemma*"…

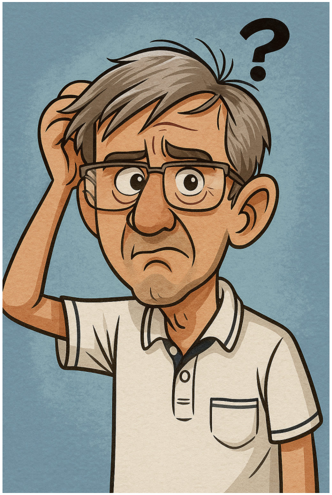
Computer arithmetic, AI and Surrealism: ChatGPT's version



Computer arithmetic, AI and Surrealism: Gemini's version



Conclusion: I'll have to figure it out myself / Il va falloir que je me débrouille.



Let's start with the past!

44 years ago… a chaos of incompatible arithmetics

Machine	Underflow	Overflow
DEC PDP-11, VAX, F and D formats	$2^{-128} \approx 2.9 \times 10^{-39}$	$2^{127} \approx 1.7 \times 10^{38}$
DEC PDP-10; Honeywell 600, 6000; Univac 110x single; IBM 709X, 704X	$2^{-129} \approx 1.5 \times 10^{-39}$	$2^{127} \approx 1.7 \times 10^{38}$
Burroughs 6X00 single	$8^{-51} \approx 8.8 \times 10^{-47}$	$8^{76} \approx 4.3 \times 10^{68}$
H-P 3000	$2^{-256} \approx 8.6 \times 10^{-78}$	$2^{256} \approx 1.2 \times 10^{77}$
IBM 360, 370; Amdahl; DG Eclipse M/600; …	$16^{-65} \approx 5.4 \times 10^{-79}$	$16^{63} \approx 7.2 \times 10^{75}$
Most handheld calculators	10^{-99}	10^{100}
CDC 6X00, 7X00, Cyber	$2^{-976} \approx 1.5 \times 10^{-294}$	$2^{1070} \approx 1.3 \times 10^{322}$
DEC VAX G format	$2^{-1024} \approx 5.6 \times 10^{-309}$	$2^{1023} \approx 9 \times 10^{307}$

Source: W. Kahan, *Why do we need a Floating-Point Standard*, 1981.
A must-read!

44 years ago… a chaos of incompatible arithmetics

- vastly different environments (precision, range, exception handling . . .);
- some good, many quite poor;
- almost no portability of numerical software;
- impossible to prove anything (what can you prove if you don't even know what $c \leftarrow a + b$ gives?);
- need to know "wizard tricks":
  - `(0.5 - x) + 0.5` often better than `(1 - x)`;
  - multiplying by 1 sometimes helps.

Then came IEEE 754-1985

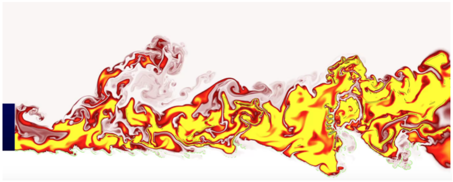
- two formats: single and double precisions (more later);
- correctly-rounded operations:
  - uniform bound on relative error of operations;

  $\rightarrow$ key to clean and rigorous numerical analysis (Wilkinson);
  - fully specified result of the operations;
- well-specified exception handling.

… and just like that, the chaos *almost** became a well-kept garden!

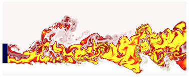
* *almost*: extended formats $\rightarrow$ double roundings, intermediate results stored in 80-bit registers or 64-bit memory, no specification of the math functions…

Was facilitated by the small number of application domains


Numerical simulation

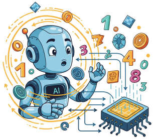
Financial calculations

Now the situation is quite different


Numerical simulation

Financial calculation

Games, entertainment



Machine learning

Embedded computing

• • •

Now the situation is quite different

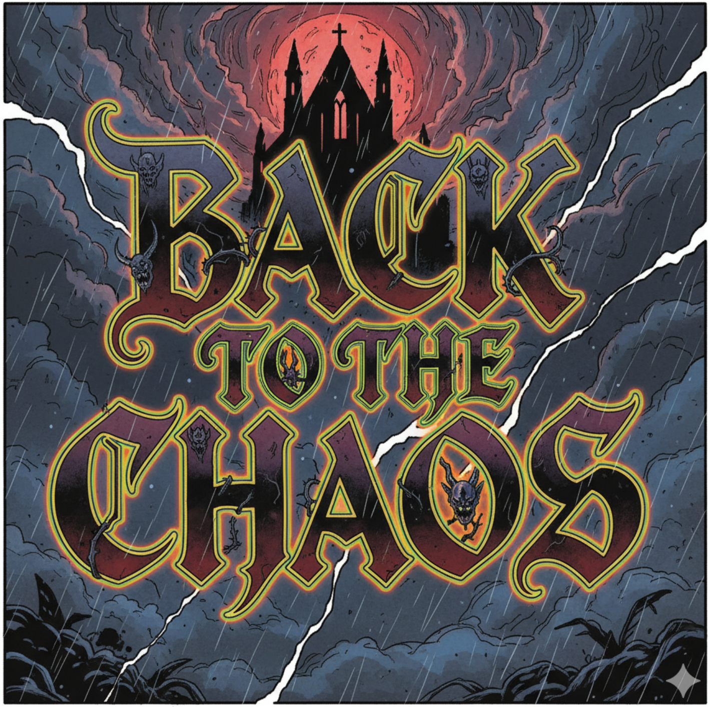
- Extremely diverse requirements in terms of speed, reliability, accuracy, dynamic range, certification, reproducibility, power, data transfer between memory and processor, …
- Another big change in the last 40 years is the considerable increase of the ratio

$$\frac{\text{time to read/write in memory}}{\text{time to perform } +, \times, \div, \sqrt{}}$$

$\rightarrow$ small formats, and strong incentive to use them even for applications that would need much accuracy and/or wide range (mixed precision in numerical analysis)

- heterogeneity of computing platforms: CPUs, GPUs, Tensor cores, etc.

The consequence of all this is…


BACK
TO THE
CHAOS

Welcome back to the chaos?

Format	Precision	smallest normal (black) or smallest nonzero (orange)	max
binary8p3	3	3.052×10^{-5}	49152
binary8p4	4	7.812×10^{-3}	224
OCP MX FP8E5M2	3	6.104×10^{-5}	57344
OCP MX FP8E4M3	4	0.0156	448
Posit8	1–4	5.96×10^{-8}	1.677×10^{7}
binary16	11	6.1035×10^{-5}	65504
BFloat16	8	1.175×10^{-38}	3.389×10^{38}
Posit16	1–12	1.387×10^{-17}	7.205×10^{16}
binary32	24	1.175×10^{-38}	3.402×10^{38}
Posit32	1–28	7.723×10^{-37}	1.329×10^{36}
TensorFloat32	11	1.175×10^{-38}	3.402×10^{38}

Plus many ugly things: shared exponents, ad-hoc emulations (without decent exception handling), underspecified operations (reverse engineering needed), number systems without uniform bounds on errors of operations (posits)...

What some seem to think at Nvidia and similar companies

If I remove the seat belts, airbags and brakes, my car will be lighter and therefore faster!

Si j’enlève les ceintures de sécurité, les airbags et les freins, ma voiture sera plus légère et donc ira plus vite.

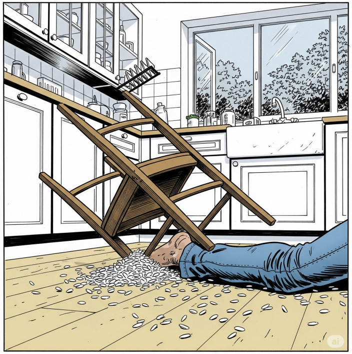
What can we do to avoid complete chaos?

Specify, specify and specify:

- consistent exception handling;
- clear, simple and complete specification of arithmetic operations, conversions, math functions, etc.
- help interoperability;
- don't have only one application in mind (*design thinking*)
  (usual excuse: "for my application I don't need that feature")

A lesson from Design thinking

$\Rightarrow$



- those who design chairs know that they will also be used as stepladders;
- warn these "careless" users, call them idiots?... like it or not, they are there anyway;
- it's best to take this use into account when designing a chair.

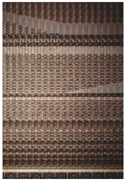
What does it mean for us?

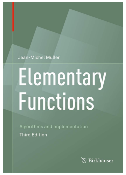
There is no such thing as…

- an arithmetic dedicated to one application: if your piece of hardware/software is fast, it will be used elsewhere: success of mixed-precision computing;
- a purely storage format: some will do arithmetic with it;
- an extended format that only serves for implementing functions for the corresponding basic format.

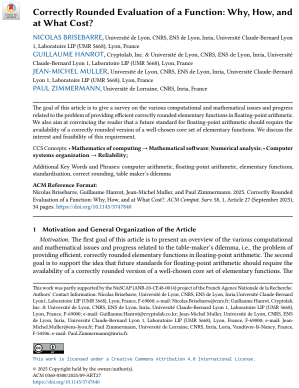
Remember: Some applications need uniform relative error bound (error control in numerical analysis), possibility of formal proof (avionics, automated transportation) and therefore full specification.

42 years of love with the elementary functions





Felin (1985) Editions: 1997-2006-2016



Correctly Rounded Evaluation of a Function: Why, How, and at What Cost?

NICOLAS BRISEBARRE, Université de Lyon, CNRS, ENS de Lyon, Inria, Université Claude-Bernard Lyon 1, Laboratoire LIP (UMR 5668), Lyon, France
GUILLAUME HANROT, Cryptolab, Inc. & Université de Lyon, CNRS, ENS de Lyon, Inria, Université Claude-Bernard Lyon 1, Laboratoire LIP (UMR 5668), Lyon, France
JEAN-MICHEL MULLER, Université de Lyon, CNRS, ENS de Lyon, Inria, Université Claude-Bernard Lyon 1, Laboratoire LIP (UMR 5668), Lyon, France
PAUL ZIMMERMANN, Université de Lorraine, CNRS, Inria, France

The goal of this article is to give a survey on the various computational and mathematical issues and progress related to the problem of providing efficient correctly rounded elementary functions in floating-point arithmetic. We also aim at convincing the reader that a future standard for floating-point arithmetic should require the availability of a correctly rounded version of a well-chosen core set of elementary functions. We discuss the interest and feasibility of this requirement.

CCS Concepts: • **Mathematics of computing → Mathematical software**; **Numerical analysis**; • **Computer systems organization → Reliability;**

Additional Key Words and Phrases: computer arithmetic, floating-point arithmetic, elementary functions, standardization, correct rounding, table maker's dilemma

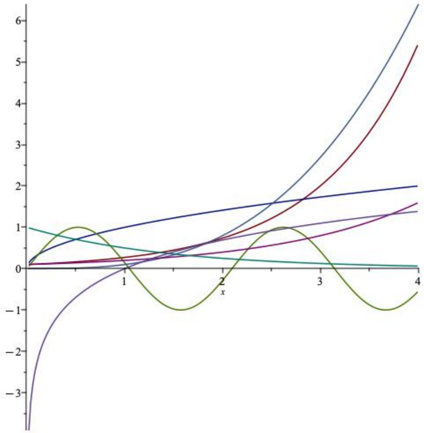
ACM Reference Format:
Nicolas Brisebarre, Guillaume Hanrot, Jean-Michel Muller, and Paul Zimmermann. 2025. Correctly Rounded Evaluation of a Function: Why, How, and at What Cost?. *ACM Comput. Surv.* 58, 1, Article 27 (September 2025), 34 pages. https://doi.org/10.1145/3747840

1 Motivation and General Organization of the Article

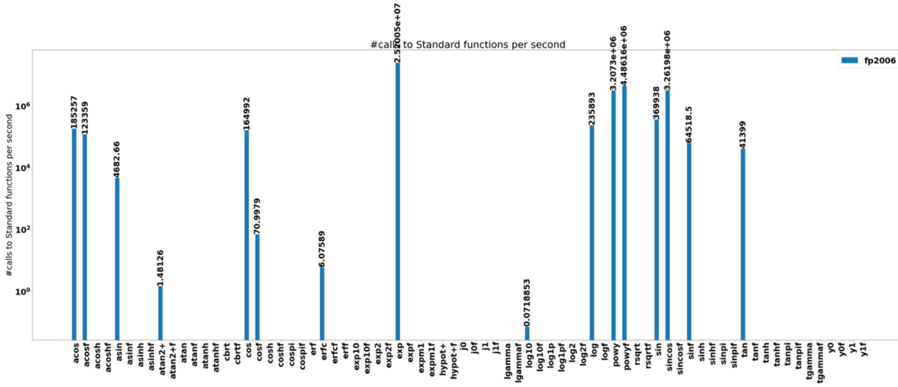
Motivation. The first goal of this article is to present an overview of the various computational and mathematical issues and progress related to the table-maker's dilemma, i.e., the problem of providing efficient, correctly rounded elementary functions in floating-point arithmetic. The second goal is to support the idea that future standards for floating-point arithmetic should require the availability of a correctly rounded version of a well-chosen core set of elementary functions. The

This work was partly supported by the NuSCAP (ANR-20-CE48-0014) project of the French Agence Nationale de la Recherche.
Authors' Contact Information: Nicolas Brisebarre, Université de Lyon, CNRS, ENS de Lyon, Inria,Université Claude-Bernard Lyon1, Laboratoire LIP (UMR 5668), Lyon, France, F-69000; e-mail: Nicolas.Brisebarre@cnrs.fr; Guillaume Hanrot, Cryptolab, Inc. & Université de Lyon, CNRS, ENS de Lyon, Inria, Université Claude-Bernard Lyon 1, Laboratoire LIP (UMR 5668), Lyon, France, F-69000; e-mail: Guillaume.Hanrot@cryptolab.co.kr; Jean-Michel Muller, Université de Lyon, CNRS, ENS de Lyon, Inria, Université Claude-Bernard Lyon 1, Laboratoire LIP (UMR 5668), Lyon, France, F-69000; e-mail: Jean-Michel.Muller@ens-lyon.fr; Paul Zimmermann, Université de Lorraine, CNRS, Inria, Loria, Vandŭvre-lš-Nancy, France, F-54506; e-mail: Paul.Zimmermann@inria.fr.

ACM 0360-0300/2025/09-ART27
https://doi.org/10.1145/3747840



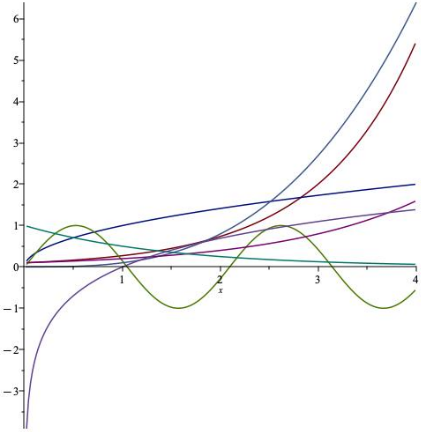
42 years of love with the elementary functions



Number of calls per second of various functions in a CERN proton collision application (V. Innocente).

42 years of love with the elementary functions

- Core set of “atomic functions”: exp, log, sin, arctan…
    - by far the most frequently called;
    - building blocks for more complex functions.
- highest possible quality: reproducibility, proven bounds, etc.
- best possible quality: correct rounding;
- Peter Tang, John Harrison.



Ultimate quality is tricky…

No room for loose estimates. When you approximate f by a polynomial:

- reliable ∞-norm to get a certain and tight bound on approximation error;
- certain and tight bound on the polynomial evaluation error;
- double-word or triple-word arithmetic for very last steps of Horner/Estrin evaluation;
- best (or nearly best) approximations with constrained coefficients (such as being FP numbers);
- final rounding test;
- accuracy of intermediate steps: table maker's dilemma. (G. Hanrot's talk);

It has been a 30-year program

Nicolas Brisebarre, Sylvain Chevillard, David Defour, Florent de Dinechin, Nicolas Fabiano, Silviu Filip, Guillaume Hanrot, John Harrison, Mioara Joldes, Christoph Lauter, Vincent Lefèvre, Erik Martin-Dorel, Guillaume Melquiond, Valentina Popescu, Olivier Robert, Laurence Rideau, Laurent Théry, Damien Stehlé, Arnaud Tisserand, Serge Torres, Paul Zimmermann, …

Libraries of math functions: big improvements

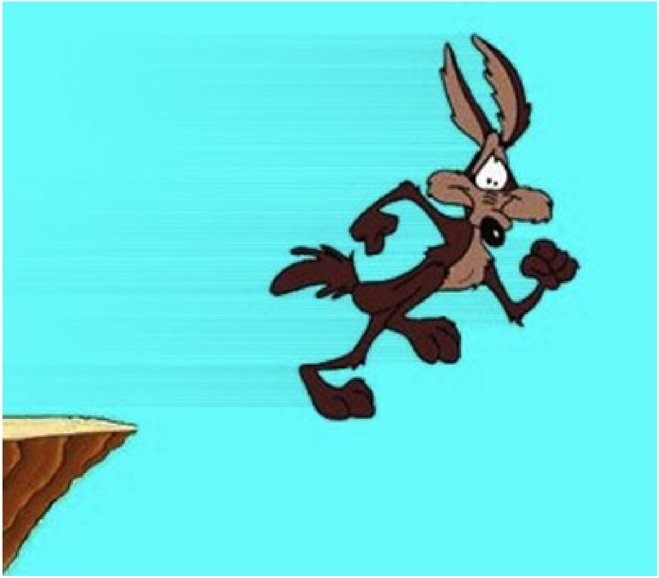
- CRLIBM: Proof of concept. Wonderful work by Florent, David and Christoph;
- success of CORE-MATH, led by Paul Zimmermann;
- starting to be at least partly adopted by major players (Meta, GNU libc, AMD, European Space Agency...)

`https://core-math.gitlabpages.inria.fr`

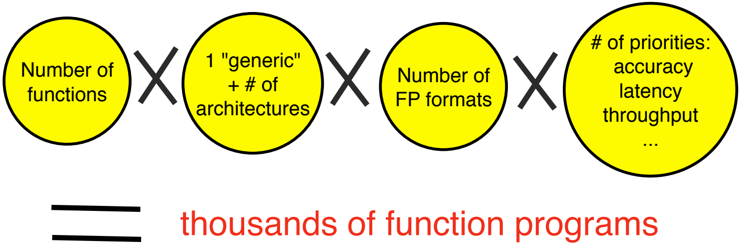
CORE-MATH is a remarkable achievement, Bravo Paul!

But...

Heading straight for disaster / On va droit dans le mur



Libraries of math functions: heading straight for disaster


Number of functions
X
1 "generic" + # of architectures
X
Number of FP formats
X
# of priorities: accuracy latency throughput ...
=
thousands of function programs

- almost impossible to debug, maintain, keep consistent, improve… by a small team;
- and physicists, chemists, statisticians, … would like many other functions

First solution: computer-assisted library design

Prototype: Metalibm project, launched by Florent de Dinechin.
Existed in two versions:

- fully automated for the end user;
- assistance for the specialist.

`http://www.metalibm.org`

Maybe this is the ultimate goal

Maybe not

Second solution: Generation of functions at compile-time

- take into account the exact context: underlying architecture, accuracy requirements, priorities (latency/throughput/code size...);
- possibly, information on input domain ($\rightarrow$ simplify/avoid range reduction), or special cases (e.g., infinities, NaNs known not to happen);
- compound functions: if you need

  $$E_4(x) = \frac{x}{e^x - 1} - \ln(1 - e^{-x}),$$

  then you directly generate $E_4(x)$ instead of generating exp, ln and combining them.
- formal proof absolutely necessary (no library to heavily test beforehand);
- interactions: computer arithmetic, number theory, computer algebra, compilation, formal proof...