The Table Maker's Dilemma: Old Stories and New Results

Guillaume Hanrot

Nov. 6, 2025

To the memory of Serge Torres.

Once upon a time

THE TABLE-MAKERS' DILEMMA

W. Kahan

Computer Science Department

University of California at Berkeley

August 1971

Once upon a time (II)

"It is confidently believed that the cases where the error exceeds ±0.51 units of the last decimal could be counted on the fingers of one hand; those that are known to exist form an uncomfortable trap for any would-be plagiarist."

Chambers's Shorter Six-Figure Mathematical Tables (1959)
L.J. Comrie

Once upon a time (III)

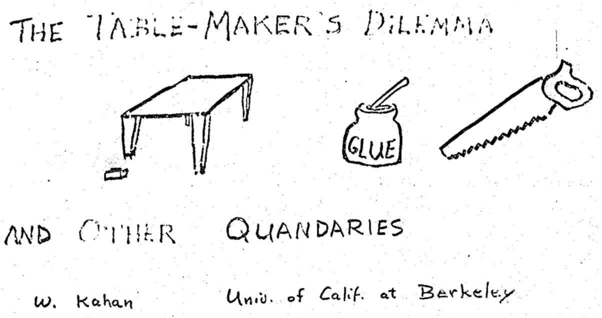
$$1/3^3 + 1/5^3 + 1/7^3 + \cdots + 1/(2n+1)^3 + \cdots = (7/8)\zeta(3) - 1$$

where $\zeta(s) = \Sigma_1^\infty n^{-s}$ is Riemann's Zeta-function. Working to 15 significant decimals yields a value

$$0.0\ 51799\ 79026\ 464\ 50\ldots,$$

uncertain by 1 in the last decimal cited. Should the 13^{th} significant figure be rounded up or down? Rather than guess we recompute working to

Once upon a time (IV)



Once upon a time (V)

than will the confusion and recriminations it saves. While programming elementary functions for the IBM 7094-II at the University of Toronto between 1962 and 1965 (see my 1968 notes) I found that extra care cost less than one month's extra work per program, and slowed the program by less than 10% if at all. Moreover, much of the extra work was devoted to proving, mathematically first and then by running tests on data, that the program performed as well as I claimed. The kinds of claims I made are

Why the table maker's dilemma?

- Correct rounding: IEEE-754 core philosophy;
- $+$, $\times$, $-$, $/$, $\sqrt{\ }$;

Why the table maker's dilemma?

- Correct rounding: IEEE-754 core philosophy;
- $+$, $\times$, $-$, $/$, $\sqrt{\ }$;
- Elementary functions: too hard;
- Cody (1980):

  Software for the elementary functions normally resides in system libraries accompanying compilers for high level languages. Unless there is strong evidence of poor performance, users tend to regard these programs in the same way they regard the arithmetic operations in the computer. That is, they view them as friendly 'black boxes' that can be trusted to be efficient and accurate. Only careful preparation of software guarantees that the trust will not be violated.

HEAAN CRYPTO LAB

Correct rounding for elementary functions

[Advertisement / Publi-communiqué]

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

Authors: Nicolas Brisebarre, PhD, Guillaume Hanrot, Jean-Michel Muller, Paul Zimmermann | Aut & Claims

ACM Computing Surveys, Volume 58, Issue 1 • Article No.: 27, Pages 1 - 34 • https://doi.org/10.1145/3747840

Published: 04 September 2025 Publication History

Abstract

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.

Ziv's meta-algorithm

Input : f, x, rounding function rnd, precision p;

$p_0 \leftarrow p + \delta$;

While(1)

Compute $I = [y_0, y_1]$ with $|y_1 - y_0| < 2^{-p_0}$ and $f(x) \in I$;

If $(\mathrm{rnd}_p(y_0) == \mathrm{rnd}_p(y_1))$ {

return $\mathrm{rnd}(y_0)$

}

Increase p_0.

Does it terminate? How much does it cost?

Ziv's meta-algorithm (II)

- This will work unless:

$$f(x) \neq 1.\underbrace{.....}_{p-1}\underbrace{00\ldots00}_{p_0-p}2^{e_{f(x)}}$$

$$f(x) \neq 1.\underbrace{.....}_{p-1}\underbrace{11\ldots11}_{p_0-p}2^{e_{f(x)}}$$

- for fixed p, f, $x \in X$: find the largest such $p_0 =: \mu_{p,f}(x)$.

HEAAN CRYPTO LAB

Ziv's meta-algorithm (II)

- This will work unless:

$$f(x) \neq 1.\underbrace{.....}_{p-1}\underbrace{00\ldots00}_{p_0-p}2^{e_{f(x)}}$$

$$f(x) \neq 1.\underbrace{.....}_{p-1}\underbrace{11\ldots11}_{p_0-p}2^{e_{f(x)}}$$

- for fixed p, f, $x \in X$: find the largest such $p_0 =: \mu_{p,f}(x)$.
- Termination: $p_0 < \infty$;
- For exp family $\Leftarrow$ Hermite-Lindemann's theorem.

HEAAN CRYPTO LAB

Back to history: dark ages

- Few traces in the 80s;
- Probabilistic approaches (next slide) beginning of 90s;
- Rebirth end-90s (Muller, Lefèvre, Tisserand);

Back to history: dark ages

- Few traces in the 80s;
- Probabilistic approaches (next slide) beginning of 90s;
- Rebirth end-90s (Muller, Lefèvre, Tisserand);
- ... probably with CRlibm in mind.

Revival: Lefèvre-Muller-Tisserand

- First papers with a clear formulation as a diophantine problem;
- Clean probabilistic study;
- First (non-trivial) algorithmic ingredients + results.

Back to history: probabilistic approaches

- Simple model: (Dunham, Gal-Bachelis, Muller-Tisserand – 90s).

$$2^p f(x) - \lfloor 2^p f(x) \rceil = 0.B_1 B_2 B_3 ... B_k.$$

HEAAN CRYPTO LAB

Back to history: probabilistic approaches

- Simple model: (Dunham, Gal-Bachelis, Muller-Tisserand – 90s).
$$2^p f(x) - \lfloor 2^p f(x) \rceil = 0.B_1 B_2 B_3 \ldots B_k.$$

- Prob $(\mu_p(f,x) \geq k) = 2^{-k}$;

Back to history: probabilistic approaches

- Simple model: (Dunham, Gal-Bachelis, Muller-Tisserand – 90s).
$$2^p f(x) - \lfloor 2^p f(x) \rceil = 0.B_1 B_2 B_3 ... B_k.$$

- Prob $(\mu_p(f, x) \geq k) = 2^{-k}$;
- $\mathbb{E}[\#\{x \in X / \mu(f, x) \geq j\}] = 2^{-j} \# X$

Back to history: probabilistic approaches

- Simple model: (Dunham, Gal-Bachelis, Muller-Tisserand – 90s).
$$2^p f(x) - \lfloor 2^p f(x) \rceil = 0.B_1 B_2 B_3 ... B_k.$$

- Prob $(\mu_p(f, x) \geq k) = 2^{-k}$;
- $\mathbb{E}[\#\{x \in X / \mu(f, x) \geq j\}] = 2^{-j} \# X$
- Expect largest $j \approx \log_2 \# X$.

Back to history: probabilistic approaches

- Simple model: (Dunham, Gal-Bachelis, Muller-Tisserand – 90s).
$$2^p f(x) - \lfloor 2^p f(x) \rceil = 0.B_1 B_2 B_3 \dots B_k.$$
- Prob $(\mu_p(f, x) \geq k) = 2^{-k}$;
- $\mathbb{E}[\#\{x \in X / \mu(f, x) \geq j\}] = 2^{-j} \# X$
- Expect largest $j \approx \log_2 \# X$.
- X = binade: $\# X = 2^p$, so $\max_{x \in X} \mu_p(f, x) \approx p$;

Back to history: probabilistic approaches

- Simple model: (Dunham, Gal-Bachelis, Muller-Tisserand – 90s).
$$2^p f(x) - \lfloor 2^p f(x) \rceil = 0.B_1 B_2 B_3 \ldots B_k.$$

- Prob $(\mu_p(f, x) \geq k) = 2^{-k}$;
- $\mathbb{E}[\#\{x \in X / \mu(f, x) \geq j\}] = 2^{-j} \# X$
- Expect largest $j \approx \log_2 \# X$.
- $X =$ binade: $\# X = 2^p$, so $\max_{x \in X} \mu_p(f, x) \approx p$;
- for ℓ binades, $\max_x \mu_p(f, x) \approx p + \log \ell$.

From runs of zeros to a diophantine view

$$f(x) \neq 1.\underbrace{.....}_{p}\underbrace{00\ldots00}_{p-p_0+1}2^{e_{f(x)}}$$

$$f(x) \neq 1.\underbrace{.....}_{p}\underbrace{11\ldots11}_{p-p_0+1}2^{e_{f(x)}}$$

- Translates as $|2^{e(f(x))+p}f(x) - y| < 2^{-p_0}$, for y integer;

From runs of zeros to a diophantine view

$$f(x) \neq 1.\underbrace{.....}_{p}\underbrace{00\ldots00}_{p-p_0+1}2^{e_{f(x)}}$$

$$f(x) \neq 1.\underbrace{.....}_{p}\underbrace{11\ldots11}_{p-p_0+1}2^{e_{f(x)}}$$

- Translates as $|2^{e(f(x))+p}f(x) - y| < 2^{-p_0}$, for y integer;
- Theory says everything in the algebraic case (think of $\sqrt{x}, 1/x, x^{1/3}$, etc.)...

From runs of zeros to a diophantine view

$$f(x) \neq 1.\underbrace{.....}_{p}\underbrace{00\ldots00}_{p-p_0+1}2^{e_{f(x)}}$$

$$f(x) \neq 1.\underbrace{.....}_{p}\underbrace{11\ldots11}_{p-p_0+1}2^{e_{f(x)}}$$

- Translates as $|2^{e(f(x))+p}f(x) - y| < 2^{-p_0}$, for y integer;
- Theory says everything in the algebraic case (think of $\sqrt{x}, 1/x, x^{1/3}$, etc.)...*qualitatively*.

HEAAN CRYPTO LAB

From runs of zeros to a diophantine view

- Opens the way to Lang-Muller (2001), Brisebarre-Muller (2007).
- Exploring Liouville-type methods in TMD language.
- Key ingredient: if $n/2^p = y \approx f(x)$ and $P(f(x)) = 0$, $\deg P = d$,

$$|P'(f(x))||y - f(x)| \approx |P(y) - P(f(x))| = A/2^{pd} > 1/2^{pd}.$$

- Typically gives bounds on $|y - f(x)|$ of the order of dp.

From runs and zeros to a diophantine view (II)

- Nesterenko-Waldschmidt: lower bounds on $|\exp(\beta) - \alpha|$;
- Gives TMD-type bounds for $\exp, \log, \cos, \sin$, etc;

From runs and zeros to a diophantine view (II)

- Nesterenko-Waldschmidt: lower bounds on $|\exp(\beta) - \alpha|$;
- Gives TMD-type bounds for $\exp, \log, \cos, \sin$, etc;
- But really bad ones (10^6 bits);
- Slight improvement [BHMZ25]: use Khémira-Voutier.

Don't forget the geometric view!

- Function f / curve $C : y = f(x)$
- Exact cases: points (x, y) with fp coordinates *on* C

Don't forget the geometric view!

- Function f / curve $C : y = f(x)$
- Exact cases: points (x, y) with fp coordinates *on* C
- Bad/worst cases: points (x, y) with fp coordinates *close to* C;

Don't forget the geometric view!

- Function f / curve $C : y = f(x)$
- Exact cases: points (x, y) with fp coordinates *on* C
- Bad/worst cases: points (x, y) with fp coordinates *close to* C;
- analytic number theory techniques $\Rightarrow$ rigorous version of the probabilistic heuristic (Brisebarre, Robert, H.).

Don't forget the geometric view!

- Function f / curve $C : y = f(x)$
- Exact cases: points (x, y) with fp coordinates *on* C
- Bad/worst cases: points (x, y) with fp coordinates *close to* C;
- analytic number theory techniques $\Rightarrow$ rigorous version of the probabilistic heuristic (Brisebarre, Robert, H.).

HEAAN CRYPTO LAB

Algorithmic approaches – Lefèvre-Muller-Tisserand

- Let us take $f(x) = \alpha x - \beta$.
- We want to find small values of $f(x) + y$, x, y integers;
- Cf. Jean-Claude's talk: Ostrowski's basis:
- $\alpha = p_n/q_n$, $\theta_i = q_i\alpha - p_i$;

Algorithmic approaches – Lefèvre-Muller-Tisserand

- Let us take $f(x) = \alpha x - \beta$.
- We want to find small values of $f(x) + y$, x, y integers;
- Cf. Jean-Claude's talk: Ostrowski's basis:
- $\alpha = p_n/q_n$, $\theta_i = q_i\alpha - p_i$;
- $x = \sum x_i q_i$, $\beta = \sum b_i\theta_i$, x_i, b_i integers;

Algorithmic approaches – Lefèvre-Muller-Tisserand

- Let us take $f(x) = \alpha x - \beta$.
- We want to find small values of $f(x) + y$, x, y integers;
- Cf. Jean-Claude's talk: Ostrowski's basis:
- $\alpha = p_n/q_n$, $\theta_i = q_i\alpha - p_i$;
- $x = \sum x_i q_i$, $\beta = \sum b_i\theta_i$, x_i, b_i integers;
- $$\alpha x - \beta = \sum (x_i - b_i)\theta_i + \underbrace{\sum x_i p_i}_{y}$$

Algorithmic approaches – Lefèvre-Muller-Tisserand

- Let us take $f(x) = \alpha x - \beta$.
- We want to find small values of $f(x) + y$, x, y integers;
- Cf. Jean-Claude's talk: Ostrowski's basis:
- $\alpha = p_n/q_n$, $\theta_i = q_i\alpha - p_i$;
- $x = \sum x_i q_i$, $\beta = \sum b_i\theta_i$, x_i, b_i integers;
-
$$\alpha x - \beta = \sum (x_i - b_i)\theta_i + \underbrace{\sum x_i p_i}_{y}$$
- Small values: take $x_i = b_i$ for $i < i_0$.

Lefèvre-Muller-Tisserand's algorithm

Vincent's thesis (2000):

E	mantisse	R	m
0	1.1011010011101011111001000000110010010101101000000001	N	**107**
1	1.0001101111010001110011111111100101000111000111110101 0	D	**106**
2	1.00011011110100011100111111111001010001110001111101010	D	**107**
2	1.10001001110110010100100010101001010011111111000010111	N	**106**
4	1.00011011110100011100111111111001010001110001111101010	N	**108**
16	1.10010101011010110110111100110100000111110101110100 00	N	**106**
64	1.01100001010101010101111110111010110001000010110110100	D	**106**
128	1.01100001010101010101111110111010110001000010110110100	D	**107**
128	1.11010011000010100100001101110111001111011101000110 11	D	**106**
256	1.01100001010101010101111110111010110001000010110110100	D	**108**
256	1.11010011000010100100001101110111001111011101000110 11	N	**107**
512	1.01100001010101010101111110111010110001000010110110100	D	**109**

TAB. 3.4: *Pires cas de* $\log_2(x)$ *pour lesquels* $m \geqslant 106$. *La première colonne indique l'exposant de l'argument ; seules les valeurs* -1, 0 *et les puissances de* 2 *sont données, car les autres exposants s'en déduisent ; en particulier, notons qu'il n'y a aucun pire cas pour* $E = -1$. *La deuxième colonne donne la mantisse de l'argument. La troisième colonne donne le mode d'arrondi pour lequel il s'agit d'un pire cas : D pour arrondi dirigé, N pour arrondi au plus près. La quatrième colonne donne la valeur de* m.

(Three-distance theorem version – subtractive view).

Lefèvre-Muller-Tisserand's algorithm

From linear functions to the general case:

- over $[x_0 - u/2^p, x_0 + u/2^p]$,
  $f(x) \approx f(x_0) + f'(x_0)(x - x_0) + \mathrm{error};$

Lefèvre-Muller-Tisserand's algorithm

From linear functions to the general case:

- over $[x_0 - u/2^p, x_0 + u/2^p]$,
  $f(x) \approx f(x_0) + f'(x_0)(x - x_0) + \mathrm{error}$;
- error is $\lessapprox u^2/2^{2p}$;

Lefèvre-Muller-Tisserand's algorithm

From linear functions to the general case:

- over $[x_0 - u/2^p, x_0 + u/2^p]$,
  $f(x) \approx f(x_0) + f'(x_0)(x - x_0) + \mathrm{error}$;
- error is $\lessapprox u^2/2^{2p}$;
- ok for worst cases at distance $\delta \geq 2u^2/2^{2p}$.

Lefèvre-Muller-Tisserand's algorithm

From linear functions to the general case:

- over $[x_0 - u/2^p, x_0 + u/2^p]$,
  $f(x) \approx f(x_0) + f'(x_0)(x - x_0) + \mathrm{error}$;
- error is $\lessapprox u^2/2^{2p}$;
- ok for worst cases at distance $\delta \geq 2u^2/2^{2p}$.
- We have a *reduction* from general case to deg. 1 polynomials
- over small intervals.

Lefèvre-Muller-Tisserand's algorithm (II)

Complexity analysis:

- need $\approx 2^p/u$ intervals;

Lefèvre-Muller-Tisserand's algorithm (II)

Complexity analysis:

- need $\approx 2^p/u$ intervals;
- $\approx \delta 2^{2p}$ solutions overall;

Lefèvre-Muller-Tisserand's algorithm (II)

Complexity analysis:

- need $\approx 2^p/u$ intervals;
- $\approx \delta 2^{2p}$ solutions overall;
- minimise $2^p/u + \delta 2^{2p}$:

Lefèvre-Muller-Tisserand's algorithm (II)

Complexity analysis:

- need $\approx 2^p/u$ intervals;
- $\approx \delta 2^{2p}$ solutions overall;
- minimise $2^p/u + \delta 2^{2p}$:
- $u \approx 2^{p/3}$, $\delta \approx 2^{-4p/3}$,

Lefèvre-Muller-Tisserand's algorithm (II)

Complexity analysis:

- need $\approx 2^p/u$ intervals;
- $\approx \delta 2^{2p}$ solutions overall;
- minimise $2^p/u + \delta 2^{2p}$:
- $u \approx 2^{p/3}$, $\delta \approx 2^{-4p/3}$,
- complexity $O(2^{2p/3})$.

Beyond LMT

- Cost of LMT: combinatorial term (# of intervals);
- Higher degree approx $\Rightarrow$ less intervals;
- But how to solve TMD for higher degree pols?

Beyond LMT (2)

- Forget about Ostrowski;
- Replace continued fractions by *lattice basis reduction*.
- LMT $\Leftrightarrow$ find small int. x, y st. $\alpha x - y$ close to β.

Beyond LMT (3)

- Lattice basis reduction:
  - find "small integer linear combinations";
  - given n vectors in $\mathbb{R}^d$ $(n < d)$, find small $x_i \in \mathbb{Z}$ st. $\sum x_i v_i$ is "small";
- In lattice terms: a vector $x \cdot \begin{pmatrix} \alpha \\ 1 \end{pmatrix} + y \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ close to $\begin{pmatrix} \beta \\ 0 \end{pmatrix}$
- CVP problem, lattice basis reduction gives decent algorithms.

Coppersmith's method

(see also Bombieri-Pila (1989) with a geometric view)

- We want to find integers x, y such that $P(x/2^p) - y/2^p$ is small, x small-ish, P polynomial;
- Reformulate & clear denominators as $\tilde{P}(x) - z = 0 \bmod 2^{kp}$,
- x is in a small interval, z is small (function of target $\mu_{f,p}$).

Coppersmith's method (II)

Remark: if $|x|, |z|$ and $\tilde{P}$ small enough, $|\tilde{P}(x)| + |z| < 2^{kp}$

- $\tilde{P}(x) - z = 0 \bmod 2^{kp} \Rightarrow$ equation over the integers;
- (a) too good to be true;
- (b) one equation is not enough.

Coppersmith's method (III)

Build plenty of auxiliary equations!

- If $\tilde{P}(x) - z = 0 \bmod 2^{kp}$
- then $P_{k,l,m} = 2^{kp(d-m)} x^k z^l \left(\tilde{P}(x) - z\right)^m = 0 \bmod 2^{kpd}$,

Coppersmith's method (III)

Build plenty of auxiliary equations!

- If $\tilde{P}(x) - z = 0 \bmod 2^{kp}$
- then $P_{k,l,m} = 2^{kp(d-m)} x^k z^l \left(\tilde{P}(x) - z\right)^m = 0 \bmod 2^{kpd}$,
- with all their linear combinations.

Coppersmith's method (III)

Build plenty of auxiliary equations!

- If $\tilde{P}(x) - z = 0 \bmod 2^{kp}$
- then $P_{k,l,m} = 2^{kp(d-m)} x^k z^l \left(\tilde{P}(x) - z\right)^m = 0 \bmod 2^{kpd}$,
- with all their linear combinations.
- Find integers $c_{k,l,m}$ such that $\sum c_{k,l,m} P_{k,l,m}$ is small...
- aka small integer linear combinations.

Coppersmith's method (III)

Build plenty of auxiliary equations!

- If $\tilde{P}(x) - z = 0 \bmod 2^{kp}$
- then $P_{k,l,m} = 2^{kp(d-m)} x^k z^l \left(\tilde{P}(x) - z\right)^m = 0 \bmod 2^{kpd}$,
- with all their linear combinations.
- Find integers $c_{k,l,m}$ such that $\sum c_{k,l,m} P_{k,l,m}$ is small...
- aka small integer linear combinations.
- That's what lattice basis reduction (LLL) is good at!

Coppersmith's method (IV)

- Start from $\tilde{P}(x) - z = 0 \bmod 2^{kp}$
- Find two small $\sum c_{k,l,m} P_{k,l,m}$, $\sum c'_{k,l,m} P_{k,l,m}$;
- if small enough, they must be 0 at any "bad case" (x, z).
- Get two pol. equations in two variables, solve, done.

Coppersmith's method (IV)

- Start from $\tilde{P}(x) - z = 0 \bmod 2^{kp}$
- Find two small $\sum c_{k,l,m} P_{k,l,m}$, $\sum c'_{k,l,m} P_{k,l,m}$;
- if small enough, they must be 0 at any "bad case" (x, z).
- Get two pol. equations in two variables, solve, done.
- **Bad news:** It may fail.

Coppersmith's method (IV)

- Start from $\tilde{P}(x) - z = 0 \bmod 2^{kp}$
- Find two small $\sum c_{k,l,m} P_{k,l,m}$, $\sum c'_{k,l,m} P_{k,l,m}$;
- if small enough, they must be 0 at any "bad case" (x, z).
- Get two pol. equations in two variables, solve, done.
- **Bad news:** It may fail.
- *Good news:* It usually does not.

Coppersmith's method (IV)

- Start from $\tilde{P}(x) - z = 0 \bmod 2^{kp}$
- Find two small $\sum c_{k,l,m} P_{k,l,m}$, $\sum c'_{k,l,m} P_{k,l,m}$;
- if small enough, they must be 0 at any "bad case" (x, z).
- Get two pol. equations in two variables, solve, done.
- **Bad news:** It may fail.
- *Good news:* It usually does not.
- **Bad news:** except for algebraic functions.

From polynomial case to transcendental functions via LMT-type reduction.

The new frontier: binary128 and lower bounds

- SLZ, then Stehlé:
  - "worst cases" ($\mu_{f,p}(x) = p$): $O(2^{p/2+\varepsilon})$;
  - gives access to *lower bounds*:
  - ie. proves non-existence of x with $\mu(f, x) > k$;
- theoretical: polynomial time algorithm for $k = p^2$.
- weak spot: long computation $\rightarrow \emptyset$.
- need for certification: Martin-Dorel, Mayero, Théry, H. (2015).

Beyond SLZ!

Joint work with Nicolas Brisebarre.

- Main cost of SLZ: lattice reduction calls over each interval;
- Two lattices for two neighbouring intervals should be close!

Beyond SLZ!

Joint work with Nicolas Brisebarre.

- Main cost of SLZ: lattice reduction calls over each interval;
- Two lattices for two neighbouring intervals should be close!
- Reuse change-of-basis matrix? *pre-reduction*
- ...[ST] Does not work for SLZ.

Beyond SLZ!

- Revisited SLZ with “modern” ingredients;
- Working with f rather than reducing to P.
- Chebyshev polynomials rather than Taylor;
- Sharper analysis: better constants in some exponents;
- Sharper analysis: $p^2/\log p$ rather than p^2;
- **Avoid *reduction* from function to polynomial**.

Beyond SLZ!

- LMT, SLZ *reduce* TMD(f) to TMD(P);
- build auxiliary polynomials with $Q_1(X, P(X)) = Q_2(X, P(X)) = 0$.
- we work *with* f: $Q_1(X, f(X)) = Q_2(X, f(X)) = 0$.
- by using a representation of $x^j y^k f^l$ as $P_{jkl}(x, y) + R_{jkl}$.

f does not change a lot while $x^j y^k P^l$ does...

Beyond SLZ! – practical results

$\log_2(w)$	d	N	N_1	N_2	ρ_1	$b_1 - a_1$	Timing	% LLL
$4p$	6	28	20	3	$2^{39.8}$	$2^{-36.75}$	193^* years	78%
$6p$	8	90	45	2	$2^{28.4}$	$2^{-26.55}$	1.56^* years	82%
$8p$	9	55	50	2	$2^{24.3}$	$2^{-23.55}$	36.3 days	89%
$10p$	10	66	70	2	2^{25}	$7/2^{23}$	11.5 days	89%
$12p$	12	91	80	2	2^{21}	$5/2^{19}$	5.0 days	96%

Table 7.1. Algo. 1 and 2: exp over the binade $[1/4, 1/2)$

Beyond SLZ! – practical results

$\log_2(w)$	α_{Ste}	d_{Ste}	t_{Ste}	Timing	Comparison with this paper
$4p$	5	10	71	$\approx 19200^*$ years	$\times 100$
$6p$	8	20	83.7	422^* years	$\times 270$
$8p$	9	30	87.4	90^* years	$\times 906$
$10p$	10	42	91	30^* years	$\times 953$
$12p$	12	56	94.3	31^* years	$\times 2264$

Table 7.2. Stehlé's BaCSeL parameters and timings for the exponential function over the binade $[1/4, 1/2)$

Beyond SLZ! – practical results

- Previous comments apply (certification/formal proof needed);
- Embarrassingly parallel;
- $6p$ for a few binades / functions seems realistic;
- Beyond Coppersmith?

As a conclusion

A tribute to JMM.