Continued Fractions and Double-Word Arithmetic

Jean-Michel Muller

CNRS - Laboratoire LIP

http://perso.ens-lyon.fr/jean-michel.muller/

Continued fractions

- Basic question: how close can a rational number of denominator $\leq q$ be to some real number α ?
- example of application to FP arithmetic: in a given FP system (base β, precision p, extremal exponents $e_{\min}$ and $e_{\max}$), how close can a FP number be to an integer multiple of $\pi/2$? Would give answers to:
  - can the tangent of a FP number overflow?
  - can the sine, cosine, tangent of a normal FP number be less than $\beta^{e_{\min}}$?
  - range reduction for implementing trigonometric functions: preliminary calculation of $y = x \mod 2\pi$ (so that the problem is reduced to approximating the function in $[0, 2\pi)$). With which accuracy must that calculation be done ?

What happens if range reduction is overlooked (Ng)

System	$\sin\left(10^{22}\right)$
exact result	$-0.8522008497671888017727\cdots$
HP 48 GX	-0.852200849762
HP 700	0.0
HP 375, 425t (4.3 BSD)	$-0.65365288\cdots$
matlab V.4.2 c.1 for Macintosh	0.8740
matlab V.4.2 c.1 for SPARC	-0.8522
Silicon Graphics Indy	$0.87402806\cdots$
SPARC	-0.85220084976718879
IBM RS/6000 AIX 3005	$-0.852200849\cdots$
DECstation 3100	NaN
Casio fx-8100, fx180p, fx 6910 G	Error

Until 2008, no standard for the elementary functions.

What happens if range reduction is overlooked (Ng)

System	$\sin\left(10^{22}\right)$
exact result	$-0.8522008497671888017727\cdots$
HP 48 GX	-0.852200849762
HP 700	0.0
HP 375, 425t (4.3 BSD)	$-0.65365288\cdots$
matlab V.4.2 c.1 for Macintosh	0.8740
matlab V.4.2 c.1 for SPARC	-0.8522
Silicon Graphics Indy	$0.87402806\cdots$
SPARC	-0.8522008497671879
IBM RS/6000 AIX 3005	$-0.852200849\cdots$
DECstation 3100	NaN
Casio fx-8100, fx180p, fx 6910 G	Error

Until 2008, no standard for the elementary functions.

What happens if range reduction is overlooked (Ng)

System	$\sin\left(10^{22}\right)$
exact result	$-0.8522008497671888017727\cdots$
HP 48 GX	-0.852200849762
HP 700	0.0
HP 375, 425t (4.3 BSD)	$-0.65365288\cdots$
matlab V.4.2 c.1 for Macintosh	0.8740
matlab V.4.2 c.1 for SPARC	-0.8522
Silicon Graphics Indy	$0.87402806\cdots$
SPARC	-0.85220084976718879
IBM RS/6000 AIX 3005	$-0.852200849\cdots$
DECstation 3100	NaN
Casio fx-8100, fx180p, fx 6910 G	Error

Until 2008, no standard for the elementary functions.

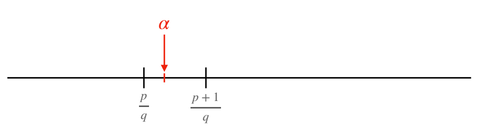
What happens if range reduction is overlooked (Ng)

System	$\sin\left(10^{22}\right)$
exact result	$-0.8522008497671888017727\cdots$
HP 48 GX	-0.852200849762
HP 700	0.0
HP 375, 425t (4.3 BSD)	$-0.65365288\cdots$
matlab V.4.2 c.1 for Macintosh	0.8740
matlab V.4.2 c.1 for SPARC	-0.8522
Silicon Graphics Indy	$0.87402806\cdots$
SPARC	-0.85220084976718879
IBM RS/6000 AIX 3005	$-0.852200849\cdots$
DECstation 3100	NaN
Casio fx-8100, fx180p, fx 6910 G	Error

Until 2008, no requirement for the elementary functions.

Continued fractions

- any real number α can be approximated as closely as desired by rationals... however, size of these rationals?
- intuitively, a fraction of denominator q can approximate α with accuracy better than $1/(2q)$:



- can we do significantly better ? Given α and q_{max}, what is the fraction of denominator $< q_{\text{max}}$ that best approximates α?

Continued fractions

$$
\begin{array}{rrcl}
 & a_0 & = & \lfloor \alpha \rfloor \\
\text{if } a_0 \neq \alpha & r_1 & = & \frac{1}{\alpha - a_0}
\end{array}
$$

$$\alpha = a_0 + \frac{1}{r_1}$$

approximation $\alpha \approx a_0$

$$
\begin{array}{rrcl}
 & a_1 & = & \lfloor r_1 \rfloor \\
\text{if } a_1 \neq r_1 & r_2 & = & \frac{1}{r_1 - a_1}
\end{array}
$$

$$\alpha = a_0 + \frac{1}{a_1 + \frac{1}{r_2}}$$

approximation $\alpha \approx a_0 + \frac{1}{a_1}$

$$
\begin{array}{rrcl}
 & a_2 & = & \lfloor r_2 \rfloor \\
\text{if } a_2 \neq r_2 & r_3 & = & \frac{1}{r_2 - a_2}
\end{array}
$$

$$\alpha = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{r_3}}}$$

approximation $\alpha \approx a_0 + \frac{1}{a_1 + \frac{1}{a_2}}$

Continued fractions

The sequence

$$\begin{cases} & r_0 & = & \alpha \\ & a_i & = & \lfloor r_i \rfloor \\ \text{if } r_i \neq a_i & r_{i+1} & = & \frac{1}{r_i - a_i} \end{cases}$$

Gives

$$\alpha = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{\ddots + \cfrac{1}{r_i}}}}}$$

and the rational approximation

$$\alpha \approx \frac{P_i}{Q_i} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{\ddots + \cfrac{1}{a_i}}}}}$$

Continued fractions

- P_i/Q_i is called the i^{th} convergent of α (french word: réduite);
- the sequence $(a_0, a_1, a_2, \ldots)$ is called the continued fraction expansion of α. It is finite iff $\alpha \in \mathbb{Q}$;

Exercise: give the continued fraction expansion of $\sqrt{2}$.

- we can choose (up to multiplication of numerator & denominator by the same factor):

$$\begin{array}{rclcrcl} P_0 & = & a_0 & \quad & Q_0 & = & 1 \\ P_1 & = & a_1 a_0 + 1 & \quad & Q_1 & = & a_1 \end{array}$$

$$\begin{aligned} \tfrac{P_2}{Q_2} &= a_0 + \tfrac{1}{a_1 + \frac{1}{a_2}} \\ &= a_0 + \tfrac{a_2}{a_1 a_2 + 1} \\ &= \tfrac{a_0 a_1 a_2 + a_0 + a_2}{a_1 a_2 + 1} \end{aligned}$$

$\rightarrow$ $P_2 = P_1 a_2 + P_0$ and $Q_2 = Q_1 a_2 + Q_0$.

Computation of the convergents

Lemma 1

One can choose (still defined up to multiplication by same factor):

$$\begin{cases} P_n &= P_{n-1}a_n + P_{n-2} \\ Q_n &= Q_{n-1}a_n + Q_{n-2}. \end{cases} \tag{1}$$

Proof:

- true for $n = 2$ (previous slide);
- assume true for n. We get P_{n+1}/Q_{n+1} from P_n/Q_n by replacing a_n by $a_n + \frac{1}{a_{n+1}}$;
- let us do that replacement in (1), multiplying both terms by a_{n+1} to keep integers.

Computation of the convergents

Given
$$\begin{cases} P_{n+1} = \left[P_{n-1}\left(a_n + \dfrac{1}{a_{n+1}}\right) + P_{n-2} \right] \times a_{n+1} \\ Q_{n+1} = \left[Q_{n-1}\left(a_n + \dfrac{1}{a_{n+1}}\right) + Q_{n-2} \right] \times a_{n+1} \end{cases}$$

a_n does not appear in these terms

$$\longrightarrow \quad P_{n+1} = [\underbrace{P_{n-1} a_n + P_{n-2}}_{P_n}] a_{n+1} + P_{n-1}$$

$$Q_{n+1} = [\underbrace{Q_{n-1} a_n + Q_{n-2}}_{Q_n}] a_{n+1} + Q_{n-1}$$

A miraculous lemma

With the above-defined formulas for P_n and Q_n.

Lemma 2

$$P_n Q_{n-1} - P_{n-1} Q_n = (-1)^{n+1}.$$

$P_n = P_{n-1} a_n + P_{n-2} \longrightarrow P_n Q_{n-1} = P_{n-1} Q_{n-1} a_n + P_{n-2} Q_{n-1}$

$Q_n = Q_{n-1} a_n + Q_{n-2} \longrightarrow P_{n-1} Q_n = P_{n-1} Q_{n-1} a_n + P_{n-1} Q_{n-2}$

<u>Consequence</u> $P_n Q_{n-1} - P_{n-1} Q_n = -\left[P_{n-1} Q_{n-2} - P_{n-2} Q_{n-1}\right]$

<u>From</u> $P_0 = a_0$; $Q_0 = 1$; $P_1 = a_1 a_0 + 1$; $Q_1 = a_1$

we deduce $P_1 Q_0 - P_0 Q_1 = 1$

Hence $P_n Q_{n-1} - P_{n-1} Q_n = (-1)^{n+1}$

Why is the lemma miraculous?

- Bezout Theorem $\to \gcd(P_n, Q_n) = 1 \to$ the formulas give irreducible fractions;
- the lemma can be written (with substitution $n \to n+1$):

$$\frac{P_{n+1}}{Q_{n+1}} - \frac{P_n}{Q_n} = \frac{(-1)^n}{Q_n Q_{n+1}}$$

- α deduced from P_{n+1}/Q_{n+1} by replacing a_{n+1} by r_{n+1} $\to$ gives

$$\alpha = \frac{P_n r_{n+1} + P_{n-1}}{Q_n r_{n+1} + Q_{n-1}}.$$

  Function $x \to (P_n x + P_{n-1})/(Q_n x + Q_{n-1})$ is monotone (derivative $(P_n Q_{n-1} - Q_n P_{n-1})/(Q_n x + Q_{n-1})^2 = (-1)^{n+1}/(Q_n x + Q_{n-1})^2) \to \alpha$ is between $\frac{P_{n-1}}{Q_{n-1}}$ and $\frac{P_n}{Q_n}$.

Hence

$$\left|\alpha - \frac{P_n}{Q_n}\right| \le \left|\frac{P_{n+1}}{Q_{n+1}} - \frac{P_n}{Q_n}\right| = \frac{1}{Q_n Q_{n+1}} \le \frac{1}{Q_n^2}.$$

Consequences

- if for some n, $a_n = 0$ then $\alpha = \frac{P_n}{Q_n}$ and the sequence ends;
- otherwise, $\forall n,\ a_n \geq 1$,so that from $Q_n = Q_{n-1}a_n + Q_{n-2}$ we deduce $Q_n > Q_{n-1}$ so that $Q_n > 2Q_{n-2}$, hence the bound

$$\frac{1}{Q_n Q_{n+1}} < \frac{1}{2Q_n Q_{n-1}} < \frac{1}{4Q_{n-1}Q_{n-2}} < \dots$$

goes to zero faster than $1/2^n$.

Theorem 3

$P_n/Q_n \to \alpha$.We also have $P_n/Q_n \leq \alpha$ when n is even, and $P_n/Q_n \geq \alpha$ when n is odd,

Consequences

We write:

$$\alpha = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{\ddots}}}}$$

or

$$\alpha = [a_0; a_1, a_2, a_3, a_4, \cdots].$$

A few examples

$$\sqrt{2} = [1; 2, 2, 2, 2, \ldots]$$
$$\frac{1+\sqrt{5}}{2} = [1; 1, 1, 1, 1, \ldots]$$
$$\pi = [3; 7, 15, 1, 292, 1, 1, 1, 2, 1, 3, 1, 14, \ldots]$$

which gives the following convergents ($\to$ very good rational approximations to π):

$$3; \quad \frac{22}{7}; \quad \frac{333}{106}; \quad \frac{355}{113}; \quad \frac{103993}{33102}; \ldots$$
$$e = [2; 1, 2, 1, 1, 4, 1, 1, 6, 1, 1, 8, 1, 1, 10, 1, 1, 12, \ldots]$$

Theorem 4 (Lagrange)

The continued fraction expansion of α is ultimately periodic iff α is a root of a degree-2 polynomial with integer coefficients.

Continued fractions are "best" rational approximations

Theorem 5

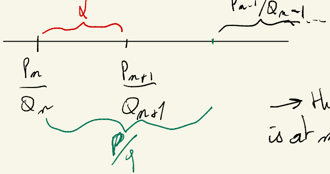
Let (P_n/Q_n) be the n^{th} convergent to α. If an irreducible fraction p/q is a better approximation to α than P_n/Q_n then $q > Q_n$.

a) If $\frac{p}{q}$ is between $\frac{P_n}{Q_n}$ and $\frac{P_{n+1}}{Q_{n+1}}$, then

$$\left|\frac{P_n}{Q_n} - \frac{p}{q}\right| = \frac{N}{Q_n q} < \left|\frac{P_{n+1}}{Q_{n+1}} - \frac{P_n}{Q_n}\right| = \frac{1}{Q_n Q_{n+1}}$$

$$\longrightarrow \frac{N}{q} < \frac{1}{Q_{n+1}} \longrightarrow q > Q_{n+1}$$

b) If p/q is <u>not</u> between $\frac{P_n}{Q_n}$ and $\frac{P_{n+1}}{Q_{n+1}}$



$$\frac{1}{Q_n Q_{n+1}} < \frac{1}{2Q_{n-1}Q_n}$$

$\rightarrow$ the length of the intervall $\left(\frac{P_n}{Q_n}, \frac{P_{n+1}}{Q_{n+1}}\right)$ is at most half the length of $\left(\frac{P_{n-1}}{Q_{n-1}}, \frac{P_n}{Q_n}\right)$

$\longrightarrow$ $\frac{p}{q}$ is in the interval $\left(\frac{P_{n-1}}{Q_{n-1}}, \frac{P_n}{Q_n}\right)$

$\longrightarrow$ the case a) applies with $\frac{P_{n-1}}{Q_{n-1}} \Rightarrow q > Q_n$

Theorem 6

Let (P_n/Q_n) be the convergents to α. For any $(p, q) \in \mathbb{N} \times \mathbb{N}^*$ with $q < Q_{n+1}$, we have

$$|p - \alpha q| \geq |P_n - \alpha Q_n| .$$

Theorem 7

Let $(p, q) \in \mathbb{N} \times \mathbb{N}^*$. If $\left|\frac{p}{q} - \alpha\right| < \frac{1}{2q^2}$ then p/q is one of the convergents to α.

How close can a FP number be to a nonzero multiple of an irrational number C?

- We look for a normal FP number

$$x = M_x \cdot \beta^{e_x - p + 1},$$

with $\beta^{p-1} \leq M_x \leq \beta^p - 1$, as close as possible to a multiple of C.

- e_x assumed fixed (new analysis for each value of the exponent).

$$M_x \cdot \beta^{e_x - p + 1} = k_x \cdot C + \epsilon_x, \quad \text{with } k_x = \left\lfloor M_x \cdot \beta^{e_x - p + 1} / C \right\rceil,$$

smallest possible value of $|\epsilon_x|$?

- let (P_i/Q_i) be the sequence of the convergents to $\beta^{e_x - p + 1}/C$;
- Integer j: largest such that $Q_j \leq \beta^p - 1$.

How close can a FP number be to a nonzero multiple of an irrational number C?

Theorem 6 $\Rightarrow$ for any (k_x, M_x) with $M_x \leq \beta^p - 1 < Q_{j+1}$ we have

$$\left| k_x - \frac{\beta^{e_x - p + 1}}{C} \cdot M_x \right| \geq \left| P_j - \frac{\beta^{e_x - p + 1}}{C} \cdot Q_j \right]$$

Gives

$$\underbrace{\left| k_x \cdot C - \beta^{e_x - p + 1} M_x \right|}_{|\epsilon_x|} \geq \left| P_j \cdot C - \beta^{e_x - p + 1} Q_j \right|$$

Solution: for each value of e_x between $\lceil \log_\beta(C) \rceil$ and $e_{\max}$, compute the corresponding P_j / Q_j, the lowest value of $|\epsilon_x|$ for that e_x will be attained for $M_x = Q_j$.

```
worstcaseRR := proc(B,p,emin,emax,C,ndigits)
  local epsilonmin,powerofBoverC,e,a,Plast,r,Qlast, Q,P,NewQ,NewP,epsilon, numbermin,expmin,ell;
      epsilonmin := 12345.0 ; Digits := ndigits;
      powerofBoverC := B^(emin-p)/C;
      for e from emin-p+1 to emax-p+1 do
          powerofBoverC := B*powerofBoverC;
          a := floor(powerofBoverC); Plast := a;
          r := 1/(powerofBoverC-a);  a := floor(r);
          Qlast := 1;  Q := a;
          P := Plast*a+1;
          while Q < B^p-1 do
              r := 1/(r-a);
              a := floor(r);
              NewQ := Q*a+Qlast;
              NewP := P*a+Plast;
              Qlast := Q;
              Plast := P;
              Q := NewQ;
              P := NewP
          od;
          epsilon :=
              evalf(C*abs(Plast-Qlast*powerofBoverC));
            if epsilon < epsilonmin then
              epsilonmin := epsilon; numbermin := Qlast;
              expmin := e
          fi
      od;
      print('significand',numbermin);
      print('exponent',expmin);
      print('epsilon',epsilonmin);
      ell := evalf(log(epsilonmin)/log(B),10);
      print('numberofdigits',ell)
  end
```

β	p	C	$e_{\max}$	Worst Case	$-\log_\beta(\epsilon)$
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
2	53	$\pi/2$	1023	$6381956970095103 \times 2^{+797}$	60.9
2	53	$\ln(2)$	1023	$5261692873635770 \times 2^{+499}$	66.8

What can we deduce ?

In all binary formats of the IEEE 754 standard, a FP number x of absolute value $> \pi/2$ is always far enough from an integer multiple of $\pi/2$ to make sure that:

- $\tan(x)$, $1/\tan(x)$ cannot overflow;
- $\sin(x)$, $\cos(x)$, $\tan(x)$ is always of absolute value $> 2^{e_{\min}}$ ($\rightarrow$ never in subnormal domain).

Also gives the precision with which range reduction must be done.

Other application: multiplication by "infinitely precise" constants

- We want $\mathrm{RN}\,(Cx)$, where x is a FP number, and C a real constant (i.e., known at compile-time);
- Base 2, precision-p FP arithmetic;
- Typical values of C: π, $1/\pi$, $\ln(2)$, $\ln(10)$, e, $1/k!$, $\cos(k\pi/N)$ and $\sin(k\pi/N)$, ...
- another frequent case: $C = \frac{1}{\text{FP number}}$ (division by a constant);

The naive method

- replace C by $C_h = \mathrm{RN}(C)$;
- compute $\mathrm{RN}(C_h x)$ (instruction `y = Ch * x`).

p	Prop. of correctly-rounded results
5	0.93750
6	0.78125
7	0.59375
$\cdots$	$\cdots$
16	0.86765
17	0.73558
$\cdots$	$\cdots$
24	0.66805

Proportion of FP numbers x for which $RN(C_h x) = RN(Cx)$ for $C = \pi$ and various p.

Assumptions

- C is not a FP number;
- An fma instruction is available (remember: it computes $\mathrm{RN}(xy+z)$);
- no underflows, no overflows;
- We assume that the two following FP numbers are pre-computed:

$$\left\{ \begin{array}{lll} C_h & = & \mathrm{RN}(C), \\ C_\ell & = & \mathrm{RN}(C - C_h), \end{array} \right.$$

The algorithm

Algorithm 1 (Multiplication by C with a product and an `fma`)
From x, compute

$$\begin{cases} y_1 &= RN(C_\ell x), \\ y_2 &= RN(C_h x + y_1). \end{cases}$$

Returned result: y_2.

- Warning! There exist C and x s.t. $y_2 \neq \mathrm{RN}(Cx)$ – easy to build;
- Without l.o.g., we assume that $1 < x < 2$ and $1 < C < 2$, that C is not exactly representable, and that $C - C_h$ is not a power of 2;

The algorithm

Algorithm 1

From x, compute

$$\begin{cases} y_1 & = & RN(C_\ell x), \\ y_2 & = & RN(C_h x + y_1). \end{cases}$$

Returned result: y_2.

Continued Fractions theory gives two methods for checking if $\forall x, y_2 = \mathrm{RN}(Cx)$.

- the 1st one is simple but does not always give a complete answer;
- the other one gives all "bad cases", or certifies that there are none, i.e. that the algorithm always returns $\mathrm{RN}(Cx)$.

Here we just develop the 1st method.

Analyzing the algorithm

Maximum possible distance between y_2 and Cx:

Property 1

For all FP number x, we have

$$|y_2 - Cx| < \frac{1}{2}\,ulp\,(y_2) + 2\,ulp\,(C_\ell).$$

Proof on next slide.

Analyzing the algorithm

$$C_\ell = \mathrm{RN}(C - C_h) \implies |(C - C_h) - C_\ell| \le \frac{1}{2}\,\mathrm{ulp}(C_\ell)$$

$$|C_\ell x| < 2\cdot|C_\ell| \implies \mathrm{ulp}(C_\ell \cdot x) \le 2\,\mathrm{ulp}(C_\ell)$$

$$\implies |\underbrace{\mathrm{RN}(C_\ell \cdot x)}_{y_1} - C_\ell \cdot x| \le \mathrm{ulp}(C_\ell)$$

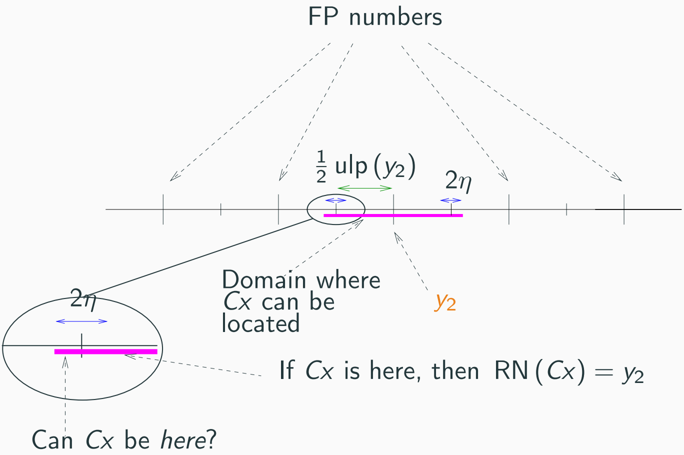
$$|y_2 - (C_h x + y_1)| \le \frac{1}{2}\,\mathrm{ulp}(y_2)$$

$$|(C_h x + y_1) - C_x| = |(C_h x + y_1) - (C_h + C_\ell + (C - C_h - C_\ell))x|$$

$$\le \underbrace{|y_1 - C_\ell x|}_{\le \mathrm{ulp}(C_\ell)} + \underbrace{|C - C_h - C_\ell|}_{\le \frac{1}{2}\mathrm{ulp}(C_\ell)} \cdot \underbrace{x}_{<2}$$

Analyzing the algorithm

Reminder: $|y_2 - Cx| < \frac{1}{2}\,\mathrm{ulp}\,(y_2) + \eta$ with $\eta = 2\,\mathrm{ulp}\,(C_\ell)$.



Analyzing the algorithm

- We know that Cx is within $1/2\,\mathrm{ulp}\,(y_2) + 2\,\mathrm{ulp}\,(C_\ell)$ from the FP number y_2.
- If we prove that Cx cannot be at a distance $\leq \eta = 2\,\mathrm{ulp}\,(C_\ell)$ from the middle of two consecutive FP numbers, then y_2 will be the FP number that is closest to Cx.

Analyzing the algorithm

- Remark: Cx can be in $[1, 2)$ or $[2, 4) \rightarrow$ two (very similar) cases;
- define $x_{\text{cut}} = 2/C$. Let $X = 2^{p-1}x$ and $X_{\text{cut}} = \lfloor 2^{p-1}x_{\text{cut}} \rfloor$.
- we detail the case $x < x_{\text{cut}}$ below.

Middle of two consecutive FP numbers around Cx: $\frac{2A+1}{2^p}$ where $A \in \mathbb{Z}$, $2^{p-1} \le A \le 2^p - 1 \rightarrow$ we try to know if there can be such an A such that

$$\left| Cx - \frac{2A+1}{2^p} \right| < \eta.$$

This is equivalent to

$$|2CX - (2A+1)| < 2^p \eta.$$

Analyzing the algorithm

We want to know if there exists X between 2^{p-1} and X_{cut} and A between 2^{p-1} and $2^p - 1$ such that

$$|2CX - (2A + 1)| < 2^p \eta.$$

- $(p_i/q_i)_{i \geq 1}$: convergents of $2C$;
- k: smallest integer such that $q_{k+1} > X_{\text{cut}}$,
- define $\delta = |p_k - 2Cq_k|$.

Theorem 6 $\Rightarrow \forall B, X \in \mathbb{Z}$, with $0 < X \leq X_{\text{cut}} < q_{k+1}$, $|2CX - B| \geq \delta$.

Analyzing the algorithm

Therefore

1. If $\delta \geq 2^p \eta$ then $|Cx - A/2^p| < \eta$ is impossible $\Rightarrow$ the algorithm returns $\mathrm{RN}(Cx)$ for all $x < x_{\mathrm{cut}}$;
2. if $\delta < 2^p \eta$, we try the algorithm with $x = q_k 2^{-p+1}$ $\rightarrow$ either we get a counter-example, or we cannot conclude

Case $x > x_{\mathrm{cut}}$: similar (convergents of C instead of those of $2C$)

Example: $C = \pi$, double precision ($p = 53$)

```
> method1(Pi/2,53);
Ch = 884279719003555/562949953421312
Cl = 4967757600021511/81129638414606681695789005144064
xcut = 1.2732395447351626862, Xcut = 5734161139222658
eta = .8069505497e-32
pk/qk = 6134899525417045/1952799169684491
delta = .9495905771e-16
OK for X < 5734161139222658
etaprime = .1532072145e-31
pkprime/qkprime = 12055686754159438/7674888557167847
deltaprime = .6943873667e-16
OK for 5734161139222658 <= X < 9007199254740992
```

$\Rightarrow$ We always get a correctly rounded result for $C = 2^k\pi$ and $p = 53$, with $C_h = 2^{k-48} \times 884279719003555$ and $C_\ell = 2^{k-105} \times 4967757600021511$.

Consequence 1

Correctly rounded multiplication by π: in double precision one multiplication and one fma.

Double-Word Arithmetic

Reminder 1: Fast2Sum

Theorem 8 (Fast2Sum (Dekker))

(only radix 2). Let a and b be FP numbers, s.t. $|a| \geq |b|$.
Following algorithm: s and r such that

- $s + r = a + b$ *exactly;*
- s *is "the" FP number that is closest to* $a + b$*;*

Algorithm 2 (FastTwoSum)

$s \leftarrow RN(a+b)$
$z \leftarrow RN(s-a)$
$r \leftarrow RN(b-z)$

Reminder 2: TwoSum (Moller-Knuth)

- no need to compare a and b;
- works in all bases.

Algorithm 3 (TwoSum)

$s \leftarrow RN(a+b)$
$a' \leftarrow RN(s-b)$
$b' \leftarrow RN(s-a')$
$\delta_a \leftarrow RN(a-a')$
$\delta_b \leftarrow RN(b-b')$
$r \leftarrow RN(\delta_a+\delta_b)$

Knuth: if no underflow nor overflow occurs then $a+b=s+r$, and s is nearest $a+b$.

Reminder 3: TwoMultFMA

- Fused multiply-add (fma) instruction: computes $\mathrm{RN}(ab + c)$.
- If a and b are FP numbers and $e_a + e_b \geq e_{\min} + p - 1$, then

$$\begin{cases} p = \mathrm{RN}(ab) \\ r = \mathrm{RN}(ab - p) \end{cases}$$

gives $p + r = ab$.

Double-Word arithmetic

- Fast2Sum, 2Sum and 2MultFMA return their result as the unevaluated sum of two FP numbers.
- idea: manipulate such unevaluated sums to perform more accurate calculations in critical parts of a numerical program.

$\rightarrow$ "double word" or "double-double" arithmetic. Most recent avatar: Rump and Lange's "pair arithmetic" (2020).

Definition 9

A double-word (DW) number x is the unevaluated sum $x_h + x_\ell$ of two floating-point numbers x_h and x_ℓ such that

$$x_h = \mathrm{RN}\,(x).$$

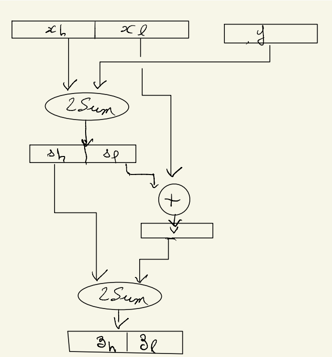
In the following: base 2, precision p floating-point arithmetic.

DW+FP

- Implemented in Bailey's QD library (1999);
- DW number $x = x_h + x_\ell$ plus FP number $y \to$ DW number z;
- measure of error $u = 2^{-p}$.

DWPlusFP

$(s_h, s_\ell) \leftarrow \text{2Sum}(x_h, y)$
$v \leftarrow \text{RN}\,(x_\ell + s_\ell)$
$(z_h, z_\ell) \leftarrow \text{Fast2Sum}(s_h, v)$
return (z_h, z_ℓ)



Exercise: what is the relative error in the case $x_h = 1$, $x_\ell = (2^p - 1) \cdot 2^{-2p}$, $y = -\frac{1}{2} \cdot (1 - 2^{-p})$?

DW+FP

Theorem 10

The relative error

$$\left|\frac{(z_h + z_\ell) - (x + y)}{x + y}\right|$$

of Algorithm DWPlusFP is bounded by $2 \cdot u^2$.

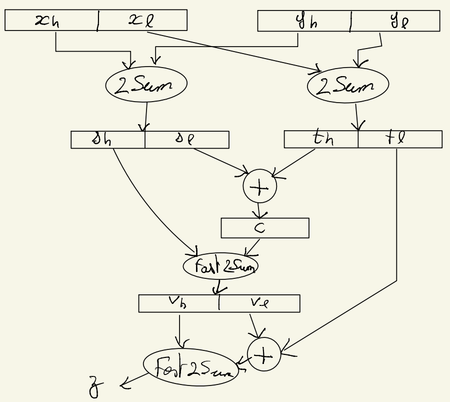
The bound cannot be improved (it is asymptotically optimal). See previous exercise.

DW+DW: "accurate version"

Sum of two DW numbers. There exist a "quick & dirty" algorithm, but its relative error is unbounded.

DWPlusDW

1: $(s_h, s_\ell) \leftarrow \mathrm{2Sum}(x_h, y_h)$
2: $(t_h, t_\ell) \leftarrow \mathrm{2Sum}(x_\ell, y_\ell)$
3: $c \leftarrow \mathrm{RN}\,(s_\ell + t_h)$
4: $(v_h, v_\ell) \leftarrow \mathrm{Fast2Sum}(s_h, c)$
5: $w \leftarrow \mathrm{RN}\,(t_\ell + v_\ell)$
6: $(z_h, z_\ell) \leftarrow \mathrm{Fast2Sum}(v_h, w)$
7: **return** (z_h, z_ℓ)



We have (after a very long and tedious proof):

Theorem 11

If $p \geq 3$, the relative error of Algorithm DWPlusDW is bounded by

$$\frac{3u^2}{1-4u} = 3u^2 + 12u^3 + 48u^4 + \cdots, \tag{2}$$

DW+DW: "accurate version"

So the theorem gives an error bound $3u^2/(1-4u) \simeq 3u^2 \ldots$
That theorem has an interesting history:

- the authors of the paper where the algorithm was published claimed (without proof) an error bound $2u^2$ (in binary64 arithmetic);
- when trying (without success) to prove that bound, we found an example with error $\approx 2.25u^2$;
- we finally proved the theorem, and started to write a formal proof in Coq;
- of course, that led to finding a (minor) flaw in our proof...

DW+DW: "accurate version"

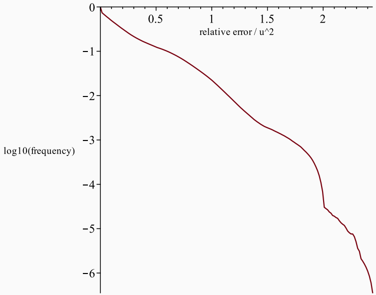
- fortunately the flaw was quickly corrected!
- still, the gap between worst case found ($\approx 2.25u^2$) and the bound ($\approx 3u^2$) was frustrating, so we spent months trying to improve the theorem...
- and of course this could not be done: it was the worst case that needed spending time!
- we finally found that with

$$\begin{array}{rcl} x_h & = & 1 \\ x_\ell & = & u - u^2 \\ y_h & = & -\frac{1}{2} + \frac{u}{2} \\ y_\ell & = & -\frac{u^2}{2} + u^3. \end{array}$$

error $\frac{3u^2-2u^3}{1+3u-3u^2+2u^3}$ is attained. With $p = 53$ (binary64 arithmetic), gives error $2.99999999999999877875 \cdots \times u^2$.

DW+DW: "accurate version"

- We suspect the initial authors hinted their claimed bound by performing zillions of random tests
- in this domain, the worst cases are extremely unlikely: you must build them. Almost impossible to find them by chance.



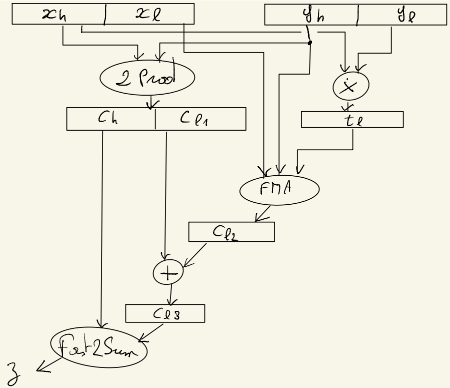
$\log_{10}$ of the frequency of cases for which the relative error of DWPlusDW is $\geq \lambda u^2$ as a function of λ.

DW $\times$ DW

- Product $z = (z_h, z_\ell)$ of two DW numbers $x = (x_h, x_\ell)$ and $y = (y_h, y_\ell)$;
- several algorithms $\rightarrow$ tradeoff speed/accuracy. We just give one of them.

DWTimesDW

$(c_h, c_{\ell 1}) \leftarrow \text{2Prod}(x_h, y_h)$
$t_\ell \leftarrow \text{RN}\,(x_h \cdot y_\ell)$
$c_{\ell 2} \leftarrow \text{RN}\,(t_\ell + x_\ell y_h)$
$c_{\ell 3} \leftarrow \text{RN}\,(c_{\ell 1} + c_{\ell 2})$
$(z_h, z_\ell) \leftarrow \text{Fast2Sum}(c_h, c_{\ell 3})$
return (z_h, z_ℓ)



We have

Theorem 12 (Error bound for Algorithm DWTimesDW)

If $p \geq 5$, the relative error of Algorithm DWTimesDW2 is less than or equal to

$$\frac{5u^2}{(1+u)^2} < 5u^2.$$

and that theorem too has an interesting history!

- initial bound $6u^2$;
- again, we tried formal proof... and it turned out the proof was based on a wrong lemma.

DW $\times$ DW

- after a few nights of very bad sleep, we found a turn-around. . . that also improved the bound !
- no proof of asymptotic optimality, but in binary64 arithmetic, we have examples with error $> 4.98u^2$;
- without the flaw, we would never have found the better bound.

Conclusion: that class of algorithms really needs formal proof. Proofs have too many subcases to be certain you have not forgotten one.