Number Theory and Efficient Evaluation of Functions on a Machine

Nicolas Brisebarre Sylvain Chevillard Guillaume Hanrot
Mioara Joldeș Jean-Michel Muller Arnaud Tisserand
Serge Torres

Floating Point (FP) Arithmetic

Given

$$\begin{cases} \text{a radix} & \beta \geq 2, \\ \text{a precision} & n \geq 1, \\ \text{a set of exponents} & E_{\mathsf{min}} \cdots E_{\mathsf{max}}. \end{cases}$$

A finite FP number x is represented by 2 integers:

- integer mantissa : M, $\beta^{n-1} \leq |M| \leq \beta^n - 1$;
- exponent e, $E_{\mathsf{min}} \leq e \leq E_{\mathsf{max}}$

such that

$$x = \frac{M}{\beta^{n-1}} \times \beta^e.$$

We assume binary FP arithmetic (that is to say $\beta = 2$.)

IEEE Precisions

Voir http://en.wikipedia.org/wiki/IEEE_754-2008 ou (plus ancien) http://babbage.cs.qc.edu/courses/cs341/IEEE-754references.html.

	precision	minimal exponent	maximal exponent
single (binary 32)	24	-126	127
double (binary 64)	53	-1022	1023
extended double	64	-16382	16383
quadruple (binary 128)	113	-16382	16383

Applications

Two targets:

- specific hardware implementations in low precision (~ 15 bits). Reduce the cost (time and silicon area) keeping a correct accuracy;
- single or double IEEE precision software implementations. Get very high accuracy keeping an acceptable cost (time and memory).

Evaluation of Elementary Functions

$\exp, \ln, \cos, \sin, \arctan, \sqrt{\ }, \ldots$

Goal: evaluation of φ to a given accuracy η.

Evaluation of Elementary Functions

$\exp, \ln, \cos, \sin, \arctan, \sqrt{\ }, \ldots$

Goal: evaluation of φ to a given accuracy η.

- Step 1. Argument reduction (Payne & Hanek, Ng, Daumas *et al*): evaluation of a function φ over $\mathbb{R}$ or a subset of $\mathbb{R}$ is reduced to the evaluation of a function f over $[a, b]$.

Evaluation of Elementary Functions

$\exp, \ln, \cos, \sin, \arctan, \sqrt{\ }, \ldots$

Goal: evaluation of φ to a given accuracy η.

- Step 1. Argument reduction (Payne & Hanek, Ng, Daumas *et al*): evaluation of a function φ over $\mathbb{R}$ or a subset of $\mathbb{R}$ is reduced to the evaluation of a function f over $[a, b]$.
- Step 2. Computation of $p^\star$, a "machine-efficient" polynomial approximation of f.

Evaluation of Elementary Functions

$\exp, \ln, \cos, \sin, \arctan, \sqrt{\ }, \ldots$

Goal: evaluation of φ to a given accuracy η.

- Step 1. Argument reduction (Payne & Hanek, Ng, Daumas *et al*): evaluation of a function φ over $\mathbb{R}$ or a subset of $\mathbb{R}$ is reduced to the evaluation of a function f over $[a, b]$.
- Step 2. Computation of $p^\star$, a "machine-efficient" polynomial approximation of f.
- Step 3. Computation of a rigorous approximation error $||f - p^\star||$.

Evaluation of Elementary Functions

$\exp, \ln, \cos, \sin, \arctan, \sqrt{\ }, \ldots$

Goal: evaluation of φ to a given accuracy η.

- Step 1. Argument reduction (Payne & Hanek, Ng, Daumas *et al*): evaluation of a function φ over $\mathbb{R}$ or a subset of $\mathbb{R}$ is reduced to the evaluation of a function f over $[a, b]$.
- Step 2. Computation of $p^\star$, a "machine-efficient" polynomial approximation of f.
- Step 3. Computation of a rigorous approximation error $||f - p^\star||$.
- Step 4. Computation of a certified evalutation error of $p^\star$: GAPPA (G. Melquiond).

Evaluation of Elementary Functions

$\exp, \ln, \cos, \sin, \arctan, \sqrt{\ }, \ldots$

Goal: evaluation of φ to a given accuracy η.

- Step 0. Computation of hardest-to-round cases: V. Lefèvre and J.-M. Muller.
- Step 1. Argument reduction (Payne & Hanek, Ng, Daumas *et al*): evaluation of a function φ over $\mathbb{R}$ or a subset of $\mathbb{R}$ is reduced to the evaluation of a function f over $[a, b]$.
- Step 2. Computation of $p^\star$, a "machine-efficient" polynomial approximation of f.
- Step 3. Computation of a rigorous approximation error $||f - p^\star||$.
- Step 4. Computation of a certified evalutation error of $p^\star$: GAPPA (G. Melquiond).

Rational Approximation

Let $p/q \in \mathbb{Q}$, we define the height of p/q by $h(p/q) = \max(|p|, |q|)$.

Property

$\mathbb{Q}$ *is dense in* $\mathbb{R}$ *: for all* $x \in \mathbb{R}$*, for all* $\varepsilon > 0$*, there exists* $p/q \in \mathbb{Q}$ *s.t.*

$$\left| x - \frac{p}{q} \right| < \varepsilon.$$

Rational Approximation

Let $p/q \in \mathbb{Q}$, we define the height of p/q by $h(p/q) = \max(|p|, |q|)$.

Property

$\mathbb{Q}$ *is dense in* $\mathbb{R}$ *: for all* $x \in \mathbb{R}$, *for all* $\varepsilon > 0$, *there exists* $p/q \in \mathbb{Q}$ *s.t.*

$$\left| x - \frac{p}{q} \right| < \varepsilon.$$

Not precise enough to be useful to us: we'd rather have $h(p, q)$ as small as possible.

Rational Approximation

Let $p/q \in \mathbb{Q}$, we define the height of p/q by $h(p/q) = \max(|p|, |q|)$.

Property

$\mathbb{Q}$ is dense in $\mathbb{R}$: for all $x \in \mathbb{R}$, for all $\varepsilon > 0$, there exists $p/q \in \mathbb{Q}$ s.t.

$$\left| x - \frac{p}{q} \right| < \varepsilon.$$

Not precise enough to be useful to us: we'd rather have $h(p, q)$ as small as possible.
More precisely: given $x \in \mathbb{R}$, we want to minimize $|x - p/q|$ or $|qx - p|$ with $p \in \mathbb{Z}$, $q \in \mathbb{N} \setminus \{0\}$ as small as possible.

Notations

Let $x \in \mathbb{R}$. We denote $\lfloor x \rfloor$ the floor part of x, $\{x\}$ the fractional part of x and $\|x\| = \min_{n \in \mathbb{Z}} |x - n|$ the distance of x to the nearest integer.

Notations

Let $x \in \mathbb{R}$. We denote $\lfloor x \rfloor$ the floor part of x, $\{x\}$ the fractional part of x and $\|x\| = \min_{n \in \mathbb{Z}} |x - n|$ the distance of x to the nearest integer.

We want to minimize $|x - p/q|$ ou $|qx - p|$ with $p \in \mathbb{Z}$, $q \in \mathbb{N} \setminus \{0\}$ as small as possible.

Notations

Let $x \in \mathbb{R}$. We denote $\lfloor x \rfloor$ the floor part of x, $\{x\}$ the fractional part of x and $\|x\| = \min_{n \in \mathbb{Z}} |x - n|$ the distance of x to the nearest integer.

We want to minimize $|x - p/q|$ ou $|qx - p|$ with $p \in \mathbb{Z}$, $q \in \mathbb{N} \setminus \{0\}$ as small as possible.

We consider $\|qx\|$ for $q \in \mathbb{N}^*$ in the sequel.

Best Approximations to a Real Number

Let $x \in \mathbb{R}$.

Definition

A fraction p/q ($p \in \mathbb{Z}$ and $q \in \mathbb{N}$, $q \neq 0$) is a best approximation to x if and only if

$$\begin{cases} \|qx\| = |qx - p| & \textit{and} \\ \|q'x\| > \|qx\| & \textit{for } 0 < q' < q. \end{cases}$$

Best Approximations to a Real Number

Theorem (Dirichlet)

Let $x \in \mathbb{R}$ *and* $Q \in]1, +\infty[$*. Then there exists* $q \in \mathbb{N} \cap]0, Q[$ *s. t.* $\|qx\| \leq Q^{-1}$*.*

Best Approximations to a Real Number

Theorem (Dirichlet)

Let $x \in \mathbb{R}$ *and* $Q \in]1, +\infty[$*. Then there exists* $q \in \mathbb{N} \cap]0, Q[$ *s. t.* $\|qx\| \leq Q^{-1}$*.*

We set $q_1 = 1$. Let p_1 s.t.. $\|q_1 x\| = \|x\| = |x - p_1|$. p_1/q_1 is a best rational approximation to x.

Best Approximations to a Real Number

Theorem (Dirichlet)

Let $x \in \mathbb{R}$ and $Q \in]1, +\infty[$. Then there exists $q \in \mathbb{N} \cap]0, Q[$ s. t. $\|qx\| \leq Q^{-1}$.

We set $q_1 = 1$. Let p_1 s.t.. $\|q_1 x\| = \|x\| = |x - p_1|$. p_1/q_1 is a best rational approximation to x.
If $\|q_1 x\| = 0$, $x \in \mathbb{Z}$ and p_1/q_1 is its only one best approx.

Best Approximations to a Real Number

Theorem (Dirichlet)

Let $x \in \mathbb{R}$ and $Q \in]1, +\infty[$. Then there exists $q \in \mathbb{N} \cap]0, Q[$ s. t. $\|qx\| \leq Q^{-1}$.

We set $q_1 = 1$. Let p_1 s.t.. $\|q_1 x\| = \|x\| = |x - p_1|$. p_1/q_1 is a best rational approximation to x.
If $\|q_1 x\| = 0$, $x \in \mathbb{Z}$ and p_1/q_1 is its only one best approx.
If $\|q_1 x\| > 0$, Dirichlet Theorem with $Q > \|q_1 x\|^{-1} \Rightarrow$ there exists $q \in \mathbb{N}$ s.t.. $\|qx\| \leq Q^{-1} < \|q_1 x\|$.

Best Approximations to a Real Number

Theorem (Dirichlet)

Let $x \in \mathbb{R}$ and $Q \in]1, +\infty[$. Then there exists $q \in \mathbb{N} \cap]0, Q[$ s. t. $\|qx\| \leq Q^{-1}$.

We set $q_1 = 1$. Let p_1 s.t.. $\|q_1 x\| = \|x\| = |x - p_1|$. p_1/q_1 is a best rational approximation to x.
If $\|q_1 x\| = 0$, $x \in \mathbb{Z}$ and p_1/q_1 is its only one best approx.
If $\|q_1 x\| > 0$, Dirichlet Theorem with $Q > \|q_1 x\|^{-1} \Rightarrow$ there exists $q \in \mathbb{N}$ s.t.. $\|qx\| \leq Q^{-1} < \|q_1 x\|$.
Let q_2 be the smallest integer q s.t. $\|qx\| < \|q_1 x\|$. Let p_2 s.t. $\|q_2 x\| = |q_2 x - p_2|$.

Best Approximations to a Real Number

Theorem (Dirichlet)

Let $x \in \mathbb{R}$ and $Q \in]1, +\infty[$. Then there exists $q \in \mathbb{N} \cap]0, Q[$ s. t. $\|qx\| \leq Q^{-1}$.

We set $q_1 = 1$. Let p_1 s.t.. $\|q_1 x\| = \|x\| = |x - p_1|$. p_1/q_1 is a best rational approximation to x.
If $\|q_1 x\| = 0$, $x \in \mathbb{Z}$ and p_1/q_1 is its only one best approx.
If $\|q_1 x\| > 0$, Dirichlet Theorem with $Q > \|q_1 x\|^{-1} \Rightarrow$ there exists $q \in \mathbb{N}$ s.t.. $\|qx\| \leq Q^{-1} < \|q_1 x\|$.
Let q_2 be the smallest integer q s.t. $\|qx\| < \|q_1 x\|$. Let p_2 s.t. $\|q_2 x\| = |q_2 x - p_2|$.
By construction, p_2/q_2 is a best rational approximation to x, with the smallest denominator greater than q_1.

Best Approximations to a Real Number

We iterate this process. We get, by induction, a strictly increasing sequence (finite if $x \in \mathbb{Q}$) of integers $1 = q_1 < q_2 < \cdots$ and a sequence of integers $p_1, p_2, \ldots$ s. t. :

$$\|q_n x\| = |q_n x - p_n|\,, \tag{1}$$
$$\|q_{n+1} x\| < \|q_n x\|\,, \tag{2}$$
$$\|q x\| \geq \|q_n x\| \quad \text{pour } 0 < q < q_{n+1}. \tag{3}$$

From (1-3), the p_n/q_n are best approximations to x.

Best Approximations to a Real Number

By construction, p_1/q_1 is a best approximation, and p_{n+1}/q_{n+1} is the best approximation of x with the smallest denominator greater than q_n.

Best Approximations to a Real Number

By construction, p_1/q_1 is a best approximation, and p_{n+1}/q_{n+1} is the best approximation of x with the smallest denominator greater than q_n. We proved

Theorem

The p_n/q_n are the best approximations to x, sorted by increasing size of the denominators.

If x is a rational number

If $x \in \mathbb{Q}$ i.e. $x = a/b$ with $a \in \mathbb{Z}$, $b \in \mathbb{N}^*$ and $\mathrm{pgcd}(a, b) = 1$, then a/b is a best approximation to x.
There exists $N \in \mathbb{N}$ s.t. $x = p_N/q_N$ and, as $\|q_N x\| = 0$, the process stops at the order N : the number of best approximations is finite.

If x is an irrational number

If $x \notin \mathbb{Q}$, then the sequence p_n/q_n converges to x. From the previous results, we get

$$q_n\|q_n x\| < q_{n+1}\|q_n x\| \leq 1\,.$$

If x is an irrational number

If $x \notin \mathbb{Q}$, then the sequence p_n/q_n converges to x. From the previous results, we get

$$q_n \|q_n x\| < q_{n+1} \|q_n x\| \leq 1 \, .$$

Hence, we have

$$\left| x - \frac{p_n}{q_n} \right| = \frac{1}{q_n} \|q_n x\| \leq \frac{1}{q_n q_{n+1}} < \frac{1}{q_n^2} \, ,$$

which shows $\lim_{n \to \infty} p_n/q_n = x$.

If x is an irrational number

If $x \notin \mathbb{Q}$, we have, for all $n \geq 0$,

$$\left| x - \frac{p_n}{q_n} \right| \leq \frac{1}{q_n q_{n+1}} \left(\leq \frac{1}{q_n^2} \right).$$

If x is an irrational number

If $x \notin \mathbb{Q}$, we have, for all $n \geq 0$,

$$\left| x - \frac{p_n}{q_n} \right| \leq \frac{1}{q_n q_{n+1}} \left(\leq \frac{1}{q_n^2} \right).$$

Note that we can also show, for all $n \geq 0$,

$$\frac{1}{q_n(q_n + q_{n+1})} \leq \left| x - \frac{p_n}{q_n} \right|.$$

An Algorithm for Computing Best Rational Approximations to x

We need some preliminary properties.

Proposition

i) For $n \in \mathbb{N}^*$, we have

$$|p_n q_{n+1} - p_{n+1} q_n| = q_n \|q_{n+1} x\| + q_{n+1} \|q_n x\| = 1\,.$$

ii) For $n \in \mathbb{N}^*$, the numbers $q_n x - p_n$, $p_{n+1} - q_{n+1} x$, $p_{n+1} q_n - p_n q_{n+1}$ (and $p_{n-1} q_n - p_n q_{n-1}$ for $n \geq 2$) have the same sign.

An Algorithm for Computing Best Rational Approximations to x

For $n \geq 2$, $p_{n+1}q_n - p_n q_{n+1}$ et $p_{n-1}q_n - p_n q_{n-1}$ have the same absolute value and sign: they are equal. Therefore

$$p_n(q_{n+1} - q_{n-1}) = (p_{n+1} - p_{n-1})q_n \, .$$

An Algorithm for Computing Best Rational Approximations to x

For $n \geq 2$, $p_{n+1}q_n - p_nq_{n+1}$ et $p_{n-1}q_n - p_nq_{n-1}$ have the same absolute value and sign: they are equal. Therefore

$$p_n(q_{n+1} - q_{n-1}) = (p_{n+1} - p_{n-1})q_n \,.$$

But p_n are q_n coprime. There exists $a_n \in \mathbb{Z}$ s.t. $p_{n+1} - p_{n-1} = a_np_n$ and $q_{n+1} - q_{n-1} = a_nq_n$. Moreover, $a_n \in \mathbb{N}^*$ since $q_{n+1} > q_{n-1}$.

An Algorithm for Computing Best Rational Approximations to x

We have (from the previous Prop.)

$$\|q_{n-1}x\| - \|q_{n+1}x\| = |(q_{n+1}x - p_{n+1}) - (q_{n-1}x - p_{n-1})| = |a_n(q_nx - p_n)|,$$

that is to say

$$\|q_{n-1}x\| = a_n\|q_nx\| + \|q_{n+1}x\|.$$

An Algorithm for Computing Best Rational Approximations to x

We have (from the previous Prop.)

$$\|q_{n-1}x\| - \|q_{n+1}x\| = |(q_{n+1}x - p_{n+1}) - (q_{n-1}x - p_{n-1})| = |a_n(q_nx - p_n)|,$$

that is to say

$$\|q_{n-1}x\| = a_n\|q_nx\| + \|q_{n+1}x\|.$$

It makes it possible to compute (p_{n+1}, q_{n+1}) from $(p_{n-1}, q_{n-1}, p_n, q_n)$.

An Algorithm for Computing Best Rational Approximations to x

We have (from the previous Prop.)

$$\|q_{n-1}x\| - \|q_{n+1}x\| = |(q_{n+1}x - p_{n+1}) - (q_{n-1}x - p_{n-1})| = |a_n(q_nx - p_n)|\,,$$

that is to say

$$\|q_{n-1}x\| = a_n\|q_nx\| + \|q_{n+1}x\|\,.$$

It makes it possible to compute (p_{n+1}, q_{n+1}) from $(p_{n-1}, q_{n-1}, p_n, q_n)$.
If $\|q_nx\| = 0$, $x \in \mathbb{Q}$ and p_n/q_n is the last best rational approximation to x ; otherwise, as $\|q_{n+1}x\|/\|q_nx\| \in [0, 1[$, we have the formula

$$a_n = \left\lfloor \frac{\|q_{n-1}x\|}{\|q_nx\|} \right\rfloor = \left\lfloor \frac{|q_{n-1}x - p_{n-1}|}{|q_nx - p_n|} \right\rfloor\,,$$

and the recurrence relations

$$\begin{cases} p_{n+1} = a_np_n + p_{n-1}\,, \\ q_{n+1} = a_nq_n + q_{n-1}\,. \end{cases}$$

An Algorithm for Computing Best Rational Approximations to x

Theorem

Let $x \in \mathbb{R}$. We define the sequences (finite or infinite) (p_n), (q_n) and (a_n) by the initial conditions $a_0 = \lfloor x \rfloor$, $(p_0, q_0, p_1, q_1) = (1, 0, \lfloor x \rfloor, 1)$ and the recurrence relations defined for $n \in \mathbb{N}^*$

$$\begin{cases} p_{n+1} = a_n p_n + p_{n-1} \, , \\ q_{n+1} = a_n q_n + q_{n-1} \, , \end{cases}$$

where

$$a_n = \left\lfloor \frac{|q_{n-1}x - p_{n-1}|}{|q_n x - p_n|} \right\rfloor$$

if $q_n x \neq p_n$ (the sequences are finite if $q_n x = p_n$). Then the p_n/q_n are best rational approximations to x for $n \geq 1$ if $a_1 \geq 2$, and for $n \geq 2$ if $a_1 = 1$.

An Algorithm for Computing Best Rational Approximations to x

Theorem

Let $x \in \mathbb{R}$. We define the sequences (finite or infinite) (p_n), (q_n) and (a_n) by the initial conditions $a_0 = \lfloor x \rfloor$, $(p_0, q_0, p_1, q_1) = (1, 0, \lfloor x \rfloor, 1)$ and the recurrence relations defined for $n \in \mathbb{N}^*$

$$\begin{cases} p_{n+1} = a_n p_n + p_{n-1}\,, \\ q_{n+1} = a_n q_n + q_{n-1}\,, \end{cases}$$

where

$$a_n = \left\lfloor \frac{|q_{n-1}x - p_{n-1}|}{|q_n x - p_n|} \right\rfloor$$

if $q_n x \neq p_n$ (the sequences are finite if $q_n x = p_n$). Then the p_n/q_n are best rational approximations to x for $n \geq 1$ if $a_1 \geq 2$, and for $n \geq 2$ if $a_1 = 1$.

If we put $r_0 = x$, $r_n = \dfrac{1}{r_{n-1} - a_{n-1}}$, we have $a_n = \lfloor r_n \rfloor$.

An Algorithm for Computing Best Rational Approximations to x

As $p_0q_1 - p_1q_0 = 1$ et $q_0x - p_0 = -1$, the initial Proposition gives

$$(-1)^{n+1}(q_nx - p_n) \geq 0 \text{ and } p_nq_{n+1} - p_{n+1}q_n = (-1)^n \,.$$

We have, for all $n \geq 2$,

$$\frac{p_{n-2}}{q_{n-2}} - \frac{p_n}{q_n} = (-1)^{n-1}\frac{a_n}{q_nq_{n-2}}.$$

An Algorithm for Computing Best Rational Approximations to x

The fractions p_n/q_n are called the convergents and the a_n the partial quotients. Since the knowledge of the sequence (a_n) is equivalent to the one of x ($x = p_N/q_N$ if $x \in \mathbb{Q}$ and $x = \lim_{n\to\infty} p_n/q_n$ if $x \notin \mathbb{Q}$), we will use the notation $x = [a_0, a_1, a_2, \ldots]$ (the number of terms between brackets can be finite).

An Algorithm for Computing Best Rational Approximations to x

The fractions p_n/q_n are called the convergents and the a_n the partial quotients. Since the knowledge of the sequence (a_n) is equivalent to the one of x ($x = p_N/q_N$ if $x \in \mathbb{Q}$ and $x = \lim_{n\to\infty} p_n/q_n$ if $x \notin \mathbb{Q}$), we will use the notation $x = [a_0, a_1, a_2, \ldots]$ (the number of terms between brackets can be finite).

Let, for $n \in \mathbb{N}^*$,

$$x_n = \frac{|q_n x - p_n|}{|q_{n-1}x - p_{n-1}|},$$

so that we have

$$x_0 = x^{-1} \text{ and } x_n^{-1} = a_n + x_{n+1}.$$

An Algorithm for Computing Best Rational Approximations to x

So we get

$$x = \frac{1}{x_0} = a_0 + x_1 = a_0 + \frac{1}{a_1 + x_2}$$
$$= a_0 + \frac{1}{a_1 + \frac{1}{a_2 + x_3}} = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \cdots}}}.$$

This writing explains the name "continued fraction". Notice that

$$x_n = \frac{1}{a_n + x_{n+1}} = \frac{1}{a_n + \frac{1}{a_{n+1} + \cdots}} = [a_n, a_{n+1}, \ldots].$$

Examples

An original example: $\pi = 3.14159265358979....$

$$\pi = 3 + \cfrac{1}{7 + \cfrac{1}{15 + \cfrac{1}{1 + \cfrac{1}{292 + \ldots,}}}}$$

Examples

An original example: $\pi = 3.14159265358979....$

$$\pi = 3 + \cfrac{1}{7 + \cfrac{1}{15 + \cfrac{1}{1 + \cfrac{1}{292 + \dots,}}}}$$

Its first convergents are $\frac{p_1}{q_1} = 3$, $\frac{p_2}{q_2} = \frac{22}{7} = 3.142...$, $\frac{p_3}{q_3} = \frac{333}{106} = 3.14150...$, $\frac{p_4}{q_4} = \frac{355}{113} = 3.1415929...$, $\frac{p_5}{q_5} = \frac{103993}{33102} = 3.1415926530....$

Examples

We have

$$\sqrt{2} = 1 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{2 + \dots}}}}$$

which we can denote $\sqrt{2} = [1, \overline{2}]$.

Examples

We have

$$\sqrt{2} = 1 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{2 + \dots}}}}$$

which we can denote $\sqrt{2} = [1, \overline{2}]$. We also have $\sqrt{3} = [1, \overline{1, 2}]$, $\sqrt{5} = [2, \overline{4}]$, $\sqrt{7} = [2, \overline{1, 1, 1, 4}]$.

A quadratic number has a periodic continued fraction expansion: there exist k and $L \in \mathbb{N}$ s.t. $a_l = a_{l+k}$ for all $l \geq L$.

Euler : $e = [2, 1, 2, 1, 1, 4, 1, 1, 6, 1, \dots, 1, 2n, 1, \dots]$.

Two classical results

Reminder. The best approximations p_n/q_n of a real number x satisfy $q_n|q_n x - p_n| < 1$.

Theorem

Among two consecutive best approximations to x, one at least satisfies the inequality $q\|qx\| < 1/2$.

Two classical results

Reminder. The best approximations p_n/q_n of a real number x satisfy $q_n|q_n x - p_n| < 1$.

Theorem

Among two consecutive best approximations to x, one at least satisfies the inequality $q\|qx\| < 1/2$.

And the reciprocal:

Theorem (Legendre)

Si $q|qx - p| < 1/2$, alors p/q est une réduite de x.

Worst Cases for Argument Reduction

W. Kahan. "Minimizing $q * m - n$", available at
`http://www.cs.berkeley.edu/~wkahan/testpi/`, text at the beginning of nearpi.c

Evaluation of Elementary Functions

$\exp, \ln, \cos, \sin, \arctan, \sqrt{\ }, \ldots$

Goal: evaluation of φ to a given accuracy η.

- Step 0. Computation of hardest-to-round cases: V. Lefèvre and J.-M. Muller.
- Step 1. Argument reduction (Payne & Hanek, Ng, Daumas *et al*): evaluation of a function φ over $\mathbb{R}$ or a subset of $\mathbb{R}$ is reduced to the evaluation of a function f over $[a, b]$.
- Step 2. Computation of $p^\star$, a "machine-efficient" polynomial approximation of f .
- Step 3. Computation of a rigorous approximation error $||f - p^\star||$.
- Step 4. Computation of a certified evalutation error of $p^\star$: GAPPA (G. Melquiond).

Argument reduction

In order to evaluate $\varphi(x)$ with x a floating-point number, we transform x into x^*, the reduced argument. x^* belongs to the convergence domain of an elementary function f. We know how to get $\varphi(x)$ from $f(x^*)$.

Example: The cos function (additive argument reduction)

The goal is to evaluate $\cos$. The convergence domain of the evaluation algorithm of $\sin$ and $\cos$ contains $[-\pi/4, +\pi/4]$.

Example: The cos function (additive argument reduction)

The goal is to evaluate $\cos$. The convergence domain of the evaluation algorithm of $\sin$ and $\cos$ contains $[-\pi/4, +\pi/4]$. We decompose the computation of $\cos(x)$ into three steps:

- we compute x^* and k s. t. $x^* \in [-\pi/4, +\pi/4]$ and $x^* = x - k\pi/2$;

Example: The cos function (additive argument reduction)

The goal is to evaluate $\cos$. The convergence domain of the evaluation algorithm of $\sin$ and $\cos$ contains $[-\pi/4, +\pi/4]$. We decompose the computation of $\cos(x)$ into three steps:

- we compute x^* and k s. t. $x^* \in [-\pi/4, +\pi/4]$ and $x^* = x - k\pi/2$;
- we compute $g(x^*, k) =$

$$\begin{cases} \cos(x^*) & \text{if} \quad k \text{ mod } 4 = 0 \\ -\sin(x^*) & \text{if} \quad k \text{ mod } 4 = 1 \\ -\cos(x^*) & \text{if} \quad k \text{ mod } 4 = 2 \\ \sin(x^*) & \text{if} \quad k \text{ mod } 4 = 3; \end{cases}$$

Example: The cos function (additive argument reduction)

The goal is to evaluate $\cos$. The convergence domain of the evaluation algorithm of $\sin$ and $\cos$ contains $[-\pi/4, +\pi/4]$. We decompose the computation of $\cos(x)$ into three steps:

- we compute x^* and k s. t. $x^* \in [-\pi/4, +\pi/4]$ and $x^* = x - k\pi/2$;
- we compute $g(x^*, k) =$

$$\begin{cases} \cos(x^*) & \text{if} \quad k \bmod 4 = 0 \\ -\sin(x^*) & \text{if} \quad k \bmod 4 = 1 \\ -\cos(x^*) & \text{if} \quad k \bmod 4 = 2 \\ \sin(x^*) & \text{if} \quad k \bmod 4 = 3; \end{cases}$$

- we obtain $\cos(x) = g(x^*, k)$.

Argument reduction: loss of accuracy problems

Reminder. We compute x^* and k s. t. $x^* \in [-\pi/4, +\pi/4]$ and $x^* = x - k\pi/2$.

Argument reduction: loss of accuracy problems

Reminder. We compute x^* and k s. t. $x^* \in [-\pi/4, +\pi/4]$ and $x^* = x - k\pi/2$.
Extreme caution required!!! We can encounter terrible loss of accuracy problems.

Argument reduction: loss of accuracy problems

Reminder. We compute x^* and k s. t. $x^* \in [-\pi/4, +\pi/4]$ and $x^* = x - k\pi/2$.
Extreme caution required!!! We can encounter terrible loss of accuracy problems.

Naive method: compute

$$\begin{aligned} k &= \left\lfloor \frac{x}{\pi/2} \right\rfloor, \\ x^* &= x - k\pi/2, \end{aligned}$$

using machine precision.

Argument reduction: loss of accuracy problems

Reminder. We compute x^* and k s. t. $x^* \in [-\pi/4, +\pi/4]$ and $x^* = x - k\pi/2$.
Extreme caution required!!! We can encounter terrible loss of accuracy problems.

Naive method: compute

$$\begin{aligned} k &= \left\lfloor \frac{x}{\pi/2} \right\rfloor, \\ x^* &= x - k\pi/2, \end{aligned}$$

using machine precision.

Problem: when $k\pi/2$ near x, almost all (or all) the accuracy is lost when computing $x - k\pi/2$.

Argument Reduction: loss of accuracy problems

Example : if $x = 8248.251512$, right value of $x^* = -2.14758367\cdots \times 10^{-12}$, and $k = 5251$.

Argument Reduction: loss of accuracy problems

Example : if $x = 8248.251512$, right value of $x^* = -2.14758367 \cdots \times 10^{-12}$, and $k = 5251$.

Computation of $x - k\pi/2$ on a pocket calculator (10 digits) using rounding to nearest mode and a rounding of $\pi/2$. We get -1.0×10^{-6}.

Argument Reduction: loss of accuracy problems

Example : if $x = 8248.251512$, right value of $x^* = -2.14758367 \cdots \times 10^{-12}$, and $k = 5251$.

Computation of $x - k\pi/2$ on a pocket calculator (10 digits) using rounding to nearest mode and a rounding of $\pi/2$. We get -1.0×10^{-6}.

Conclusion: the computed value of $\cos(x)$ is -1.0×10^{-6}, but the correct value is $-2.14758367 \cdots \times 10^{-12}$.

Argument Reduction: loss of accuracy problems

First solution: multiple precision. Problems: slow, difficult to know the necessary accuracy beforehand.

Argument Reduction: loss of accuracy problems

First solution: multiple precision. Problems: slow, difficult to know the necessary accuracy beforehand.

Second solution: computing the worst cases, i.e. those which are the hardest to round. Method due to W. Kahan.

We still address the $\cos$ example with a reduction mod $\pi/2$. Let $C = \pi/2$.
We use radix r, with mantissas over n r-its and exponents between e_{min} and e_{max}.
Our x has the following form

$$x = x_0.x_1x_2x_3\cdots x_{n-1} \times r^E, \text{ avec } x_0 \neq 0$$

or

$$x = M \times r^{E-n+1},$$

with $M = x_0x_1x_2x_3\cdots x_{n-1}$, $r^{n-1} \leq M \leq r^n - 1.$

We search for $p \in \mathbb{Z}$ and $s \in \mathbb{R}$, $|s| < 1/2$ s.t. $\dfrac{x}{C} = p + s.$

We search for $p \in \mathbb{Z}$ and $s \in \mathbb{R}$, $|s| < 1/2$ s.t. $\dfrac{x}{C} = p + s.$
$\Rightarrow$ x^* is equal to sC.

We search for $p \in \mathbb{Z}$ and $s \in \mathbb{R}$, $|s| < 1/2$ s.t. $\dfrac{x}{C} = p + s.$

$\Rightarrow x^*$ is equal to sC.

Remember that $x = M \times r^{E-n+1}$. Therefore we search for

$$\frac{r^{E-n+1}}{C} = \frac{p}{M} + \frac{s}{M}.$$

x^* is very small: p/M is a very good approximation to r^{E-n+1}/C.

Let $(p_n/q_n)_{n\geq 0}$ the convergent sequence of $\dfrac{r^{E-n+1}}{C}$.
Let $j = \max\{k \in \mathbb{N} \text{ t.q. } q_k \leq r^n - 1\}$.
We have
$$\left|p - M\frac{r^{E-n+1}}{C}\right| \geq \left|p_j - q_j\frac{r^{E-n+1}}{C}\right|.$$

Let $(p_n/q_n)_{n\geq 0}$ the convergent sequence of $\dfrac{r^{E-n+1}}{C}$.
Let $j = \max\{k \in \mathbb{N} \text{ t.q. } q_k \leq r^n - 1\}$.
We have

$$\left|p - M\frac{r^{E-n+1}}{C}\right| \geq \left|p_j - q_j\frac{r^{E-n+1}}{C}\right|.$$

Therefore

$$\left|pC - Mr^{E-n+1}\right| \geq \varepsilon_E := C\left|p_j - q_j\frac{r^{E-n+1}}{C}\right|.$$

Let $(p_n/q_n)_{n\geq 0}$ the convergent sequence of $\dfrac{r^{E-n+1}}{C}$.
Let $j = \max\{k \in \mathbb{N} \text{ t.q. } q_k \leq r^n - 1\}$.
We have

$$\left| p - M\frac{r^{E-n+1}}{C} \right| \geq \left| p_j - q_j\frac{r^{E-n+1}}{C} \right|.$$

Therefore

$$\left| pC - Mr^{E-n+1} \right| \geq \varepsilon_E := C\left| p_j - q_j\frac{r^{E-n+1}}{C} \right|.$$

Let $\varepsilon = \min_{e_{\mathsf{min}}\leq E\leq e_{\mathsf{max}}} \varepsilon_E$. The number $-\log_r(\varepsilon)$ makes it possible to know the precision accuracy which is necessary to perform a safe reduction argument.

Worst Cases for the Additive Argument Reduction for Several Floating-Points Systems and Constants C

r	n	C	e_{max}	Worst case	$-\log_r(\varepsilon)$
2	24	$\pi/2$	127	$16367173 \times 2^{+72}$	29.2
2	24	$\ln(2)$	127	8885060×2^{-11}	31.6
10	10	$\pi/2$	99	$8248251512 \times 10^{-6}$	11.7
10	10	$\pi/4$	99	$4124125756 \times 10^{-6}$	11.9
10	10	$\ln(10)$	99	$7908257897 \times 10^{+30}$	11.7
2	53	$\pi/2$	1023	$6381956970095103 \times 2^{+797}$	60.9
2	53	$\ln(2)$	1023	$5261692873635770 \times 2^{+499}$	66.8
2	113	$\pi/2$	1024	$614799 \cdots 1953734 \times 2^{+797}$	122.79

Minimax Approximation

Reminder. Let $g : [a, b] \to \mathbb{R}$, $||g||_{[a,b]} = \sup_{a \leq x \leq b} |g(x)|$.

We denote $\mathbb{R}_n[X] = \{p \in \mathbb{R}[X]; \deg p \leq n\}$.

Minimax Approximation

Reminder. Let $g : [a, b] \to \mathbb{R}$, $||g||_{[a,b]} = \sup_{a \leq x \leq b} |g(x)|$.

We denote $\mathbb{R}_n[X] = \{p \in \mathbb{R}[X]; \deg p \leq n\}$.

Minimax approximation: let $f : [a, b] \to \mathbb{R}$, $n \in \mathbb{N}$, we search for $p \in \mathbb{R}_n[X]$ s.t.

$$||p - f||_{[a,b]} = \inf_{q \in \mathbb{R}_n[X]} ||q - f||_{[a,b]}.$$

Minimax Approximation

Reminder. Let $g : [a, b] \to \mathbb{R}$, $||g||_{[a,b]} = \sup_{a \le x \le b} |g(x)|$.

We denote $\mathbb{R}_n[X] = \{p \in \mathbb{R}[X]; \deg p \le n\}$.

Minimax approximation: let $f : [a, b] \to \mathbb{R}$, $n \in \mathbb{N}$, we search for $p \in \mathbb{R}_n[X]$ s.t.

$$||p - f||_{[a,b]} = \inf_{q \in \mathbb{R}_n[X]} ||q - f||_{[a,b]}.$$

An algorithm due to Remez gives p (`minimax` function in Maple, also available in Sollya `http://sollya.gforge.inria.fr/`).

Minimax Approximation

Reminder. Let $g : [a, b] \to \mathbb{R}$, $||g||_{[a,b]} = \sup_{a \le x \le b} |g(x)|$.

We denote $\mathbb{R}_n[X] = \{p \in \mathbb{R}[X]; \deg p \le n\}$.

Minimax approximation: let $f : [a, b] \to \mathbb{R}$, $n \in \mathbb{N}$, we search for $p \in \mathbb{R}_n[X]$ s.t.

$$||p - f||_{[a,b]} = \inf_{q \in \mathbb{R}_n[X]} ||q - f||_{[a,b]}.$$

An algorithm due to Remez gives p (`minimax` function in Maple, also available in Sollya `http://sollya.gforge.inria.fr/`).

Problem: we can't directly use minimax approx. in a computer since the coefficients of p can't be represented on a finite number of bits.

Truncated Polynomials

Our context: the coefficients of the polynomials must be written on a finite (imposed) number of bits.

Truncated Polynomials

Our context: the coefficients of the polynomials must be written on a finite (imposed) number of bits.

Let $m = (m_i)_{0 \leq i \leq n}$ a finite sequence of rational integers. Let

$\mathcal{P}_n^m = \{q = q_0 + q_1 x + \cdots + q_n x^n \in \mathbb{R}_n[X]; q_i \text{ integer multiple of } 2^{-m_i}, \forall i\}.$

Truncated Polynomials

Our context: the coefficients of the polynomials must be written on a finite (imposed) number of bits.

Let $m = (m_i)_{0 \leq i \leq n}$ a finite sequence of rational integers. Let

$\mathcal{P}_n^m = \{q = q_0 + q_1 x + \cdots + q_n x^n \in \mathbb{R}_n[X]; q_i$ integer multiple of $2^{-m_i}, \forall i\}$.

Question: find $p^\star \in \mathcal{P}_n^m$ which minimizes $||f - q||$, $q \in \mathcal{P}_n^m$.

Truncated Polynomials

Our context: the coefficients of the polynomials must be written on a finite (imposed) number of bits.

Let $m = (m_i)_{0 \leq i \leq n}$ a finite sequence of rational integers. Let

$\mathcal{P}_n^m = \{q = q_0 + q_1 x + \cdots + q_n x^n \in \mathbb{R}_n[X]; q_i$ integer multiple of $2^{-m_i}, \forall i\}$.

Question: find $p^\star \in \mathcal{P}_n^m$ which minimizes $||f - q||$, $q \in \mathcal{P}_n^m$.

First idea. Remez $\longrightarrow p(x) = p_0 + p_1 x + \cdots + p_n x^n$. Every p_i rounded to $\hat{a}_i / 2^{m_i}$, the nearest integer multiple of 2^{-m_i} $\longrightarrow$
$\hat{p}(x) = \frac{\hat{a}_0}{2^{m_0}} + \frac{\hat{a}_1}{2^{m_1}} x + \cdots + \frac{\hat{a}_n}{2^{m_n}} x^n$.

Truncated Polynomials

Our context: the coefficients of the polynomials must be written on a finite (imposed) number of bits.

Let $m = (m_i)_{0 \leq i \leq n}$ a finite sequence of rational integers. Let

$\mathcal{P}_n^m = \{q = q_0 + q_1 x + \cdots + q_n x^n \in \mathbb{R}_n[X]; q_i$ integer multiple of $2^{-m_i}, \forall i\}$.

Question: find $p^\star \in \mathcal{P}_n^m$ which minimizes $||f - q||$, $q \in \mathcal{P}_n^m$.

First idea. Remez $\longrightarrow p(x) = p_0 + p_1 x + \cdots + p_n x^n$. Every p_i rounded to $\hat{a}_i / 2^{m_i}$, the nearest integer multiple of $2^{-m_i} \longrightarrow$
$\hat{p}(x) = \dfrac{\hat{a}_0}{2^{m_0}} + \dfrac{\hat{a}_1}{2^{m_1}} x + \cdots + \dfrac{\hat{a}_n}{2^{m_n}} x^n$.

Problem: $\hat{p}$ not necessarily a minimax approx. of f among the polynomials of $\mathcal{P}_n^m$.

Approximation of the Function $\cos$ over $[0, \pi/4]$ by a Degree-3 Polynomial

Maple or Sollya tell us that the polynomial

$$p = 0.9998864206 + 0.00469021603x - 0.5303088665x^2 + 0.06304636099x^3$$

is $\sim$ the best approximant to $\cos$. We have
$\varepsilon = ||\cos - p||_{[0,\pi/4]} = 0.0001135879...$.

We look for $a_0, a_1, a_2, a_3 \in \mathbb{Z}$ such that

$$\max_{0 \le x \le \pi/4} \left| \cos x - \left(\frac{a_0}{2^{12}} + \frac{a_1}{2^{10}} x + \frac{a_2}{2^6} x^2 + \frac{a_3}{2^4} x^3 \right) \right|$$

is minimal.

Approximation of the Function $\cos$ over $[0, \pi/4]$ by a Degree-3 Polynomial

Maple or Sollya tell us that the polynomial

$$p = 0.9998864206 + 0.00469021603x - 0.5303088665x^2 + 0.06304636099x^3$$

is $\sim$ the best approximant to $\cos$. We have
$\varepsilon = ||\cos - p||_{[0,\pi/4]} = 0.0001135879...$.

We look for $a_0, a_1, a_2, a_3 \in \mathbb{Z}$ such that

$$\max_{0 \le x \le \pi/4} \left| \cos x - \left(\frac{a_0}{2^{12}} + \frac{a_1}{2^{10}} x + \frac{a_2}{2^6} x^2 + \frac{a_3}{2^4} x^3 \right) \right|$$

is minimal.

The naive approach gives the polynomial

$$\hat{p} = \frac{2^{12}}{2^{12}} + \frac{5}{2^{10}} x - \frac{34}{2^6} x^2 + \frac{1}{2^4} x^3.$$

We have $\hat{\varepsilon} = ||\cos - \hat{p}||_{[0,\pi/4]} = 0.00069397...$.

Approximation of the Function $\cos$ over $[0, \pi/4]$ by a Degree-3 Polynomial

Maple or Sollya computes a polynomial p which is $\sim$ the best approximant to $\cos$. We have $\varepsilon = ||\cos - p||_{[0,\pi/4]} = 0.0001135879...$.
We look for $a_0, a_1, a_2, a_3 \in \mathbb{Z}$ such that

$$\max_{0 \leq x \leq \pi/4} \left| \cos x - \left(\frac{a_0}{2^{12}} + \frac{a_1}{2^{10}} x + \frac{a_2}{2^6} x^2 + \frac{a_3}{2^4} x^3 \right) \right|$$

is minimal.
The naive approach gives the polynomial $\hat{p}$ and
$\hat{\varepsilon} = ||\cos - \hat{p}||_{[0,\pi/4]} = 0.00069397...$

Approximation of the Function $\cos$ over $[0, \pi/4]$ by a Degree-3 Polynomial

Maple or Sollya computes a polynomial p which is $\sim$ the best approximant to $\cos$. We have $\varepsilon = ||\cos - p||_{[0,\pi/4]} = 0.0001135879\ldots$.
We look for $a_0, a_1, a_2, a_3 \in \mathbb{Z}$ such that

$$\max_{0 \le x \le \pi/4} \left| \cos x - \left(\frac{a_0}{2^{12}} + \frac{a_1}{2^{10}} x + \frac{a_2}{2^6} x^2 + \frac{a_3}{2^4} x^3 \right) \right|$$

is minimal.
The naive approach gives the polynomial $\hat{p}$ and $\hat{\varepsilon} = ||\cos - \hat{p}||_{[0,\pi/4]} = 0.00069397\ldots$ But the best "truncated" approximant:

$$p^\star = \frac{4095}{2^{12}} + \frac{6}{2^{10}}\, x - \frac{34}{2^6}\, x^2 + \frac{1}{2^4}\, x^3$$

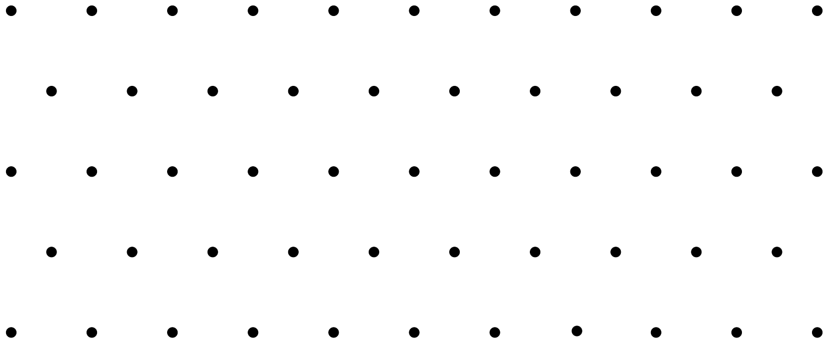
which gives $||\cos - p^\star||_{[0,\pi/4]} = 0.0002441406250$.

In this example, we gain $-\log_2(0.35) \approx 1.5$ bits of accuracy.

A First Approach through Linear Programming

It makes it possible to tackle with degree-8 or 10 polynomials: this is nice for hardware-oriented applications but not satisfying for all software-oriented applications.

A Second Approach through Lattice Basis Reduction



A Second Approach through Lattice Basis Reduction

Definition

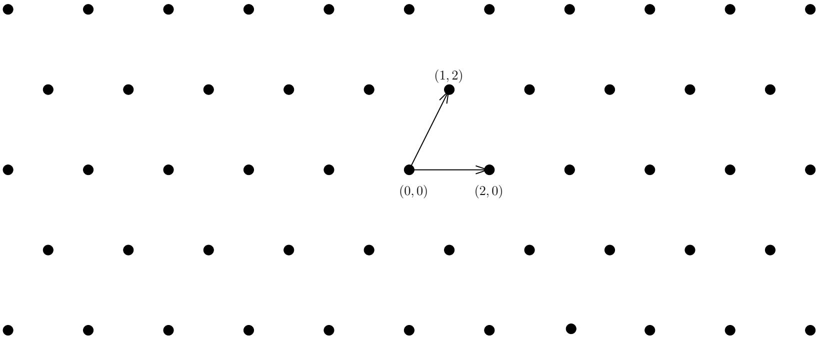
Let L be a nonempty subset of $\mathbb{R}^d$, L is a lattice iff there exists a set of vectors $b_1, \ldots, b_k$ $\mathbb{R}$-linearly independent such that

$$L = \mathbb{Z}.b_1 \oplus \cdots \oplus \mathbb{Z}.b_k.$$

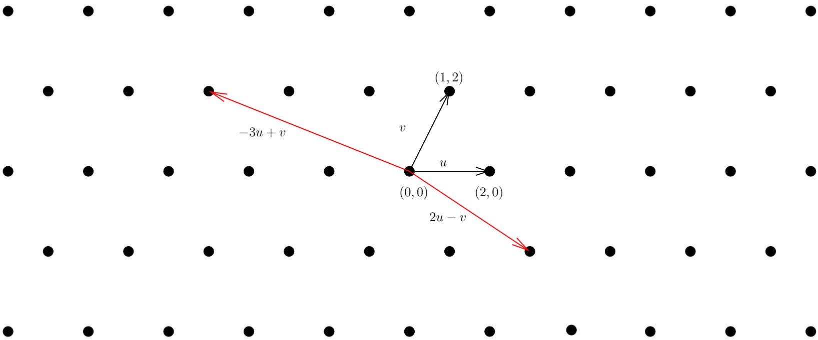
$(b_1, \ldots, b_k)$ is a basis of the lattice L.

Examples. $\mathbb{Z}^d$, every subgroup of $\mathbb{Z}^d$.

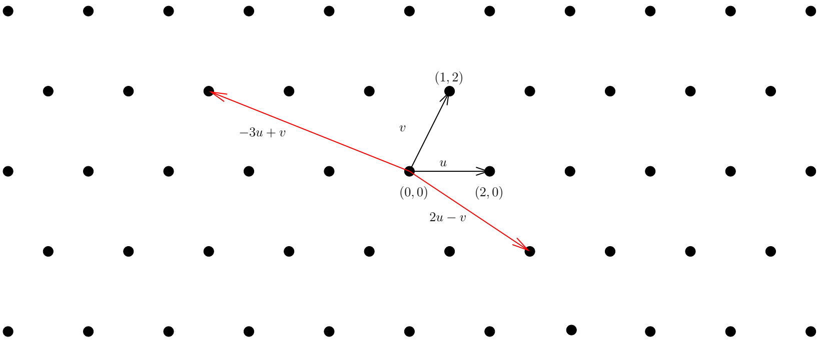
Example: The Lattice $\mathbb{Z}(2,0) \oplus \mathbb{Z}(1,2)$



Example: The Lattice $\mathbb{Z}(2,0) \oplus \mathbb{Z}(1,2)$

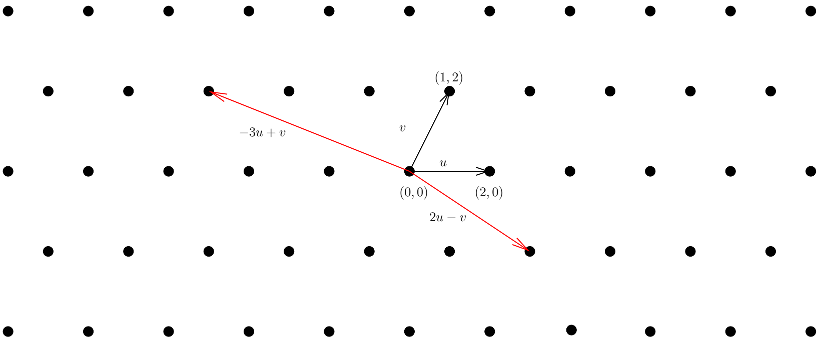


Example: The Lattice $\mathbb{Z}(2,0) \oplus \mathbb{Z}(1,2)$



SVP (Shortest Vector Problem) and CVP (Closest Vector Problem)

Example: The Lattice $\mathbb{Z}(2,0) \oplus \mathbb{Z}(1,2)$



SVP (Shortest Vector Problem) and CVP (Closest Vector Problem) are NP-hard.

Lenstra-Lenstra-Lovász Algorithm

SVP (Shortest Vector Problem) and CVP (Closest Vector Problem) are NP-hard.
Factoring Polynomials with Rational Coefficients, A. K. Lenstra, H. W. Lenstra and L. Lovász, Math. Annalen **261**, 515-534, 1982.

The LLL algorithm gives an approximate solution to SVP in polynomial time.

Babai's algorithm (based on LLL) gives an approximate solution to CVP in polynomial time.

Absolute Error Problem

We search for (one of the) best(s) polynomial of the form

$$p^\star = \frac{a_0^\star}{2^{m_0}} + \frac{a_1^\star}{2^{m_1}} X + \cdots + \frac{a_n^\star}{2^{m_n}} X^n$$

(where $a_i^\star \in \mathbb{Z}$ and $m_i \in \mathbb{Z}$) that minimizes $\|f - p\|_{[a,\, b]}$.

Discretize the continuous problem: we choose $x_1, \cdots, x_d$ points in $[a,\, b]$ such that $\frac{a_0^\star}{2^{m_0}} + \frac{a_1^\star}{2^{m_1}} x_i + \cdots + \frac{a_n^\star}{2^{m_n}} x_i^n$ as close as possible to $f(x_i)$ for all $i = 1, \ldots, d$.

That is to say we want the vectors

$$\begin{pmatrix} \frac{a_0^\star}{2^{m_0}} + \frac{a_1^\star}{2^{m_1}} x_1 + \cdots + \frac{a_n^\star}{2^{m_n}} x_1^n \\ \frac{a_0^\star}{2^{m_0}} + \frac{a_1^\star}{2^{m_1}} x_2 + \cdots + \frac{a_n^\star}{2^{m_n}} x_2^n \\ \vdots \\ \frac{a_0^\star}{2^{m_0}} + \frac{a_1^\star}{2^{m_1}} x_d + \cdots + \frac{a_n^\star}{2^{m_n}} x_d^n \end{pmatrix} \text{ and } \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_d) \end{pmatrix}$$

to be as close as possible, which can be rewritten as: we want the vectors

$$a_0^\star \underbrace{\begin{pmatrix} \frac{1}{2^{m_0}} \\ \frac{1}{2^{m_0}} \\ \vdots \\ \frac{1}{2^{m_0}} \end{pmatrix}}_{\overrightarrow{v_0}} + a_1^\star \underbrace{\begin{pmatrix} \frac{x_1}{2^{m_1}} \\ \frac{x_2}{2^{m_1}} \\ \vdots \\ \frac{x_d}{2^{m_1}} \end{pmatrix}}_{\overrightarrow{v_1}} + \cdots + a_n^\star \underbrace{\begin{pmatrix} \frac{x_1^n}{2^{m_n}} \\ \frac{x_2^n}{2^{m_n}} \\ \vdots \\ \frac{x_d^n}{2^{m_n}} \end{pmatrix}}_{\overrightarrow{v_n}} \text{ and } \underbrace{\begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_d) \end{pmatrix}}_{\overrightarrow{y}}$$

to be as close as possible.
We have to minimize $\|a_0^\star \overrightarrow{v_0} + \cdots + a_n^\star \overrightarrow{v_n} - \overrightarrow{y}\|$.

We have to minimize $\|a_0^\star\overrightarrow{v_0} + \cdots + a_n^\star\overrightarrow{v_n} - \overrightarrow{y}\|$.

This is a closest vector problem in a lattice!

It is NP-hard: LLL algorithm gives an approximate solution.

Summary

We can use the method we developed in two directions

- it is able to give us a smaller (in term of degree and/or size of the coefficients) polynomial providing the same accuracy.
- we can also use it to find a much better polynomial (in term of accuracy) with same precision for the coefficients than the rounded minimax.

We illustrate the second item with an example taken from CRLibm.

An Example from CRlibm

- CRlibm is a library designed to compute correctly rounded functions in an efficient way (target : IEEE double precision).

  `http://lipforge.ens-lyon.fr/www/crlibm/`

- It uses specific formats such as double-double or triple-double.

- Here is an example we worked on with C. Lauter, and which is used to compute $\arcsin(x)$ on $[0.79; 1]$.

Arcsine Function

After argument reduction we have the problem to approximate

$$g(z) = \frac{\arcsin(1 - (z + m)) - \frac{\pi}{2}}{\sqrt{2 \cdot (z + m)}}$$

where $\mathrm{0xBFBC28F800009107} \leq z \leq \mathrm{0x3FBC28F7FFFF6EF1}$ (i.e. approximately $-0.110 \leq z \leq 0.110$) and $m = \mathrm{0x3FBC28F80000910F} \simeq 0.110$.

Data

Target accuracy to achieve correct rounding : 2^{-119}.
The minimax of degree 21 is sufficient (error = $2^{-119.83}$).
Each approximant is of the form

$$\underbrace{p_0}_{t.d.} + \underbrace{p_1}_{t.d.} x + \underbrace{p_2}_{d.d.} x^2 + \underbrace{\cdots}_{\cdots} + \underbrace{p_9}_{d.d.} x^9 + \underbrace{p_{10}}_{d.} x^{10} + \underbrace{\cdots}_{\cdots} + \underbrace{p_{21}}_{d.} x^{21}$$

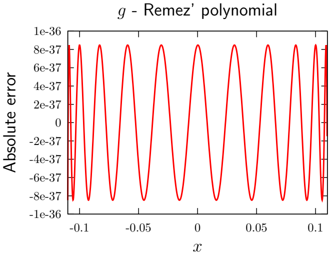
where the p_i are either double precision numbers (d.), a sum of two double precision numbers (d.d.), a sum of two double precision numbers (t.d.).

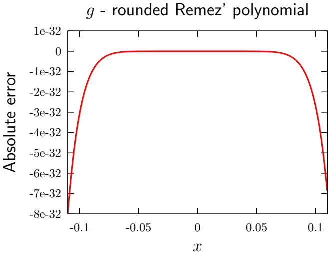
Figure: binary logarithm of the absolute error of several approximants

Target	-119
Minimax	-119.83
Rounded minimax	-103.31
Our polynomial	-119.77

Exact Minimax, Rounded Minimax, our Polynomial

We save 16 bits with our method.

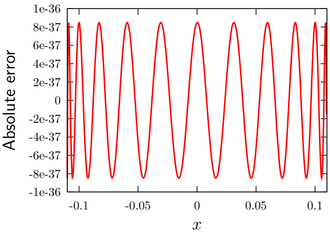




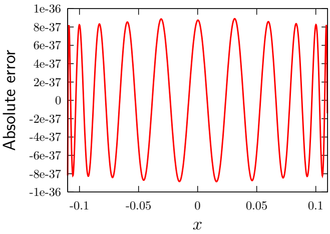
Exact Minimax, Rounded Minimax, our Polynomial

We save 16 bits with our method.

g - Remez' polynomial



g - our polynomial



Machine-Efficient Polynomial Approximation: Summary

- Two methods which improve the results provided by existing Remez' based method.
  The first method, based on linear programming, gives a best polynomial possible (for a given sequence of m_i).
  The second method, based on lattice basis reduction, much faster and more efficient than the first one, gives a very good approximant.
  We use linear programming to show that the error provided by this approach is tight.
  All these tools are or shall be part of the Sollya software `http://sollya.gforge.inria.fr/`. Sollya is a tool environment for safe floating-point code development.

- Can be adapted to several kind of coefficients (fixed-point format, multi-double, classical floating point arithmetic with several precision formats).

Evaluation of Elementary Functions

$\exp, \ln, \cos, \sin, \arctan, \sqrt{\ }, \ldots$

Goal: evaluation of φ to a given accuracy η.

- Step 0. Computation of hardest-to-round cases: V. Lefèvre and J.-M. Muller.
- Step 1. Argument reduction (Payne & Hanek, Ng, Daumas *et al*): evaluation of a function φ over $\mathbb{R}$ or a subset of $\mathbb{R}$ is reduced to the evaluation of a function f over $[a, b]$.
- Step 2. Computation of $p^\star$, a "machine-efficient" polynomial approximation of f .
- Step 3. Computation of a rigorous approximation error $||f - p^\star||$.
- Step 4. Computation of a certified evalutation error of $p^\star$: GAPPA (G. Melquiond).

Evaluation of Elementary Functions

$\exp, \ln, \cos, \sin, \arctan, \sqrt{\ }, \ldots$

Goal: evaluation of φ to a given accuracy η.

- Step 1. Argument reduction (Payne & Hanek, Ng, Daumas *et al*): evaluation of a function φ over $\mathbb{R}$ or a subset of $\mathbb{R}$ is reduced to the evaluation of a function f over $[a, b]$.
- Step 2. Computation of $p^\star$, a "machine-efficient" polynomial approximation of f .
- Step 3. Computation of a rigorous approximation error $||f - p^\star||$.
- Step 4. Computation of a certified evalutation error of $p^\star$: GAPPA (G. Melquiond).

Evaluation of Elementary Functions

$\exp, \ln, \cos, \sin, \arctan, \sqrt{\ }, \ldots$

Goal: evaluation of φ to a given accuracy η.

- Step 0. Computation of hardest-to-round cases: V. Lefèvre and J.-M. Muller.
- Step 1. Argument reduction (Payne & Hanek, Ng, Daumas *et al*): evaluation of a function φ over $\mathbb{R}$ or a subset of $\mathbb{R}$ is reduced to the evaluation of a function f over $[a, b]$.
- Step 2. Computation of $p^\star$, a "machine-efficient" polynomial approximation of f .
- Step 3. Computation of a rigorous approximation error $||f - p^\star||$.
- Step 4. Computation of a certified evalutation error of $p^\star$: GAPPA (G. Melquiond).

Correct rounding

In the IEEE 754 standard, the user defines an *active rounding mode* (or *rounding direction attribute*) among:

- round to the nearest (default) in case of a tie, value whose integral significand is even;
- round towards $+\infty$.
- round towards $-\infty$.
- round towards zero.

A correctly-rounded operation whose entries are FP numbers must return what we would get by infinitely precise operation followed by rounding.

Correct rounding

IEEE-754 (1985): Correct rounding for $+$, $-$, $\times$, $\div$, $\sqrt{}$ and some conversions. Advantages:

- if the result of an operation is exactly representable, we get it;
- if we just use the 4 arith. operations and $\sqrt{}$, deterministic arithmetic: one can elaborate algorithms and proofs that use the specifications;
- accuracy and portability are improved;
- playing with rounding towards $+\infty$ and $-\infty$ $\rightarrow$ certain lower and upper bounds.

FP arithmetic becomes a structure in itself, that can be studied.
IEEE-754 (2008): suggests correct rounding for some elementary functions.

The Table Maker's Dilemma

Consider the binary64/double precision FP number (base $2, p = 53$)

$$x = \frac{8520761231538509}{2^{62}}$$

We have

$$2^{53+x} = 9018742077413030.{\color{blue}99999999999999999}8805240837303 \cdots$$

The Table Maker's Dilemma

Consider the binary64/double precision FP number (base $2, p = 53$)

$$x = \frac{8520761231538509}{2^{62}}$$

We have

$$2^{53+x} = 9018742077413030.99999999999999999998805240837303\cdots$$

So what ?

The Table Maker's Dilemma

Consider the binary64/double precision FP number (base $2, p = 53$)

$$x = \frac{8520761231538509}{2^{62}}$$

We have

$$2^{53+x} = 9018742077413030.999999999999999999988052408373 03\cdots$$

So what ?
Hardest-to-round (HR) case for function 2^x and double precision FP numbers.

The Table Maker's Dilemma

Consider the binary64/double precision FP number (base $2, p = 53$)

$$x = \frac{8520761231538509}{2^{62}}$$

We have

$$2^{53+x} = 9018742077413030.99999999999999999 8805240837303 \cdots$$

So what ?

Hardest-to-round (HR) case for function 2^x and double precision FP numbers.

Function f: sin, cos, arcsin, arccos, tan, arctan, exp, log, sinh, cosh,

Evaluation of Elementary Functions

$\exp, \ln, \cos, \sin, \arctan, \sqrt{\ }, \ldots$

Goal: evaluation of φ to a given accuracy η.

- Step 1. Argument reduction (Payne & Hanek, Ng, Daumas *et al*): evaluation of a function φ over $\mathbb{R}$ or a subset of $\mathbb{R}$ is reduced to the evaluation of a function f over $[a, b]$.
- Step 2. Computation of $p^\star$, a "machine-efficient" polynomial approximation of f .
- Step 3. Computation of a rigorous approximation error $||f - p^\star||$.
- Step 4. Computation of a certified evalutation error of $p^\star$: GAPPA (G. Melquiond).

Evaluation of Elementary Functions

$\exp, \ln, \cos, \sin, \arctan, \sqrt{\ }, \ldots$

Goal: evaluation of φ to a given accuracy η.

- Step 0. Computation of hardest-to-round cases: V. Lefèvre and J.-M. Muller.
- Step 1. Argument reduction (Payne & Hanek, Ng, Daumas *et al*): evaluation of a function φ over $\mathbb{R}$ or a subset of $\mathbb{R}$ is reduced to the evaluation of a function f over $[a, b]$.
- Step 2. Computation of $p^\star$, a "machine-efficient" polynomial approximation of f .
- Step 3. Computation of a rigorous approximation error $||f - p^\star||$.
- Step 4. Computation of a certified evalutation error of $p^\star$: GAPPA (G. Melquiond).

Some references

- *An Introduction to Diophantine Approximation*, J.W.S. Cassels, Cambridge University Press.
- *Continued Fractions*, A.Y. Khinchin, Chicago University Press.
- *Éléments de théorie des nombres*, Roger Descombes, PUF.
- *Modern Computer Algebra*, J. Gerhard and J. von zur Gathen, Cambridge University Press.
- *Elementary Functions, Algorithms and Implementation*, J.-M. Muller, Birkhaüser.
- *Handbook of Floating-Point Arithmetic*, J.-M. Muller *et al.*, Birkhaüser.