Exact Real Computer Arithmetic Using Continued Fractions

Richard B. Elrod
Youngstown State University
August 7, 2015

Motivation

$$\begin{pmatrix} 64919121 & -159018721 \\ 41869520.5 & -102558961 \end{pmatrix} x = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Haskell

```
solveAxB (a0, a1, a2, a3) (b0, b1) =
  let det = a0 * a3 - a1 * a2
  in ((a3 * b0 - a1 * b1) / det,
      (-a2 * b0 + a0 * b1) / det)
```

Motivation

What Haskell (Ruby, Python, Java, ...) says...

$$x = \begin{pmatrix} 102558961 \\ 41869520.5 \end{pmatrix}$$

The real answer...

$$x = \begin{pmatrix} 205117922 \\ 83739041 \end{pmatrix}$$

Loss of Precision

```
>>> 0.999999999999999
0.999999999999999

>>> 0.9999999999999999
0.9999999999999999

>>> 0.99999999999999999
1.0

>>> 0.999999999999999999999999999999999999999999
1.0
```

Arbitrary Integer Arithmetic

- Many programming languages have arbitrary-precision **integer** arithmetic built-in.
- This can be used to implement arbitrary-precision floating point.
- However, it doesn't help us with *sin(x)*, *sqrt(x)*, etc.
- We want to compute with *exact* $\mathbb{R}$ arithmetic.
    - And we want to be able to pick our own output precision and have the details handled for us.
- There are several solutions.

What is a *Continued Fraction*?

- Continued fractions represent a number (iteratively) as the sum of its integer part and the reciprocal of the remainder.
- For example: $\frac{5}{27} = 0 + \frac{1}{5+\frac{1}{2+\frac{1}{2+0}}}$
- We represent this by $[0; 5, 2, 2]$

A More Detailed Conversion Example

Converting 2.54 ($\frac{254}{100}$)

$$\frac{254}{100} = \mathbf{2}\ (\textit{remainder } 54)$$
$$\frac{100}{54} = \mathbf{1}\ (\textit{remainder } 46)$$
$$\frac{54}{46} = \mathbf{1}\ (\textit{remainder } 8)$$
$$\frac{46}{8} = \mathbf{5}\ (\textit{remainder } 6)$$
$$\frac{8}{6} = \mathbf{1}\ (\textit{remainder } 2)$$
$$\frac{6}{2} = \mathbf{3}\ (\textit{remainder } 0)$$

$$2.54 = \mathbf{2} + \cfrac{1}{\mathbf{1} + \cfrac{1}{\mathbf{1} + \cfrac{1}{\mathbf{5} + \cfrac{1}{\mathbf{1} + \cfrac{1}{\mathbf{3} + 0}}}}}$$
$$= [2; 1, 1, 5, 1, 3]$$

Some Properties Of Continued Fractions

- The continued fraction expansion of a number is finite if and only if the number is rational.
- Simple rational numbers have expansions with few (and small) terms.
- Truncating the expansion of a number gives a really good approximation of that number.

Some Properties Of Continued Fractions
as applied to programming

- Operations on them happen in “most-sigificant-digit first” order.
- Because of this, we can take full advantage of lazy evaluation in certain programming languages.
    - We can pick an accuracy level, then later decide we need more accuracy, and not have to recompute everything.

A More General Example

Let's expand:

$$\frac{70t + 29}{12t + 5}$$

...knowing only that t is positive.

We know the first term is 5, because as t varies between 0 and ∞, $\frac{70}{12} < \frac{70t + 29}{12t + 5} < \frac{29}{5}$.

A More General Example

$\frac{70t+29}{12t+5}$

$$\frac{70t+29}{12t+5} = \mathbf{5}\ (\textit{remainder}\ 10t+4)$$
$$\frac{12t+5}{10t+4} = \mathbf{1}\ (\textit{remainder}\ 2t+1)$$
$$\frac{10t+4}{2t+1} = \mathbf{4}\ (\textit{remainder}\ 2t)$$

...and we are stuck. All we know is that our next result is in $[1, \infty)$.

$$\frac{70t+29}{12t+5} = \mathbf{5} + \cfrac{1}{\mathbf{1} + \cfrac{1}{\mathbf{4} + \cfrac{1}{\frac{2t+1}{2t}}}}$$
$$= [5; 1, 4, \frac{2t+1}{2t}]$$

If we knew that $t > \frac{1}{2}$, then we could construct another term: $\frac{2t+1}{2t} = 1 + \frac{1}{2t}$.

Going In Reverse

Input

Now suppose we are receiving terms of a continued fraction ($[5; 1, 4, 1, \ldots]$) which may go on forever or end with a symbolic expression.

Let's reconstruct the value as we receive terms.

Going In Reverse

Input

Let x be the "rest" of the continued fraction. Before we receive the first term, then, x represents the whole thing.

- When we get the first term (5), $x = 5 + \frac{1}{x'} = \frac{5x' + 1}{x'}$.
- When we get the next term (1), x becomes $1 + \frac{1}{x'}$ and $\frac{5x + 1}{x}$ becomes $\frac{6x' + 5}{x' + 1}$.
- When we get the next term (4), x becomes $4 + \frac{1}{x'}$ and $\frac{6x + 5}{x + 1}$ becomes $\frac{29x' + 6}{5x' + 1}$.
- When we get the next term (1), x becomes $1 + \frac{1}{x'}$ and $\frac{29x + 6}{5x + 1}$ becomes $\frac{35x' + 29}{6x' + 5}$.
- When x becomes $2t$, we have constructed the original value.

Going In Reverse

Input

Let x be the "rest" of the continued fraction. Before we receive the first term, then, x represents the whole thing.

- When we get the first term (5), $x = 5 + \frac{1}{x'} = \frac{5x' + 1}{x'}$.
- When we get the next term (1), x becomes $1 + \frac{1}{x'}$ and $\frac{5x+1}{x}$ becomes $\frac{6x'+5}{x'+1}$.
- When we get the next term (4), x becomes $4 + \frac{1}{x'}$ and $\frac{6x+5}{x+1}$ becomes $\frac{29x'+6}{5x'+1}$.
- When we get the next term (1), x becomes $1 + \frac{1}{x'}$ and $\frac{29x+6}{5x+1}$ becomes $\frac{35x'+29}{6x'+5}$.
- When x becomes $2t$, we have constructed the original value.

Going In Reverse

Input

Let x be the "rest" of the continued fraction. Before we receive the first term, then, x represents the whole thing.

- When we get the first term (5), $x = 5 + \frac{1}{x'} = \frac{5x' + 1}{x'}$.
- When we get the next term (1), x becomes $1 + \frac{1}{x'}$ and $\frac{5x+1}{x}$ becomes $\frac{6x'+5}{x'+1}$.
- When we get the next term (4), x becomes $4 + \frac{1}{x'}$ and $\frac{6x+5}{x+1}$ becomes $\frac{29x'+6}{5x'+1}$.
- When we get the next term (1), x becomes $1 + \frac{1}{x'}$ and $\frac{29x+6}{5x+1}$ becomes $\frac{35x'+29}{6x'+5}$.
- When x becomes $2t$, we have constructed the original value.

Going In Reverse

Input

Let x be the "rest" of the continued fraction. Before we receive the first term, then, x represents the whole thing.

- When we get the first term (5), $x = 5 + \frac{1}{x'} = \frac{5x' + 1}{x'}$.
- When we get the next term (1), x becomes $1 + \frac{1}{x'}$ and $\frac{5x + 1}{x}$ becomes $\frac{6x' + 5}{x' + 1}$.
- When we get the next term (4), x becomes $4 + \frac{1}{x'}$ and $\frac{6x + 5}{x + 1}$ becomes $\frac{29x' + 6}{5x' + 1}$.
- When we get the next term (1), x becomes $1 + \frac{1}{x'}$ and $\frac{29x + 6}{5x + 1}$ becomes $\frac{35x' + 29}{6x' + 5}$.
- When x becomes $2t$, we have constructed the original value.

Going In Reverse

Input

Let x be the "rest" of the continued fraction. Before we receive the first term, then, x represents the whole thing.

- When we get the first term (5), $x = 5 + \frac{1}{x'} = \frac{5x' + 1}{x'}$.
- When we get the next term (1), x becomes $1 + \frac{1}{x'}$ and $\frac{5x + 1}{x}$ becomes $\frac{6x' + 5}{x' + 1}$.
- When we get the next term (4), x becomes $4 + \frac{1}{x'}$ and $\frac{6x + 5}{x + 1}$ becomes $\frac{29x' + 6}{5x' + 1}$.
- When we get the next term (1), x becomes $1 + \frac{1}{x'}$ and $\frac{29x + 6}{5x + 1}$ becomes $\frac{35x' + 29}{6x' + 5}$.
- When x becomes $2t$, we have constructed the original value.

Going In Reverse

The pattern

When x becomes $a + \frac{1}{x'}$, then $\frac{qx + r}{sx + t}$ becomes $\frac{(aq + r)x' + q}{(as + t)x' + s}$.

Arithmetic

Consider functions of the form $f(x) = \frac{ax+b}{cx+d}$ where $a, b, c, d \in \mathbb{Z}$.
These are called **homographic functions**.
We can express these in code as a 4-tuple.

Haskell

```
type CF = [Integer]

type Hom = (Integer, Integer,
            Integer, Integer)

hom :: Hom -> CF -> CF    -- Goal.
```

Arithmetic

If $x = [x_0; \ldots] = x_0 + \frac{1}{x'} \ldots$
then by algebra,

$$\frac{\mathbf{a}x + \mathbf{b}}{\mathbf{c}x + \mathbf{d}} = \frac{a(x_0 + \frac{1}{x'}) + b}{c(x_0 + \frac{1}{x'}) + d}$$

$$= \frac{(\mathbf{a}\mathbf{x}_0 + \mathbf{b})x' + \mathbf{a}}{(\mathbf{c}\mathbf{x}_0 + \mathbf{d})x' + \mathbf{c}}$$

Arithmetic

Absorbing terms

Absorbing terms

```
absorb :: Hom -> Integer -> Hom
absorb (a, b,
        c, d) x0 = (a * x0 + b,      a,
                    c * x0 + d,      c)
```

An invariant

```
hom h (x0:rest) == hom (absorb h x0) rest
```

Arithmetic

Emitting terms

If $z = \frac{ax+b}{cx+d}$, and $x \in \mathbb{Z}^+ \cup \{0\}$, then $z \in [\frac{a}{c}, \frac{b}{d}]$.
If $\frac{a}{c}$ and $\frac{b}{d}$ have the same integer part, q, then we know $z = [q; \ldots]$.

Emitting terms

```
tryEmit :: Hom -> Maybe Integer
tryEmit (a, b,
         c, d) = if c /= 0 &&
                    d /= 0 &&
                    (a `quot` c == b `quot` d)
                 then Just (a `quot` c)
                 else Nothing
```

Arithmetic

Emitting terms

Let our result $z = q + \frac{1}{z'}$, then:

$$\begin{aligned} z' &= (z - q)^{-1} \\ &= \left(\frac{ax+b}{cx+d} - q\right)^{-1} \\ &= \frac{\mathbf{c}x + \mathbf{d}}{\mathbf{(a - cq)}x + \mathbf{(b - dq)}} \end{aligned}$$

Arithmetic

Emitting terms

Emitting terms

```
emit :: Hom -> Integer -> Hom
emit (a, b,
      c, d) q = (c,                 d,
                 a - (c * q),       b - (d * q))
```

An invariant

```
hom h cf == q : hom (emit q h) cf
```

Arithmetic

Deciding when to absorb or emit

Since we're using Haskell, we should take as much advantage of lazy evaluation as possible!

Finally, a `hom` implementation!

```
hom :: Hom -> CF -> CF
hom h cf@(x:rest) = case tryEmit h of
                      Just z   -> z : hom (emit h z) cf
                      Nothing -> hom (absorb h x) rest
```

Quick example

```
2 * pi == hom (2, 0,
               0, 1) pi
       == CF [6, 3, 1, 1, 7, 2, 146, 3, 6, 1, ...]
```

Back To Our Motivation

Before (using Double)

```
> solveAxB (64919121 :: Double, -159018721,
            41869520.5,          -102558961) (1, 0)
(1.02558961e8,4.18695205e7)
```

After (using continued fractions)

```
> solveAxB (64919121 :: CF, -159018721,
            41869520.5,     -102558961) (1, 0)
(205117922,83739041)
```

What I Haven't Shown

- Doing arithmetic on two continued fractions.
  - Same as we've done except for $\dfrac{axy + bx + cy + d}{exy + fx + gy + h}$.
  - These are called **bihomographic functions**.
  - By manipulating a through h, we can generate all of the arithmetic operations, and absorb/emit from x and y as before.
- Transcendental functions
- Integrating with Haskell's typeclass hierarchy

Future Research

- Implementation in Coq or another automated theorem assistant.
  - This would let us easily prove that our invariants hold in every possible case.
- Looking at continued fractions topologically.
- Employing type theory via domain-theoretic notions of definedness. (Peter Potts)

Thank You...

- Dr. Stephen Rodabaugh
- Jeffrey Deniston
- Dr. Jacek Fabrykowski
- Youngstown State University Department of Mathematics and Statistics
- You!

- Ralph W Gosper. "Continued fraction arithmetic". In: HAKMEM Item 101B, MIT Artificial Intelligence Memo 239 (1972).
- Mitchell Riley. Exact Real Arithmetic in Haskell. BFPG Presentation. 2015.
- Jean E Vuillemin. "Exact real computer arithmetic with continued fractions". In: Computers, IEEE Transactions on 39.8 (1990), pp. 1087–1105.