Eiffel: Inferring Input Ranges of Significant Floating-point Errors via Polynomial Extrapolation

Zuoyan Zhang, Bei Zhou, Jiangwei Hao, Hongru Yang, Mengqi Cui
Yuchang Zhou, Guanghui Song, Fei Li, Jinchen Xu, and Jie Zhao

Information Engineering University
Outline

- What is floating-point error?
- Error detection is important
- Existing approaches
Floating-point Errors

- Some inputs may trigger significant FP errors
- Consider:

\[ f(x) = \frac{\tan(x) - \sin(x)}{x^3} \]

\[ \lim_{x \to 0} f(x) = 0.5 \]

double f(double x) {
    double num = tan(x) - sin(x);
    double den = x * x * x;
    return num / den;
}

>>> f(1e-7) // 64 bits result
0.5029258124322410

Accurate result //128 bits result
0.5000000000000012
Error Detection is Crucial

- FP errors are infamous problem in software development

- Large rounding errors may lead to catastrophic software failures
  - Missile yaw [Skeel ’92]
  - Stock trading disorder [Quinn ’83]
  - Rocket launch failure [Lions ’96]
Existing approaches

Static analysis
- Abstract interpretation
- Symbolic execution
- Interval arithmetic
- Affine arithmetic
- ...

Goal: Approximate error bounds
- Over-approximated

Dynamic analysis
- Random search
- Binary guided random testing (BGRT)
- Atomic condition
- ...

Goal: Find the maximal error
+ Real error
Outline

- What is floating-point error?
- Error detection is important
- Existing approaches

- What is guided search?
- Difficulties for guided searches
Difficulties for guided searches

**Process**

- Search space: \( D = \{(x, y) : c_1 \land c_2 \land c_3\} \)
- \( c_1, c_2, c_3 \) are constraints of the two variables \( x \) and \( y \);
- The points within \( r \) are input values that may trigger significant errors;
- \( p \) is the input that triggers the maximal error.

**Goal:** Find the point \( p \)

**Difficulties**

- \( D \) may be complex and large
- \( r \) and \( p \) could both be many
Outline

BACKGROUND

• What is floating-point error?
• Error detection is important
• Existing approaches

DIFFICULTIES

• What is guided search?
• Difficulties for guided searches

APPROACH

• Error analysis using EIFFEL
• How EIFFEL works?

EVALUATION
Error analysis using EIFFEL

Core idea of EIFFEL: Inferring error-inducing ranges instead of searching them in $D$

$f(x), D$

$e.g.: f(x) = \frac{1}{\sqrt{x} + 1 + \sqrt{x}}$

Data set construction across $R$
- Using ULP error
- $R$ smaller than $D$

Data clustering
- Using DBSCAN algorithm

Curve derivation
- Using geometric progression formula

Extrapolation

Sampling in $R$

Maximal error
Data set construction

Two issues
• Determine $R$
  • FP numbers are non-uniformly distributed
    • $[-1,1]$ 49.95% (double type)
    • Dense near 0
  $\implies$ Small interval as close to 0

• Deciding the number of input values $s$ to compute the $ULP$ errors, considers
  • Performance $\implies s = 500,000$ (0.17 seconds)

• Accurate error distribution $\implies$ Uniformly distributed inputs
Boundary extraction and data clustering

• Boundary extraction
  • Reduce the number of data points to accelerate clustering
    Step 1: Gather every $g = 500$ samples in one group
    Step 2: Preserve the one that has the largest $ULP$ error in every group

• Data clustering
  • Obtain the maximal error point for each cluster for fitting the function
    • DBSCAN clustering algorithm
Curve derivation and polynomial extrapolation

• Curve derivation
  • Two challenges
    • Each peak point is a 2D coordinate
    • Extrapolated points may still follow the same distribution as that of the stars
  • Solution
    • 2D coordinates into 1D form
      
      \((x_1, \text{error}_1), (x_2, \text{error}_2) \ldots (x_i, \text{error}_i) \rightarrow (1, x_1), (2, x_2) \ldots (i, x_i)\)
    • Assume the \(x_i\) form a geometric sequence or a geometric progression
      
      \[x_i = x_1 \times q^{i-1} \quad i \geq 1\]

The assumption is observed from extensive experiments

• Derive two curves
  • Best fit for lower bound \(x_i - r\) and upper bound \(x_i + r\) that covers each \(x_i\)
Curve derivation and polynomial extrapolation

- Polynomial extrapolation
Error detection

- $num = 1$ or $num = 2$
  - Return maximal error across $R$

- $num \geq 3$
  - Consider the monotonicity
    - $error_1 < error_2 < \cdots$
    - $error_1 > error_2 > \cdots$
  - Polynomial extrapolation
Generalization for multi-variate scenarios

Similar to the single-variate case

• Two adaptations
  • Project the multi-dimensional plot onto the variable space
  
  ➙ Produce the variable coordinates of the errors
  
  • Perform curve fitting along one dimension each time
  
  ➙ Produce (hyper-)rectangular ranges but is still more effective than existing approaches
Outline

BACKGROUND
- What is floating-point error?
- Error detection is important
- Existing approaches

DIFFICULTIES
- What is guided search?
- Difficulties for guided searches

APPROACH
- Error analysis using EIFFEL
- How EIFFEL works?

EVALUATION
- How effective?
- How quality and quantity?
- How overhead?
Evaluation

• **Benchmarks**: total 70 expressions
  • 66 expressions are from FPBench
  • 4 expressions are from real-life numerical programs

• $D$ is set using large but reasonable ranges

<table>
<thead>
<tr>
<th>Total Benchmarks</th>
<th>Single-variate</th>
<th>Multi-variate</th>
</tr>
</thead>
<tbody>
<tr>
<td>70</td>
<td>30</td>
<td>40</td>
</tr>
</tbody>
</table>
Evaluation — Effectiveness

• Effectiveness
  • Compared with the state-of-the-art techniques

<table>
<thead>
<tr>
<th>Techniques</th>
<th>Number of errors detected</th>
</tr>
</thead>
<tbody>
<tr>
<td>EIFFEL</td>
<td>70</td>
</tr>
<tr>
<td>S3FP</td>
<td>43</td>
</tr>
<tr>
<td>ATOMU</td>
<td>30</td>
</tr>
</tbody>
</table>

ATOMU is only able to report errors for the 30 single-variate examples

S3FP returns empty results for 27 benchmarks
Evaluation — Inferred input ranges

• Quantity
  • Regina fails to infer input ranges at a large size of $D$
  • EIFFEL obtains more input ranges than PSAT

• Quality
  • Feed the inferred input ranges to Herbie

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Number</th>
<th>EIFFEL</th>
<th>PSAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>predatorPrey</td>
<td>15</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>sqrt_add</td>
<td>7</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>verhulst</td>
<td>12</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>nonlin1_test2</td>
<td>14</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Intro-example</td>
<td>14</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>NMSEexample35</td>
<td>15</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>NMSEexample37</td>
<td>27</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>carbonGas</td>
<td>21</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>
Time overhead is between ATOMU and S3FP

https://github.com/zuoyanzhang/EIFFEL