Towards Optimizing MPMF Numerics
Amazing Collaborators

Pavel Panchekha
Max Willsey
Bill Zorn
Chandakana Nandi
Remy Wang
Oliver Flatt
Adam Anderson
Eva Darulova
James Wilcox
Dan Grossman
Brett Saiki
Alex Sanchez-Stern
David Thien
Chen Qiu
Sorin Lerner
Today’s Menu
Today’s Menu

1. Herbie: Automated FP Accuracy Improvement
Today’s Menu

1. Herbie: Automated FP Accuracy Improvement

2. egg: Fast, Flexible Equality Saturation
Today’s Menu

1. Herbie: Automated FP Accuracy Improvement

2. egg: Fast, Flexible Equality Saturation

3. MPMF: Multi-precision, Multi-format Numerics
Floating Point’s Wild Success
Floating Point’s Wild Success

Accuracy for basic ops

$$\lceil x \times y \rceil_F = \text{Round}(\lceil x \times y \rceil_R)$$
Floating Point’s Wild Success

Accuracy for basic ops

$$[x \times y]_F = \text{Round}([x \times y]_R)$$

Compute in $\mathbb{F}$
Floating Point’s Wild Success

Accuracy
for basic ops

\[ [x \times y]_F = \text{Round} ([x \times y]_\mathbb{R}) \]
Floating Point’s Wild Success

Accuracy for basic ops

\[ [x \times y]_F = \text{Round}([x \times y]_R) \]

- Compute in \( \mathbb{F} \)
- Round to \( \mathbb{F} \)
- Compute in \( \mathbb{R} \)
Floating Point’s Wild Success

Accuracy

for basic ops

$$[x \times y]_F = \text{Round}([x \times y]_\mathbb{R})$$

Flexibility

vast range: $10^{-324}$ to $10^{308}$
Floating Point’s Wild Success

Accuracy

for basic ops

\[
[x \times y]_F = \text{Round}([x \times y]_R)
\]

Flexibility

vast range: \(10^{-324}\) to \(10^{308}\)

Performance

cheap GFLOPS
Floating Point’s Wild Success
Floating Point’s Wild Success
Floating Point’s Wild Success

\[ F \approx R \]

Often floating point is close to real arithmetic
Floating Point’s Wild Success

Often floating point is close to real arithmetic

But not always!
Rounding Error in Quadratic

\[-b + \sqrt{b^2 - 4ac} \over 2a\]
Rounding Error in Quadratic

\[-b + \sqrt{b^2 - 4ac} \over 2a\]
Rounding Error in Quadratic

\[- \frac{b + \sqrt{b^2 - 4ac}}{2a}\]

What is rounding error?
Rounding Error in Quadratic

\[-b + \sqrt{b^2 - 4ac} \over 2a\]

What is rounding error?

exact \$[e]_R\$

computed \$[e]_F\$

7 ULPs
Rounding Error in Quadratic

\[-b + \sqrt{b^2 - 4ac} \over 2a\]

What is rounding error?

exact \( [e]_R \) computed \( [e]_F \)

\[ 7 \text{ ULPs} \]

\( \log(\text{ULPs}) \) estimates \# of incorrect bits
Rounding Error in Quadratic

\[-b + \sqrt{b^2 - 4ac}\]

ULPs provide nice error measure:
- accounts for distribution of \( \mathbb{F} \)
- fast to compute

\[\text{ulps}(f1, f2) \approx |((\text{uint64}) f1) - ((\text{uint64}) f2)|\]

\[\log(\text{ULPs})\] estimates \# of incorrect bits
Rounding Error in Quadratic

\[-b + \sqrt{b^2 - 4ac}\]

ULPs provide nice error measure:
- accounts for distribution of \( \mathbb{F} \)
- fast to compute

\[\text{ulps}(f1, f2) \approx |((\text{uint64}) f1) - ((\text{uint64}) f2)|\]

Most number systems support ULP-like error measure
Rounding Error in Quadratic

\[-b + \sqrt{b^2 - 4ac}\]

\[\frac{2a}{2a}\]
Rounding Error in Quadratic

\[-b + \sqrt{b^2 - 4ac} \over 2a\]
Rounding Error in Quadratic

\[-b + \frac{\sqrt{b^2 - 4ac}}{2a} \rightarrow \]

\[\log(ULPs)\]

\[b\]
Rounding Error in Quadratic

\[-b + \frac{\sqrt{b^2 - 4ac}}{2a}\]
Rounding Error in Quadratic

\[-b + \sqrt{b^2 - 4ac} \quad \frac{2a}{2a}\]

\[
\begin{cases}
\frac{c}{b} - \frac{b}{a} & \text{if } b \in A
\end{cases}
\]

Overflow

If \( b \) is large, \( \lceil b^2 \rceil_F \) overflows and the whole expression returns \( \infty \).
Rounding Error in Quadratic

\[
-b + \frac{\sqrt{b^2 - 4ac}}{2a}
\]

\[
\begin{cases}
\frac{c}{b} - \frac{b}{a} & \text{if } b \in A
\end{cases}
\]
Rounding Error in Quadratic

\[-b + \sqrt{b^2 - 4ac} \over 2a\]

\[
\begin{align*}
\left\{ \begin{array}{ll}
\frac{c}{b} - \frac{b}{a} & \text{if } b \in \textbf{A} \\
\frac{-b + \sqrt{b^2 - 4ac}}{2a} & \text{if } b \in \textbf{B}
\end{array} \right.
\end{align*}
\]

Pretty Accurate
Rounding Error in Quadratic

\[-b + \sqrt{b^2 - 4ac} \over 2a\]

\[\begin{cases}
\frac{c}{b} - \frac{b}{a} & \text{if } b \in \mathbf{A} \\
\frac{-b + \sqrt{b^2 - 4ac}}{2a} & \text{if } b \in \mathbf{B}
\end{cases}\]
Rounding Error in Quadratic

\[ \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \Rightarrow \quad \begin{cases} \frac{c}{b} - \frac{b}{a} & \text{if } b \in \text{A} \\ \frac{-b + \sqrt{b^2 - 4ac}}{2a} & \text{if } b \in \text{B} \\ \frac{2c}{-b - \sqrt{b^2 - 4ac}} & \text{if } b \in \text{C} \end{cases} \]

Catastrophic Cancellation

If \( b \) is large, but \( a \) and \( c \) are small, \( b \approx \sqrt{b^2 - 4ac} \) and the difference is rounded off.
Rounding Error in Quadratic

\[ -b + \sqrt{b^2 - 4ac} \]
\[ \frac{2a}{2a} \]

\[ \begin{cases} \frac{c - b}{a} & \text{if } b \in \mathbb{A} \\ \frac{-b + \sqrt{b^2 - 4ac}}{2a} & \text{if } b \in \mathbb{B} \\ \frac{2c}{-b - \sqrt{b^2 - 4ac}} & \text{if } b \in \mathbb{C} \end{cases} \]

\[ \text{log(ULPs)} \]

\[ b \]

\[ A \quad B \quad C \quad D \]
Rounding Error in Quadratic

\[-b + \sqrt{b^2 - 4ac} \over 2a\]  

\[
\begin{cases}
\frac{c}{b} - \frac{b}{a} & \text{if } b \in A \\
\frac{-b + \sqrt{b^2 - 4ac}}{2a} & \text{if } b \in B \\
\frac{2c}{-b - \sqrt{b^2 - 4ac}} & \text{if } b \in C \\
-\frac{c}{b} & \text{if } b \in D
\end{cases}
\]

Overflow again
Rounding Error in Quadratic

\[-b + \sqrt{b^2 - 4ac} \over 2a\]

\[\Rightarrow\]

\[\begin{align*}
&\frac{c}{b} - \frac{b}{a} & \text{if } b \in A \\
&\frac{-b + \sqrt{b^2 - 4ac}}{2a} & \text{if } b \in B \\
&\frac{2c}{-b - \sqrt{b^2 - 4ac}} & \text{if } b \in C \\
&-\frac{c}{b} & \text{if } b \in D
\end{align*}\]
Rounding Error in Quadratic

\[-b + \frac{\sqrt{b^2 - 4ac}}{2a}\]

\[
\begin{align*}
\frac{c}{b} - \frac{b}{a} & \quad \text{if } b \in \text{A} \\
-\frac{b + \sqrt{b^2 - 4ac}}{2a} & \quad \text{if } b \in \text{B} \\
\frac{2c}{-b - \sqrt{b^2 - 4ac}} & \quad \text{if } b \in \text{C} \\
-\frac{c}{b} & \quad \text{if } b \in \text{D}
\end{align*}
\]
Rounding Error in Sculpture
Rounding Error in Sculpture

Blake Courter
@bcourter
Rounding Error in Sculpture

Blake Courter
@bcourter
Rounding Error in Sculpture

Blake Courter
@bcourter
Rounding Error in Sculpture

Blake Courter
@bcourter
Rounding Error in Sculpture

Blake Courter
@bcourter

Rounding error
Traditional Mitigations
Traditional Mitigations

Futz
Traditional Mitigations

Futz

+ Easy
+ Fast
- Unreliable
Traditional Mitigations

- Easy
- Fast
- Unreliable
Traditional Mitigations

Futz

+ Easy
+ Fast

– Unreliable
Traditional Mitigations

Futz

+ Easy
+ Fast
- Unreliable

Big Float

+ Easy
+ More Reliable
- Really Slow
Traditional Mitigations

Futz

- Easy
- Fast
- Unreliable

MPFR

+ Easy
+ More Reliable
- Really Slow

Analyze

Numerical Methods for Scientists and Engineers

R.W. Hamming
Second Edition
Traditional Mitigations

Futz

+ Easy
+ Fast
- Unreliable

Big Float

- Reliable
- Fast
- Difficult*

Analyse

+ Easy
+ More Reliable
- Really Slow

Numerical Methods for Scientists and Engineers

R.W. Hamming
Second Edition
Goal: Automatically improve floating point accuracy
Herbie Architecture

e -> sample -> cands -> regimes -> e'

cands

<table>
<thead>
<tr>
<th>focus</th>
</tr>
</thead>
<tbody>
<tr>
<td>rewrite</td>
</tr>
<tr>
<td>simplify</td>
</tr>
<tr>
<td>series</td>
</tr>
</tbody>
</table>
Herbie Architecture

- Sample
- Cands
- Regimes
- Focus
- Rewrite
- Simplify
- Series
- e
- e'
Sample Inputs, Evaluate Error
Sample Inputs, Evaluate Error

\[ P = \text{sample}(\text{domain}(e)) \]
Sample Inputs, Evaluate Error

\[ P = \text{sample}(\text{domain}(e)) \]

Get approx results: \( [e]_F(P) \)
Sample Inputs, Evaluate Error

\[ P = \text{sample}\left(\text{domain}\left( e \right)\right) \]

Get approx results: \([e]_F(P)\)

Get exact results: \([e]_R(P)\)
Sample Inputs, Evaluate Error

\[ P = \text{sample}(\text{domain}(e)) \]

Get approx results: \([e]_F(P)\)

Get exact results: \([e]_\mathbb{R}(P)\)

Can’t compute over \(\mathbb{R}\) ...
Sample Inputs, Evaluate Error

\[ P = \text{sample}(\text{domain}(e)) \]

Get approx results: \( [e]_F(P) \)

Get exact results: \( [e]_\mathbb{R}(P) \)

Herbie provides interval arithmetic using MPFR.

Increase precision (# of bits) till interval includes just 1 float.
Sample Inputs, Evaluate Error

\[ P = \text{sample}(\text{domain}(e)) \]

Get approx results: \( [e]_F(P) \)

Get exact results: \( [e]_R(P) \)
Sample Inputs, Evaluate Error

\[ P = \text{sample(\text{domain}(e))} \]

Get approx results: \( [e]_F(P) \)

Get exact results: \( [e]_R(P) \)

Determine error on \( P \) : \[ ||[e]_R(P) - [e]_F(P)|| \]
Herbie Architecture

sample → e → cand → regimes → e'
focus → rewrite → simplify → series
Herbie Architecture

e \rightarrow \text{sample} \rightarrow \text{cands} \rightarrow \text{regimes} \rightarrow e' \leftarrow \text{focus} \downarrow \text{rewrite} \downarrow \text{simplify} \downarrow \text{series}
Focus: Estimate Error Source

\[-b + \frac{\sqrt{b^2 - 4ac}}{2a}\]
Focus: Estimate Error Source

\[-b + \frac{\sqrt{b^2 - 4ac}}{2a}\]

Expand AST
Focus: Estimate Error Source

\[
\frac{\sqrt{2a}}{a - b - 4}\sqrt{ac}
\]
Focus: Estimate Error Source

Compute local errors

\[(a - b) \sqrt{2 - \frac{a c^4}{b}}\]
Focus: Estimate Error Source

Compute local errors
start at leaves
Focus: Estimate Error Source

Compute local errors
start at leaves

approx 1.0
exact 1.0
local err 0
Focus: Estimate Error Source

Compute local errors
start at leaves

```
+   *
  
-sqrt 2 a
  
  b  b 4
  
  *  *
  
  b  b  a c
```

<table>
<thead>
<tr>
<th></th>
<th>approx</th>
<th>exact</th>
<th>local err</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0.01</td>
<td>0.01</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>1.0</td>
<td>1.0</td>
<td>0</td>
</tr>
</tbody>
</table>
Focus: Estimate Error Source

Compute local errors
start at leaves

\[
\sqrt{2a - b} \cdot (4b^4) \cdot (ac)
\]
Focus: Estimate Error Source

Compute local errors
start at leaves
work bottom-up

\[ \sqrt{2a^2 - b^4} \]
Focus: Estimate Error Source

Compute local errors
start at leaves
work bottom-up
Focus: Estimate Error Source

Compute local errors
start at leaves
work bottom-up

\[ a - b \sqrt{a^2 b^2} \]

Approx: 1.0
Exact: 1.0
Local error: 0

|exact - approx| of operator
applied to exact child args
Focus: Estimate Error Source

Compute local errors

start at leaves
work bottom-up
Focus: Estimate Error Source

Compute local errors
start at leaves
work bottom-up

approx -1.0
exact -1.0
local err 0

approx 0.999
exact 0.999
local err 0
Focus: Estimate Error Source

Compute local errors
start at leaves
work bottom-up

approx ... exact ... local err 410
approx -1.0 exact -1.0 local err 0
approx 0.999 exact 0.999 local err 0
Focus: Estimate Error Source

Compute local errors
start at leaves
work bottom-up

\[
\frac{a}{\sqrt{b^2 - c^4}}
\]
Focus: Estimate Error Source

Compute local errors

- start at leaves
- work bottom-up

Local error estimates how an individual operation contributes to overall error.
Focus: Estimate Error Source

Compute local errors
start at leaves
work bottom-up

Find max local error
Focus: Estimate Error Source

Compute local errors
start at leaves
work bottom-up

Find max local error

\[
2a - b - \sqrt{4*a - b} c
\]
Focus: Estimate Error Source

Compute local errors
start at leaves
work bottom-up

Find max local error

Local error also plays a key role in other numerical tools!
Herbie Architecture

- Sample: $e$ → $cands$
- Regimes: $cands$ → $e'$
- Focus: $e'$
- Rewrite: $e'$
- Simplify: $e'$
- Series: $e'$
Herbie Architecture

e sample \rightarrow \text{cands} \leftarrow \text{focus} \downarrow \text{rewrite} \downarrow \text{simplify} \downarrow \text{series} \rightarrow \text{e'}

regimes
Rewrite: Generate Candidates

Apply rewrites to

\[-b \pm \sqrt{b^2 - 4ac} \]

\[\frac{2a}{2a}\]
Apply rewrites to

\[-b \pm \sqrt{b^2 - 4ac} \]
\[\frac{2a}{2a}\]

**Rule DB**

\[x + y \Rightarrow y + x\]
\[(x + y) + z \Rightarrow x + (y + z)\]
\[x + y \Rightarrow (x^2 - y^2) / (x - y)\]
...

Rewrite: Generate Candidates
Rewrite: Generate Candidates

Apply rewrites to

\[-b \pm \sqrt{b^2 - 4ac} \over 2a\]

\[
\begin{align*}
-x + y & => y + x \\
(x + y) + z & => x + (y + z) \\
x + y & => (x^2 - y^2) / (x - y) \\
...\end{align*}
\]
Rewrite: Generate Candidates

Apply rewrites to

\[ \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \]

x + y => y + x
(x + y) + z => x + (y + z)
x + y => (x^2 - y^2) / (x - y)
...

Commute
Associate

Rule DB
Rewrite: Generate Candidates

Apply rewrites to

\[ \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \]

Rule DB

- Commute: \( x + y \Rightarrow y + x \)
- Associate: \( (x + y) + z \Rightarrow x + (y + z) \)
- \( x + y \Rightarrow \frac{x^2 - y^2}{x - y} \)

"flip": \( (a + b) \times \left(\frac{(a - b)}{(a - b)}\right) \)
Rewrite: Generate Candidates

Apply rewrites to

\[-b \pm \sqrt{b^2 - 4ac} \over 2a\]

- Commute
- Associate
- "flip": \((a + b) \times ((a - b) / (a - b))\)
  + dozens more: trig, algebra, etc.
Rewrite: Generate Candidates

Apply rewrites to

\[-\frac{b \pm \sqrt{b^2 - 4ac}}{2a}\]

\[
\begin{align*}
x + y &= y + x \\
(x + y) + z &= x + (y + z) \\
x + y &= \frac{(x^2 - y^2)}{(x - y)} \\
&\vdots
\end{align*}
\]
Rewrite: Generate Candidates

Apply rewrites to *\star*

\[ x + y \Rightarrow y + x \]
\[ (x + y) + z \Rightarrow x + (y + z) \]
\[ x + y \Rightarrow \frac{x^2 - y^2}{x - y} \]

...
Rewrite: Generate Candidates

Apply rewrites to

\[ -b \implies \sqrt{b^2 - 4ac} \]
\[ 2a \]

\[ \sqrt{b^2 - 4ac} + (-b) \]
\[ 2a \]

\[ x + y \implies y + x \]
\[ (x + y) + z \implies x + (y + z) \]
\[ x + y \implies (x^2 - y^2) / (x - y) \]

...
Rewrite: Generate Candidates

Apply rewrites to ✭

\[-b \pm \sqrt{b^2 - 4ac} \over 2a\]  \Rightarrow  \[\sqrt{b^2 - 4ac} \pm b \over 2a\]

x + y => y + x
(x + y) + z => x + (y + z)
x + y => (x^2 - y^2) / (x - y)
...
Rewrite: Generate Candidates

Apply rewrites to ✳️

\[
\frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad \xrightarrow{\text{Apply rewrites}} \quad \frac{\sqrt{b^2 - 4ac} + -b}{2a}
\]

- \(x + y \Rightarrow y + x\)
- \((x + y) + z \Rightarrow x + (y + z)\)
- \(x + y \Rightarrow (x^2 - y^2) / (x - y)\)
- ...

 Doesn’t match ✳️
Rewrite: Generate Candidates

Apply rewrites to

\[
\frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \Rightarrow \quad \frac{\sqrt{b^2 - 4ac} + -b}{2a}
\]

\[
\begin{align*}
x + y &= y + x \\
(x + y) + z &= x + (y + z) \\
x + y &= (x^2 - y^2) / (x - y) \\
&\ldots
\end{align*}
\]
Rewrite: Generate Candidates

Apply rewrites to

\[ x + y \Rightarrow y + x \]
\[ (x + y) + z \Rightarrow x + (y + z) \]
\[ x + y \Rightarrow \frac{x^2 - y^2}{x - y} \]

\[ \frac{\sqrt{b^2 - 4ac} + -b}{2a} \]
Rewrite: Generate Candidates

Apply rewrites to

\[ x + y \Rightarrow y + x \]
\[ (x + y) + z \Rightarrow x + (y + z) \]
\[ x + y \Rightarrow \frac{x^2 - y^2}{x - y} \]
...

\[ \frac{\sqrt{b^2 - 4ac} - b}{2a} \]
Rewrite: Generate Candidates

Apply rewrites to

\[
\frac{\sqrt{b^2 - 4ac} + -b}{2a} \quad \frac{(-b)^2 - (\sqrt{b^2 - 4ac})^2}{-b - \sqrt{b^2 - 4ac}} / 2a
\]

\[
x + y \Rightarrow y + x
\]
\[
(x + y) + z \Rightarrow x + (y + z)
\]
\[
x + y \Rightarrow \frac{x^2 - y^2}{x - y}
\]

...
Rewrite: Generate Candidates

Apply rewrites to ✱

\[
-\frac{b + \sqrt{b^2 - 4ac}}{2a} \rightarrow \frac{\sqrt{b^2 - 4ac} + \sqrt{b^2 - 4ac}}{2a}
\]

\[
\frac{\sqrt{b^2 - 4ac} + \sqrt{b^2 - 4ac}}{2a} \rightarrow \frac{(-b)^2 - (\sqrt{b^2 - 4ac})^2}{-b - \sqrt{b^2 - 4ac}} / 2a
\]

x + y => y + x
(x + y) + z => x + (y + z)
x + y => (x^2 - y^2) / (x - y)
...

Rewrite: Generate Candidates

Apply rewrites to

\[
\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}
\]

\[
\frac{\sqrt{b^2 - 4ac} + -b}{2a}
\]

\[
\left(\frac{(-b)^2 - (\sqrt{b^2 - 4ac})^2}{-b - \sqrt{b^2 - 4ac}}\right) / 2a
\]

No cancellation in denominator

x + y => y + x
(x + y) + z => x + (y + z)
x + y => (x^2 - y^2) / (x - y)
...

rewrites: 
- x + y => y + x
- (x + y) + z => x + (y + z)
- x + y => (x^2 - y^2) / (x - y)
- ...

No cancellation in denominator
Rewrite: Generate Candidates

Apply rewrites to

\[
\frac{-b \star \sqrt{b^2 - 4ac}}{2a}
\]

\[
\frac{\sqrt{b^2 - 4ac} + -b}{2a}
\]

\[
\left( \frac{(-b)^2 - (\sqrt{b^2 - 4ac})^2}{-b - \sqrt{b^2 - 4ac}} \right) / 2a
\]

\[
x + y \Rightarrow y + x
\]

\[
(x + y) + z \Rightarrow x + (y + z)
\]

\[
x + y \Rightarrow \frac{(x^2 - y^2)}{(x - y)}
\]

...
Rewrite: Generate Candidates

Apply rewrites to

\[-b \pm \sqrt{b^2 - 4ac} \frac{\sqrt{b^2 - 4ac} + -b}{2a} \]

\[-b \pm \sqrt{b^2 - 4ac} \frac{(-b)^2 - (\sqrt{b^2 - 4ac})^2}{-b - \sqrt{b^2 - 4ac}} / 2a \]

\[x + y \Rightarrow y + x\]
\[(x + y) + z \Rightarrow x + (y + z)\]
\[x + y \Rightarrow (x^2 - y^2) / (x - y)\]

\[\ldots\]
Rewrite: Generate Candidates

Apply rewrites to

\[
\frac{-b + \sqrt{b^2 - 4ac}}{2a}
\]

\[
\frac{\sqrt{b^2 - 4ac} + -b}{2a}
\]

\[
\left(\frac{(-b)^2 - (\sqrt{b^2 - 4ac})^2}{-b - \sqrt{b^2 - 4ac}}\right) / 2a
\]

\[
x + y \Rightarrow y + x
\]

\[
(x + y) + z \Rightarrow x + (y + z)
\]

\[
x + y \Rightarrow (x^2 - y^2) / (x - y)
\]

\[
\ldots
\]

If operator matches, but not operands, use flexible recursive rewrite strategy.
Herbie Architecture

\[ e \xrightarrow{\text{sample}} \text{cands} \xrightarrow{\text{regimes}} e' \]

- focus
- rewrite
- simplify
- series
Herbie Architecture

e \xrightarrow{sample} \text{cands} \xrightarrow{regimes} e'

\text{focus} \quad \text{rewrite} \quad \text{simplify} \quad \text{series}
Simplify Expressions

\[
\left( \frac{(-b)^2 - (\sqrt{b^2 - 4ac})^2}{-b - \sqrt{b^2 - 4ac}} \right) / 2a
\]
Simplify Expressions

\[
\left( \frac{(-b)^2 - (\sqrt{b^2 - 4ac})^2}{-b - \sqrt{b^2 - 4ac}} \right) / 2a
\]

\[
= \left( \frac{b^2 - (\sqrt{b^2 - 4ac})^2}{-b - \sqrt{b^2 - 4ac}} \right) / 2a
\]

\[
= \left( \frac{b^2 - (b^2 - 4ac)}{-b - \sqrt{b^2 - 4ac}} \right) / 2a
\]

\[
= \left( \frac{4ac}{-b - \sqrt{b^2 - 4ac}} \right) / 2a
\]

\[
= \frac{2c}{-b - \sqrt{b^2 - 4ac}}
\]
Simplify Expressions

\[
\frac{\left( (-b)^2 - \left( \sqrt{b^2 - 4ac} \right)^2 \right)}{-b - \sqrt{b^2 - 4ac}} \bigg/ 2a
\]

\[
= \frac{\left( \frac{b^2 - \left( \sqrt{b^2 - 4ac} \right)^2}{-b - \sqrt{b^2 - 4ac}} \right)}{2a}
\]

\[
= \frac{\left( \frac{b^2 - (b^2 - 4ac)}{-b - \sqrt{b^2 - 4ac}} \right)}{2a}
\]

\[
= \frac{\left( \frac{4ac}{-b - \sqrt{b^2 - 4ac}} \right)}{2a}
\]

\[
= \frac{2c}{-b - \sqrt{b^2 - 4ac}}
\]

Tricky! [Caviness '70]
- many possible rewrites
- “simpler” not always clear
- huge search space
- avoid undoing progress!
Simplify Expressions

\[
\left( \frac{(-b)^2 - \left( \sqrt{b^2 - 4ac} \right)^2}{-b - \sqrt{b^2 - 4ac}} \right) / 2a
\]

\[
= \left( \frac{b^2 - \left( \sqrt{b^2 - 4ac} \right)^2}{-b - \sqrt{b^2 - 4ac}} \right) / 2a
\]

\[
= \left( \frac{b^2 - (b^2 - 4ac)}{-b - \sqrt{b^2 - 4ac}} \right) / 2a
\]

\[
= \left( \frac{4ac}{-b - \sqrt{b^2 - 4ac}} \right) / 2a
\]

\[
= \frac{2c}{-b - \sqrt{b^2 - 4ac}}
\]

Tricky! [Caviness ’70]
- many possible rewrites
- “simpler” not always clear
- huge search space
- avoid undoing progress!

E-graphs [Nelson ’79]
- track equiv classes
- restrict rewrites
- select smallest AST
Simplify Expressions

\[
\left( \frac{(-b)^2 - (\sqrt{b^2 - 4ac})^2}{-b - \sqrt{b^2 - 4ac}} \right) / 2a
\]

\[
= \left( \frac{b^2 - (\sqrt{b^2 - 4ac})^2}{-b - \sqrt{b^2 - 4ac}} \right) / 2a
\]

\[
= \left( \frac{b^2 - (b^2 - 4ac)}{-b - \sqrt{b^2 - 4ac}} \right) / 2a
\]

\[
= \left( \frac{4ac}{-b - \sqrt{b^2 - 4ac}} \right) / 2a
\]

\[
= \frac{2c}{-b - \sqrt{b^2 - 4ac}}
\]

Tricky! [Caviness ’70]
- many possible rewrites
- “simpler” not always clear
- huge search space
- avoid undoing progress!

E-graphs [Nelson ’79]
- track equiv classes
- restrict rewrites
- select smallest AST

Huge bottleneck!
Simplify Expressions

\[
\frac{(-b)^2 - \left(\sqrt{b^2 - 4ac}\right)^2}{-b - \sqrt{b^2 - 4ac}} / 2a
\]

= \frac{\left(\frac{b^2 - \left(\sqrt{b^2 - 4ac}\right)^2}{-b - \sqrt{b^2 - 4ac}}\right)}{2a}

= \frac{\left(\frac{b^2 - (b^2 - 4ac)}{-b - \sqrt{b^2 - 4ac}}\right)}{2a}

= \frac{\left(\frac{4ac}{-b - \sqrt{b^2 - 4ac}}\right)}{2a}

= \frac{2c}{-b - \sqrt{b^2 - 4ac}}

Tricky! [Caviness ’70]
- many possible rewrites
- “simpler” not always clear
- huge search space
- avoid undoing progress!

E-graphs [Nelson ’79]
- track equiv classes
- restrict rewrites
- select smallest AST

Huge bottleneck!
Herbie Architecture

- $e$ (sample)
- $cands$
- $e'$ (regimes)
- Focus
- Rewrite
- Simplify
- Series
Series Expansions

\[-b + \frac{\sqrt{b^2 - 4ac}}{2a}\]
Series Expansions

\[ -b + \frac{\sqrt{b^2 - 4ac}}{2a} \]

“flip” avoids cancellation
Series Expansions

\[-b + \frac{\sqrt{b^2}}{2a}\]

"flip" avoids cancellation

but what about overflow?
Series Expansions

\[-b + \frac{\sqrt{b^2} - \frac{b}{2a}}{2a}\]

"flip" avoids cancellation

but what about overflow?

Sometimes no rewrite in the Herbie rule DB will improve error.

Taking the series expansion often provides error-reducing approx.
Series Expansions

\[-b + \frac{\sqrt{b^2 - 4ac}}{2a}\]
Series Expansions

\[-b + \frac{\sqrt{b^2 - 4ac}}{2a}\]

\[b > 0 \rightarrow \infty\]
Series Expansions

\[-b + \frac{\sqrt{b^2 - 4ac}}{2a}\]

\[= \frac{-b + b\sqrt{1 - \frac{4ac}{b^2}}}{2a}\]

\[b > 0 \rightarrow \infty\]
Series Expansions

\[
\frac{-b + \sqrt{b^2 - 4ac}}{2a}
\]

\[= \frac{-b + b\sqrt{1 - 4ac/b^2}}{2a}\]

\[b > 0 \rightarrow \infty\]

\[\sqrt{1 - x} \approx 1 - x/2\]
Series Expansions

\[-b + \frac{\sqrt{b^2 - 4ac}}{2a}\]

\[= \frac{-b + b\sqrt{1 - \frac{4ac}{b^2}}}{2a}\]

\[\approx \frac{-b + b(1 - \frac{4ac}{2b^2})}{2a}\]

\[= \frac{-4ac/2b}{2a}\]

\[= -\frac{c}{b}\]
Series Expansions

\[-b + \sqrt{b^2 - 4ac} \over 2a\]

\[= \frac{-b + b\sqrt{1 - 4ac/b^2}}{2a}\]

\[\approx \frac{-b + b(1 - 4ac/2b^2)}{2a}\]

\[= \frac{-4ac/2b}{2a}\]

\[= \frac{-c}{b}\]

\[b > 0 \rightarrow \infty\]

\[\sqrt{1 - x} \approx 1 - x/2\]

Custom series expander:
- auto expands diverse exprs
- determines # terms to take
- expand around arbitrary pt
Herbie Architecture

$e \xrightarrow{\text{sample}} \text{cands} \xrightarrow{\text{regimes}} e'$

- focus
- rewrite
- simplify
- series
Herbie Architecture

Sample \(e\) \(\rightarrow\) Cands \(\rightarrow\) Regimes \(\rightarrow\) \(e'\)

Focus
Rewrite
Simplify
Series
Herbie Architecture

Only keep “good” candidates

sample → cands → regimes → e'
Herbie Architecture

1. Sample input $e$ to generate candidates $cands$.
2. Apply regimes to transform $cands$ into $e'$.
3. Focus on specific aspects to refine $e'$.
4. Rewrite expressions for better clarity.
5. Simplify complex expressions.
6. Organize series for efficient computation.
Regime Inference
Regime Inference

\[
\frac{c}{b} - \frac{b}{a} = \frac{-b + \sqrt{b^2 - 4ac}}{2a} = \frac{2c}{-b - \sqrt{b^2 - 4ac}} = -\frac{c}{b}
\]
Regime Inference

\[
\frac{c}{b} - \frac{b}{a}
\]

\[
\frac{-b + \sqrt{b^2 - 4ac}}{2a}
\]

\[
\frac{2c}{-b - \sqrt{b^2 - 4ac}}
\]

\[
-\frac{c}{b}
\]
Regime Inference

\[
\frac{c}{b} - \frac{b}{a} \quad \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \frac{2c}{-b - \sqrt{b^2 - 4ac}} \quad -\frac{c}{b}
\]
Regime Inference

\[
\begin{align*}
\frac{c}{b} & - \frac{b}{a} \\
= & -\frac{b + \sqrt{b^2 - 4ac}}{2a} \\
= & \frac{2c}{-b - \sqrt{b^2 - 4ac}} \\
= & -\frac{c}{b}
\end{align*}
\]
Regime Inference

\[
\begin{cases}
\frac{c}{b} - \frac{b}{a} & \text{if } b \in (\infty, -1.15\times10^{12}] \\
\frac{-b + \sqrt{b^2 - 4ac}}{2a} & \text{if } b \in (-1.125\times10^{12}, 1.06\times10^{-304}] \\
\frac{2c}{-b - \sqrt{b^2 - 4ac}} & \text{if } b \in (1.06\times10^{-304}, 4.62\times10^{63}] \\
-\frac{c}{b} & \text{if } b \in (4.62\times10^{63}, \infty)
\end{cases}
\]

Dynamic programming:
- Bounds quickly
- Tune: binary search
- Pick best variable
Regime Inference

\[
\begin{cases}
\frac{c}{b} - \frac{b}{a} & \text{if } b \in (\infty, -1.15E122] \\
-\frac{b + \sqrt{b^2 - 4ac}}{2a} & \text{if } b \in (-1.125E122, 1.06E-304]\n\frac{2c}{-b - \sqrt{b^2 - 4ac}} & \text{if } b \in (1.06E-304, 4.62E63] \\
-\frac{c}{b} & \text{if } b \in (4.62E63, \infty)
\end{cases}
\]

Dynamic programming:
- Bounds quickly
- Tune: binary search
- Pick best variable
Herbie Architecture

e → sample → cands → regimes → e'  
|   ▲                         ▲                  ▲     |  
|     ▼                         ▼                  ▼     |  
|   focus → rewrite → simplify → series   |  

e' → eval
While writing an MCMC sampler for a semi-parametric clustering model, I needed to calculate several likelihood ratios and posterior parameters. Each was relatively complicated, and using the naive formulas caused divide-by-zero errors, but I wasn't sure how to best rewrite the equations. Herbie found numerically stable versions of the formulas, and fixed all the divide-by-zero errors.

Clustering (bigger, darker blocks better)
ML Anecdote

\[ \frac{(\text{sig } s)^{c_p} (1 - \text{sig } s)^{c_n}}{(\text{sig } t)^{c_p} (1 - \text{sig } t)^{c_n}}, \text{ where } \text{sig } x = \frac{1}{1 + e^{-x}} \]

\[ \exp \left( c_p \ln \frac{1 + e^{-t}}{1 + e^{-s}} + c_n \ln \frac{1 - \frac{1}{1 + e^{-s}}}{1 - \frac{1}{1 + e^{-t}}} \right) \]

Clustering (bigger, darker blocks better)  Harley Montgomery
Herbie Implementation

Enter a formula below, hit Enter, and Herbie will try to improve it.

\[
\sqrt{x + 1} - \sqrt{x}
\]

To handle the high volume of requests, web requests are queued; there are 14 jobs in the queue right now. Web demo requests may also time out and cap the number of improvement iterations. To

herbie.uwplse.org

\sim 10 \text{ KLOC in Racket}
Enriching a Diverse Ecosystem

FPBench
Standards and benchmarks for floating-point research
Home  Benchmarks  Compilers  Standards

FPBench provides benchmarks, compilers, and standards for the floating-point research community.

\[ \sqrt{x + 1} - \sqrt{x} \]

(FPCore (x)
:name “Sqrt Difference”
:cite (hamming-87)
:pre (> x 0)
(- (sqrt (+ x 1)) (sqrt x)))

Benchmarks
Browse
Contribute

Compilers
Download
Documentation

Standards
Read
Implement

fpbench.org
Today’s Menu

1. Herbie: Automated FP Accuracy Improvement

   ![Herbie Diagram](image)

2. egg: Fast, Flexible Equality Saturation

   ![Egg Diagram](image)

3. MPMF: Multi-precision, Multi-format Numerics

   ![MPMF Diagram](image)
Today’s Menu

1. Herbie: Automated FP Accuracy Improvement

2. egg: Fast, Flexible Equality Saturation

3. MPMF: Multi-precision, Multi-format Numerics
Equality Saturation in egg
Equality Saturation in egg

E-graphs originally developed within the automated theorem proving community.

Used to in congruence procedures to efficiently represent many equivalent terms.
Equality Saturation in egg

E-graphs originally developed within the automated theorem proving community.

Used to in congruence procedures to efficiently represent many equivalent terms.

\[ a \equiv b \implies f(a) \equiv f(b) \]
Equality Saturation in egg

E-graphs originally developed within the automated theorem proving community.

Equality saturation repurposes e-graphs to program optimization.
Equality Saturation in egg

```python
def equality_saturation(expr, rewrites):
    egraph = initial_egraph(expr)

    while not egraph.is_saturated_or_timeout():
        for rw in rewrites:
            for (subst, eclass) in egraph.ematch(rw.lhs):
                eclass2 = egraph.add(rw.rhs.subst(subst))
                egraph.merge(eclass, eclass2)

    return egraph.extract_best()
```
Equality Saturation in egg

```
1  def equality_saturation(expr, rewrites):
2     egraph = initial_egraph(expr)
3
4     while not egraph.is_saturated_or_timeout():
5
6         for rw in rewrites:
7             for (subst, eclass) in egraph.ematch(rw.lhs):
8                 eclass2 = egraph.add(rw.rhs.subst(subst))
9                 egraph.merge(eclass, eclass2)
10            return egraph.extract_best()
```
Equality Saturation in egg

Simple: ~ 10 line algorithm!

Extensible: add domain-specific rewrites

```python
1  def equality_saturation(expr, rewrites):
2       egraph = initial_egraph(expr)
3
4       while not egraph.is_saturated_or_timeout():
5
6           for rw in rewrites:
7               for (subst, eclass) in egraph.ematch(rw.lhs):
8                   eclass2 = egraph.add(rw.rhs.subst(subst))
9                   egraph.merge(eclass, eclass2)
10
11       return egraph.extract_best()
```
Equality Saturation in egg

Simple: ~ 10 line algorithm!

Extensible: add domain-specific rewrites

Extensible: use domain-specific cost functions

```python
1  def equality_saturation(expr, rewrites):
2      egraph = initial_egraph(expr)
3
4      while not egraph.is_saturated_or_timeout():
5
6          for rw in rewrites:
7              for (subst, eclass) in egraph.ematch(rw.lhs):
8                  eclass2 = egraph.add(rw.rhs, subst)
9                  egraph.merge(eclass, eclass2)
10
11      return egraph.extract_best()
```
Equality Saturation in egg

Simple: ~ 10 line algorithm!

Extensible: add domain-specific rewrites

Complete: saturation implies you’ve found all equivalent programs.

Extensible: use domain-specific cost functions

```python
1 def equality_saturation(expr):
2     egraph = initial_egraph(expr)
3
4     while not egraph.is_saturated_or_timeout():
5         for rw in rewrites:
6             for (subst, eclass) in egraph.ematch(rw.lhs):
7                 eclass2 = egraph.add(rw.rhs, subst)
8                 egraph.merge(eclass, eclass2)
9
10 return egraph.extract_best()
```
Needs of Equality Saturation

Need to scale: e-graphs never forget.

Unlike SMT, equality saturation never makes a mistake.
What’s the bottleneck? Congruence!
What’s the bottleneck? Congruence!
What’s the bottleneck? Congruence!

Violates key congruence invariant!
What’s the bottleneck? Congruence!

Violates key congruence invariant!
What’s the bottleneck? Congruence!

Violates key congruence invariant!

Congruence restored.

Restoring dedup invariant also easy now!
Step 1: Phased EQSAT

```python
def equality_saturation(expr, rewrites):
    egraph = initial_egraph(expr)

    while not egraph.is_saturated_or_timeout():

        # reading and writing is mixed
        for rw in rewrites:
            for (subst, eclass) in egraph.ematch(rw.lhs):

                # in traditional equality saturation,
                # matches can be applied right away
                # because invariants are always maintained
                eclass2 = egraph.add(rw.rhs.subst(subst))
                egraph.merge(eclass, eclass2)

                # restore the invariants after each merge
                egraph.rebuild()

        return egraph.extract_best()
```

Traditional:
Always maintain invariants
Step 1: Phased EQSAT

```python
def equality_saturation(expr, rewrites):
    egraph = initial_egraph(expr)

    while not egraph.is_saturated_or_timeout():

        # reading and writing is mixed
        for rw in rewrites:
            for (subst, eclass) in egraph.ematch(rw.lhs):

                # in traditional equality saturation,
                # matches can be applied right away
                # because invariants are always maintained
                eclass2 = egraph.add(rw.rhs.subst(subst))
                egraph.merge(eclass, eclass2)

                # restore the invariants after each merge
                egraph.rebuild()

        return egraph.extract_best()
```

Traditional:
Always maintain invariants
Step 1: Phased EQSAT

**Traditional:**
Always maintain invariants

**Phased:**
Separate reads, writes. Restore inv each iter.
Step 1: Phased EQSAT

**Traditional:**
Always maintain invariants

**Phased:**
Separate reads, writes. Restore inv each iter.

```
1 def equality_saturation(expr, rewrites):
2     egraph = initial_egraph(expr)
3
4     while not egraph.is_saturated_or_timeout():
5
6         # reading and writing is mixed
7         for rw in rewrites:
8             for (s, ec) in egraph.ematch(rw.lhs):
9                 subst = ec
10                # invariants can be applied right away
11                # because invariants are always maintained
12                eclass2 = egraph.add(rw.rhs.subst(subst))
13                egraph.merge(eclass, eclass2)
14
15         # restore the invariants after each merge
16         egraph.rebuild()
17
18     return egraph.extract_best()
```

```
1 def equality_saturation(expr, rewrites):
2     egraph = initial_egraph(expr)
3
4     while not egraph.is_saturated_or_timeout():
5         matches = []
6
7         # read-only phase, invariants are preserved
8         for rw in rewrites:
9             for (subst, eclass) in egraph.ematch(rw.lhs):
10                matches.append((rw, subst, eclass))
11
12         # write-only phase, temporarily break invariants
13         for (rw, subst, eclass) in matches:
14             eclass2 = egraph.add(rw.rhs.subst(subst))
15             egraph.merge(eclass, eclass2)
16
17         # restore the invariants once per iteration
18         egraph.rebuild()
19
20     return egraph.extract_best()
```
Step 1: Phased EQSAT

**Traditional:**
Always maintain invariants

**Phased:**
Separate reads, writes.
Restore inv each iter.

```python
def equality_saturation(expr, rewrites):
    egraph = initial_egraph(expr)

    while not egraph.is_saturated_or_timeout():
        # reading and writing is mixed
        for rw in rewrites:
            for (subt, subst) in egraph.ematch(rw.lhs):
                # because invariants are always maintained
                eclass2 = egraph.add(rw.rhs.subst(subst))
                eclass = eclass2
                egraph.merge(eclass, eclass2)
                # restore the invariants once per iteration
                egraph.rebuild()
        return egraph.extract_best()
```

```python
def equality_saturation(expr, rewrites):
    egraph = initial_egraph(expr)

    while not egraph.is_saturated_or_timeout():
        matches = []

        # read-only phase, invariants are preserved
        for rw in rewrites:
            for (subt, eclass) in egraph.ematch(rw.lhs):
                matches.append((rw, subt, eclass))

        # write-only phase, temporarily break invariants
        for (rw, subt, eclass) in matches:
            eclass2 = egraph.add(rw.rhs.subst(subt))
            eclass = eclass2
            egraph.merge(eclass, eclass2)

        # restore the invariants once per iteration
        egraph.rebuild()
        return egraph.extract_best()```

Compute ALL matches first. (read only)

Then batch all modifications.
Step 1: Phased EQSAT

Traditional:
Always maintain invariants

Phased:
Separate reads, writes.
Restore inv each iter.

```python
def equality_saturation(expr, rewrites):
    egraph = initial_egraph(expr)

    while not egraph.is_saturated_or_timeout():
        # reading and writing is mixed
        for rw in rewrites:
            for (sub, eclass) in egraph.ematch(rw.lhs):
                if not egraph.is_sat_or_timeout():
                    eclass2 = egraph.add(rw.rhs.subst(subst))
                    egraph.merge(eclass, eclass2)
                    if egraph.is_sat_or_timeout():
                        break
            matches.append((rw, subst, eclass))

        egraph.reset()

    return egraph.extract_best()
```
Step 2: Rebuilding

```python
def rebuild():
    while self.worklist.len() > 0:
        # empty the worklist into a local variable
        todo = take(self.worklist)
        # canonicalize and deduplicate the eclass refs
        # to save calls to repair
        todo = { self.find(eclass) for eclass in todo }
        for eclass in todo:
            self.repair(eclass)

def repair(eclass):
    # update the hashcons so it always points
    # canonical enodes to canonical eclasses
    for (p_node, p_eclass) in eclass.parents:
        self.hashcons.remove(p_node)
        p_node = self.canonicalize(p_node)
        self.hashcons[p_node] = self.find(p_eclass)

    # deduplicate the parents, noting that equal
    # parents get merged and put on the worklist
    new_parents = {}
    for (p_node, p_eclass) in eclass.parents:
        p_node = self.canonicalize(p_node)
        if p_node in new_parents:
            self.merge(p_eclass, new_parents[p_node])
        new_parents[p_node] = self.find(p_eclass)
    eclass.parents = new_parents
```

Key: cache-friendly data structures!

Possible b/c no need for rollback.
Does it work?
Does it work?
Does it work?
Does it work?

Yes.
Does it work?

Yes.
Case Study: Herbie

Find and fix floating-point problems.

Try  ·  Install  ·  Learn

\[ \sqrt{x+1} - \sqrt{x} \rightarrow \frac{1}{\sqrt{x+1} + \sqrt{x}} \]

Herbie detects inaccurate expressions and finds more accurate replacements. The red expression is inaccurate when \( x > 1 \); Herbie’s replacement, in blue, is accurate for all \( x \).

https://herbie.uwplse.org
Case Study: Herbie

- Initial Racket implementation: 5022.0 minutes (98.1% of total run time)
- + batching: 49.4 minutes (68.7% of remaining time)
- + rebuilding: 22.4 minutes (48.8% of remaining time)
- egg: 1.4 minutes (4.8% of remaining time)
Case Study: Herbie

Faster and better.
More egg Users

Optimize linear algebra via relational algebra.

[VLDB 20]
More egg Users

Optimize linear algebra via relational algebra.
[VLDB 20]
More egg Users

Optimize linear algebra via relational algebra.

[VLDB 20]
Decompile 3D CAD to parameterized designs.

[PLDI 20]
More egg Users

Decompile 3D CAD to parameterized designs.

[PLDI 20]
2127 programs from Thingiverse:
- Tiny size < 30 (769)
- Small size < 100 (786)
- Med size < 300 (374)
- Large size > 300 (198)

Larger programs shrink more

All optimizations take < 1 second
More egg Users

2127 programs from Thingiverse:
- Tiny size < 30 (769)
- Small size < 100 (786)
- Med size < 300 (374)
- Large size > 300 (198)

Larger programs shrink more

All optimizations take < 1 second

[PLDI 20]
egg: Fast, Flexible Equality Saturation

The egg project uses e-graphs to provide a new way to build program optimizers and synthesizers. egg is developed by Max Willsey and his friends on GitHub.

https://egraphs-good.github.io/
Today’s Menu

1. **Herbie: Automated FP Accuracy Improvement**

![Herbie Diagram](image1)

2. **egg: Fast, Flexible Equality Saturation**

![Egg Diagram](image2)

3. **MPMF: Multi-precision, Multi-format Numerics**

![MPMF Diagram](image3)
Today’s Menu

1. **Herbie: Automated FP Accuracy Improvement**

2. **egg: Fast, Flexible Equality Saturation**

3. **MPMF: Multi-precision, Multi-format Numerics**
IBM 360/44
Precision vs. Accuracy Landscape

Overall precision: posit(8,1)

Accumulator precision

12

3

# iterations

2

50
Precision vs. Accuracy Landscape

Overall precision: \text{posit}(8,1)
Precision vs. Accuracy Landscape

Overall precision: posit(16,1)
Precision vs. Accuracy Landscape

FPBench DSL supports specifying this landscape!

fpbench.org
Precision vs. Accuracy Landscape

FPBench DSL supports specifying this landscape!

fpbench.org

Few tools can exploit MPMF tradeoffs :\
Why MPMF?

Many PLs only offer double precision IEEE754. “Doubles are basically as fast.”

Misses the point!
“Basically as fast” only for tiny amounts of data!

What about ML and HPC applications running on accelerators and in the cloud?!
Why MPMF?

Many PLs only offer double precision IEEE754. "Doubles are basically as fast." Misses the point! "Basically as fast" only for tiny amounts of data! What about ML and HPC applications running on accelerators and in the cloud?!
What’s the MPMF spread?
What’s the MPMF spread?

Titanic: Tools for specifying and simulating MPMF behavior across diverse configs.

http://titanic.uwplse.org/

Bill Zorn
What’s the MPMF spread?
What’s the MPMF spread?

Large gap between standard fixed formats and custom.
What’s the MPMF spread?

Significant gaps between traditional and “next generation” formats.

Large gap between standard fixed formats and custom.
What’s the MPMF spread?

Explore Pareto Frontier across accuracy, bandwidth, uniformity, etc.
What’s the MPMF spread?
What’s the MPMF spread?

But what if we could change the computation...
Why Search for MPMF?

\[ \frac{-b + \sqrt{b^2 - 4ac}}{2a} \]

\[ = \left( \frac{(-b)^2 - (\sqrt{b^2 - 4ac})^2}{-b - \sqrt{b^2 - 4ac}} \right) / 2a \]

\[ = \left( \frac{b^2 - (\sqrt{b^2 - 4ac})^2}{-b - \sqrt{b^2 - 4ac}} \right) / 2a \]

\[ = \left( \frac{b^2 - (b^2 - 4ac)}{-b - \sqrt{b^2 - 4ac}} \right) / 2a \]

\[ = \left( \frac{4ac}{-b - \sqrt{b^2 - 4ac}} \right) / 2a \]

\[ = \frac{2c}{-b - \sqrt{b^2 - 4ac}} \]
Why Search for MPMF?

\[
\frac{-b + \sqrt{b^2 - 4ac}}{2a}
\]

Canellation: need 40 bits to get 8!
Why Search for MPMF?

\[-\frac{b + \sqrt{b^2 - 4ac}}{2a}\]

Canellation: need 40 bits to get 8!

\[
\begin{align*}
\frac{(-b)^2 - (\sqrt{b^2 - 4ac})^2}{-b - \sqrt{b^2 - 4ac}} / 2a \\
\frac{b^2 - (\sqrt{b^2 - 4ac})^2}{-b - \sqrt{b^2 - 4ac}} / 2a \\
\frac{b^2 - (b^2 - 4ac)}{-b - \sqrt{b^2 - 4ac}} / 2a \\
\frac{4ac}{-b - \sqrt{b^2 - 4ac}} / 2a \\
\frac{2c}{-b - \sqrt{b^2 - 4ac}}
\end{align*}
\]

No cancellation in target range. Can get 8 bits computing with ~ 10!
MPMF Herbie

e \rightarrow \text{sample} \rightarrow \text{cands} \rightarrow \text{regimes} \rightarrow e'

\downarrow f\downarrow r\downarrow s\downarrow e

focus, rewrite, simplify, series
\[
\frac{a - b\sqrt{2}}{a - 4b^2}
\]
MPMF Herbie

Flip local error on its head!
Reduce prec when local error low.
MPMF Herbie

Flip local error on its head!
Reduce prec when local error low.

double -> bfloat
MPMF Herbie

Flip local error on its head! Reduce prec when local error low.

Modular plugin infra:
- easily add new formats
- customize rewrites to domain
- JIT rewrite gen for param fmts
- posit, bfloat, etc.
- find Pareto-optimal configs

double -> bfloat
Flip local error on it's head!
Reduce prec when local error low.

Modular plugin infra:
- easily add new formats
- customize rewrites to domain
- JIT rewrite gen for param fmts
- posit, bfloat, etc.
- find Pareto-optimal configs

Massive search space…

double -> bfloat
Flip local error on it’s head! Reduce prec when local error low.

Modular plugin infra:
- easily add new formats
- customize rewrites to domain
- JIT rewrite gen for param fmts
- posit, bfloat, etc.
- find Pareto-optimal configs

Massive search space...
double -> bfloat
MPMF Herbie: Vision

Let engineers program against \( \mathbb{R} \)

Today: tensorize and match available accelerator primitives via equality saturation.

Tomorrow: find Pareto-optimal designs via Herbie’s tunable MPMF search.
Thank You!

1. Herbie: Automated FP Accuracy Improvement

2. egg: Fast, Flexible Equality Saturation

3. MPMF: Multi-precision, Multi-format Numerics