# VERROU : backtrace delta-debug: application to medcoupling

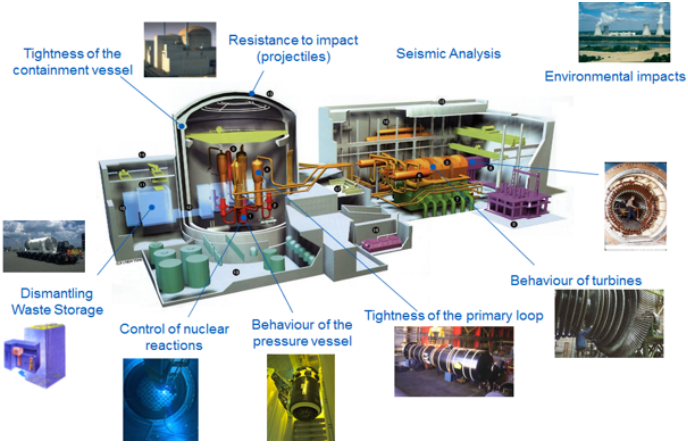
**Bruno Lathuilière**

**Supported by ANR Interflop ANR-20-CE46-0009 and FPT4 ANR-24-CE46-7572.**

EDF R&D , 26th march 2026

# Numerical simulation at EDF

- Guarantee safety
- Improve performances/costs
- Ageing issues



# Numerical software at EDF

**Large number of in-house codes:**

- code_aster (Thermo-mechanic)
- code_saturne (CFD)
- open_telemac (free surface flow)
- salome (Simulation plateform)
- ...

**Code properties:**

- Huge code base
- Various languages (C/C++, Fortran, Python ...)
- External libraries (MUMPS, numpy ...)
- V&V

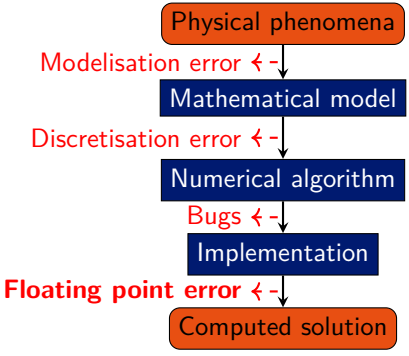
# Numerical software at EDF

**Large number of in-house codes:**

- code_aster (Thermo-mechanic)
- code_saturne (CFD)
- open_telemac (free surface flow)
- salome (Simulation plateform)
- ...

**Code properties:**

- Huge code base
- Various languages (C/C++, Fortran, Python ...)
- External libraries (MUMPS, numpy ...)
- **V&V**




**VERROU** development:

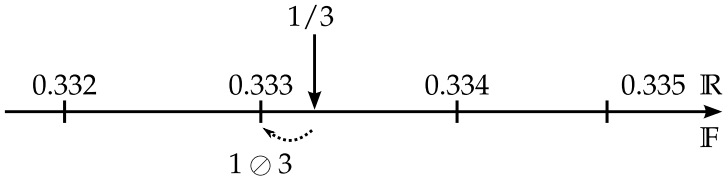
- Binary instrumentation based on valgrind
- Asynchronous stochastic arithmetic
- **Error estimation** and **error localization**

# PLAN

- Context
- Verrou and stochastic rounding
- Application to medcoupling
- verrou_dd_back development overview

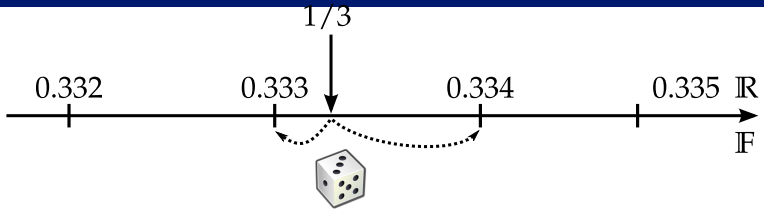
# Floating point error

- Floating point representation with limited precision
  - [`float`] binary, 24 significand bits ($\simeq 10^{-7}$)
  - [`double`] binary, 53 significand bits ($\simeq 10^{-16}$)
  - [pedagogic example] decimal, 3 significand digits (% – ‰)

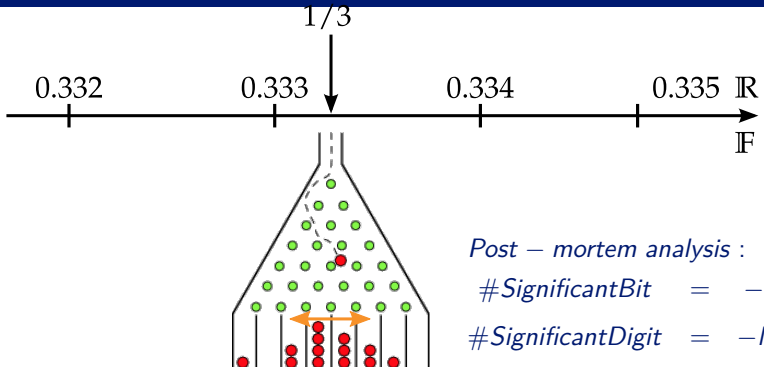



- Floating point computation $\neq$ Real computation
  - rounding error $a \oplus b \neq a + b$
  - associativity loss $(a \oplus b) \oplus c \neq a \oplus (b \oplus c)$

# Stochastic arithmetic for numerical verification



# Stochastic arithmetic for numerical verification




*Post − mortem analysis* :

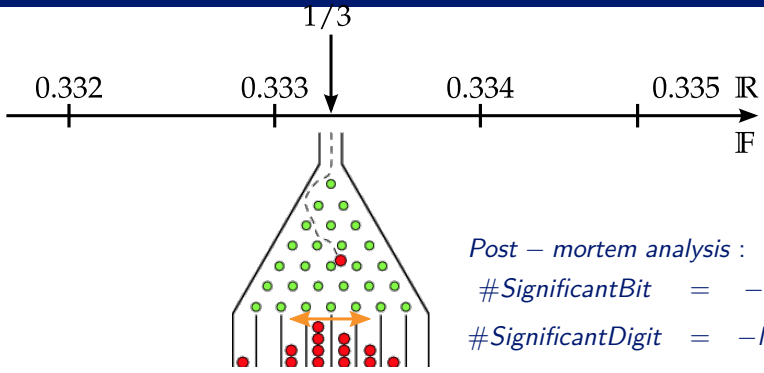
$$\#SignificantBit = -log2\left(\frac{max_i(|X_i - X_{nearest}|)}{|X_{nearest}|}\right)$$

$$\#SignificantDigit = -log10\left(\frac{max_i(|X_i - X_{nearest}|)}{|X_{nearest}|}\right)$$

| Instruction | Eval. Nearest | Eval. 1 | Eval. 2 | Eval. 3 |
|---|---|---|---|---|
| $a = 1/3$ | 0.333 | $0.333_{\downarrow}$ | $0.334^{\uparrow}$ | $0.334^{\uparrow}$ |
| $b = a \times 3$ | 0.999 | 0.999 | $1.00_{\downarrow}$ | $1.01^{\uparrow}$ |

$\#SignificantDigit \approx 1.95$

# Stochastic arithmetic for numerical verification




*Post − mortem analysis* :

$$\#SignificantBit = -log2\left(\frac{max_i(|X_i - X_{nearest}|)}{|X_{nearest}|}\right)$$

$$\#SignificantDigit = -log10\left(\frac{max_i(|X_i - X_{nearest}|)}{|X_{nearest}|}\right)$$

| Instruction | Eval. Nearest | Eval. 1 | Eval. 2 | Eval. 3 |
|---|---|---|---|---|
| $a = 1/3$ | 0.333 | $0.333_{\downarrow}$ | $0.334^{\uparrow}$ | $0.334^{\uparrow}$ |
| $b = a \times 3$ | 0.999 | 0.999 | $1.00_{\downarrow}$ | $1.01^{\uparrow}$ |

$\#SignificantDigit \approx 1.95$

- Compatible with binary instrumentation (**Verrou** based on valgrind) or LLVM pass (Verificarlo)
- Few false positive detection (due to asynchronous approach and dedicated stochastic rounding mode)

# Delta-debugging localization

1-Search space generation and reference computation.

| tool | search space |
|---|---|
| `verrou_dd_sym` | symbols |
| `verrou_dd_line` | code lines (if compiled with -g) |
| `verrou_dd_task` | user defined task |
| `verrou_dd_stdout` | task defined thanks to standard output |
| `verrou_dd_back` | backtrace addresses |
| `vfc_ddebug` | code lines (code instrumented with verificarlo) |

2-Verification:

- the full search space is perturbed: FAIL.
- the full search space is not perturbed: SUCCESS.

3-Search algorithm: try/fail approach.

# Delta-debug: Application to medcoupling

**Output** of `TestINTERP_KERNEL` :

.....

**Delta-debug search**:

```
1 verrou_dd_line --nruns=5 --num-threads=5 ddRun.sh ddCmp.sh
```

**runScript**: `ddRun.sh`

```
1 #!/bin/bash
2 valCmd="valgrind --tool=verrou --rounding-mode=sr_smonotonic --↩
      libm=instrumented"
3 export MEDCOUPLING_RESOURCE_DIR=./install/share/resources/med
4 export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:./install/lib
5 ${valCmd} ./install/bin/TestINTERP_KERNEL $1/testUnit.out > $1/res.dat
```

**cmpScript**: `ddCmp.sh`

```
1 #!/bin/bash
2 diff $1/res.dat $2/res.dat
```

45 min later...

# verrou_dd_line result 22 ddmin sets with a total of 43 lines

```
ddmin0: Fail Ratio: 100.00% Fail indexes: 0,1,2
Interlibmath/interlibmath.cxx:1690 (sqrt)

ddmin1: Fail Ratio: 100.00% Fail indexes: 0,1,2
InterpKernelGeo2DEdgeArcCircle.cxx:241 (INTERP_KERNEL::ArcCArcCIntersector::getIntersectionsCharacteristicVal[abi:cxx11]

ddmin2: Fail Ratio: 100.00% Fail indexes: 0,1,2
InterpKernelGeo2DEdgeArcCircle.cxx:739 (INTERP_KERNEL::EdgeArcCircle::GetArcOfCirclePassingThru(double const*, double co

ddmin3: Fail Ratio: 100.00% Fail indexes: 0,1,2
InterpKernelGeo2DEdgeArcCircle.cxx:738 (INTERP_KERNEL::EdgeArcCircle::GetArcOfCirclePassingThru(double const*, double co

ddmin4: Fail Ratio: 33.33% Fail indexes: 0
InterpKernelGeo2DNode.cxx:42 (INTERP_KERNEL::Node::Node(std::basic_istream<char, std::char_traits<char> >&))

ddmin5: Fail Ratio: 66.67% Fail indexes: 1,2
InterpKernelGeo2DEdgeArcCircle.cxx:737 (INTERP_KERNEL::EdgeArcCircle::GetArcOfCirclePassingThru(double const*, double co

ddmin6: Fail Ratio: 33.33% Fail indexes: 0
InterpKernelGeo2DEdgeArcCircle.cxx:736 (INTERP_KERNEL::EdgeArcCircle::GetArcOfCirclePassingThru(double const*, double co

ddmin7: Fail Ratio: 100.00% Fail indexes: 0,1,2
InterpKernelValue.cxx:156 (INTERP_KERNEL::ValueDouble::plus(INTERP_KERNEL::Value const*) const)

ddmin8: Fail Ratio: 33.33% Fail indexes: 1
Interlibmath/interlibmath.cxx:1688 (cos)
Interlibmath/interlibmath.cxx:1686 (sin)
....
```

# verrou_dd_back

```
1 VERROU_SOURCE=dd.line/ddmin7/dd.line.include  verrou_dd_back --nruns=5 --↩
      num-threads=5 --hide-common  ./run.sh ./cmp.sh
```

```
ddmin0: Fail Ratio: 66.67% Fail indexes: 0,1
36:88217265,88187398,88118786,88118634,88118634,88118634,88118634,88118634,88118634,88118634,88118634,88115643,

0    : INTERP_KERNEL::ValueDouble::plus(INTERP_KERNEL::Value const*) const+45 InterpKernelValue.cxx:156
1    : INTERP_KERNEL::PlusFunction::operate(std::vector<INTERP_KERNEL::Value*, std::allocator<INTERP_KERNEL::Value*> >&)
InterpKernelFunction.cxx:877
2    : INTERP_KERNEL::ExprParser::evaluateLowLev(INTERP_KERNEL::Value*) const+508 InterpKernelExprParser.cxx:499
3-11 : INTERP_KERNEL::ExprParser::evaluateLowLev(INTERP_KERNEL::Value*) const+356 InterpKernelExprParser.cxx:494
12   : INTERP_KERNEL::ExprParser::evaluate() const+83 InterpKernelExprParser.cxx:328
13   : INTERP_TEST::ExprEvalInterpTest::testInterpreter1()+8376 ExprEvalInterpTest.cxx:222
14-35: (common stack)
....
```

- search space size : 32
- 8 ddmin sets
- duration 1min35

# Test bug fix

Before:

```
218 INTERP_KERNEL::ExprParser expr22(
219  "3.*max(((3.2*(ln((2*5.2+6.)+(1.2*1.2+3.)))))),((3.2*(exp((6.+2*5.2)+(1.2*1.2+3.))))))"
220 );
221 expr22.parse();
222 CPPUNIT_ASSERT_DOUBLES_EQUAL(10788678114.535484, expr22.evaluate(), 1e-5);
```

After:

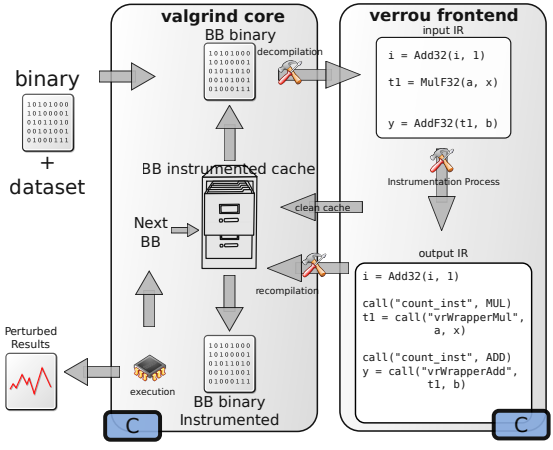
```
222 CPPUNIT_ASSERT_DOUBLES_EQUAL(10788678114.535484, expr22.evaluate(), 35.*epsilonDbl * 10788678114.535484);
```

How to find tolerance:

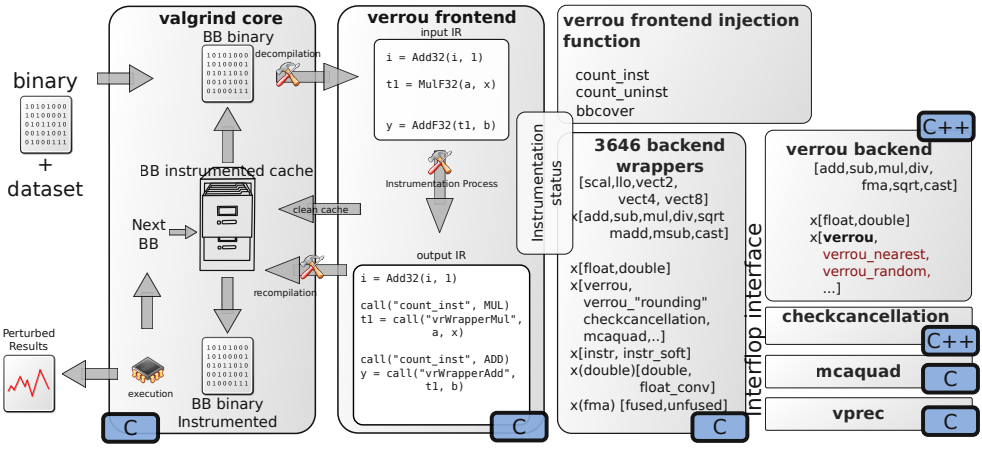
```
1 user@host:~/testDDMedCoupling$ valgrind --tool=verrou --quiet ↩
     --libm=instrumented python3
2 Python 3.9.2 (default, Mar 20 2025, 02:07:39)
3 [GCC 10.2.1 20210110] on linux
4 >>> from math import *
5 >>> import valgrind.verrouPyBinding as verrou
6 >>> verrou.compute_verrou_tolerance(lambda : eval(
7     ("3.*max(((3.2*(log((2*5.2+6.)+(1.2*1.2+3.)))))),((3.2*(exp((6.+2*5.2)+(1.2*1.2+3.))))))")
8     ))
9 (7.82012939453125e-05, 7.248459275094382e-15, 32.644158490325275)
```

Warning: compute_verrou_tolerance not integrated in verrou master branch.

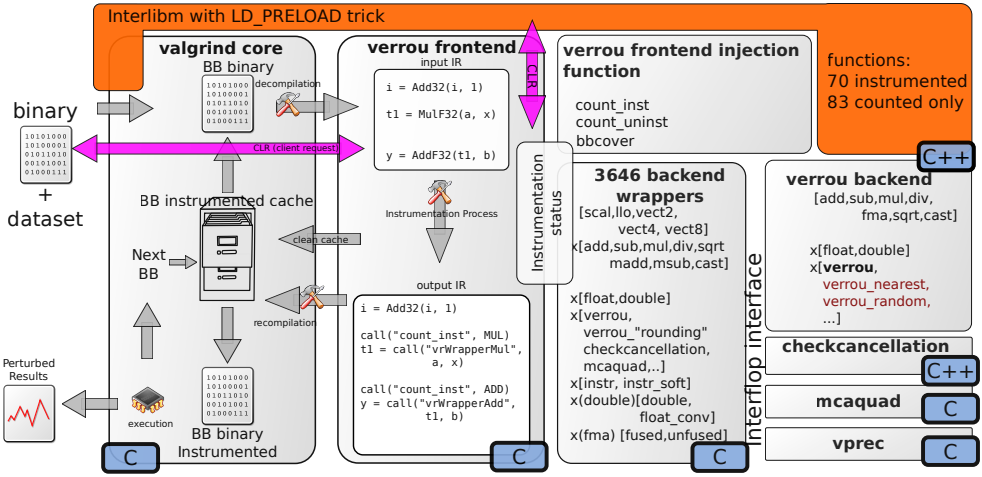
# A quick look inside verrou



# A quick look inside verrou



# A quick look inside verrou



# verrou backtrace implementation details

Delta-debug search space:

- Stack trace (list of addresses) of each instrumented BB ;
- Stack trace up to `below main` (with a maximum depth of 40) ;
- Disable ASLR with `personality(ADDR_NO_RANDOMIZE)` ;

Dynamic instrumentation status :

- Two instruction loops are needed to reduce code injection.
- Soft instrumentation status is required (it is not possible to clean the cache during BB execution).
- Easily compatible with Interlibm.

Expected problem:

- Two different binaries with the same backtrace

# Performance

| type | double | | float | |
|---|---|---|---|---|
| compilation option | O0 | O3 | O0 | O3 |
| tool_none | x6.4 | x6.7 | x7.2 | x8.5 |
| nearest | x9.8 | x23.4 | x11.5 | x28.3 |
| gen_exclude | x9.8 | x23.3 | x10.9 | x28.4 |
| exclude_all | x5.8 | x6.7 | x6.4 | x8.3 |
| gen_source | x9.8 | x22.5 | x11.0 | x28.4 |
| source_nothing | x5.8 | x6.5 | x6.4 | x8.3 |
| gen_backtrace | x37.8 | x42.3 | x47.7 | x53.3 |
| exclude_all_backtrace | x37.9 | x42.2 | x46.7 | x53.4 |
| random | x16.2 | x43.8 | x19.0 | x59.6 |

| type | double | | float | |
|---|---|---|---|---|
| option | O0 | O3 | O0 | O3 |
| #symboles | 4 | 2 | 4 | 2 |
| #lines | 29 | 25 | 29 | 27 |
| #back (max depth) | 15(4) | 17(3) | 19(4) | 18(3) |

# Performance

| type | double | | float | |
|---|---|---|---|---|
| compilation option | O0 | O3 | O0 | O3 |
| tool_none | x6.4 | x6.7 | x7.2 | x8.5 |
| nearest | x9.8 | x23.4 | x11.5 | x28.3 |
| gen_exclude | x9.8 | x23.3 | x10.9 | x28.4 |
| exclude_all | x5.8 | x6.7 | x6.4 | x8.3 |
| gen_source | x9.8 | x22.5 | x11.0 | x28.4 |
| source_nothing | x5.8 | x6.5 | x6.4 | x8.3 |
| gen_backtrace | x37.8 | x42.3 | x47.7 | x53.3 |
| exclude_all_backtrace | x37.9 | x42.2 | x46.7 | x53.4 |
| random | x16.2 | x43.8 | x19.0 | x59.6 |

| type | double | | float | |
|---|---|---|---|---|
| option | O0 | O3 | O0 | O3 |
| #symboles | 4 | 2 | 4 | 2 |
| #lines | 29 | 25 | 29 | 27 |
| #back (max depth) | 15(4) | 17(3) | 19(4) | 18(3) |

# Conclusions and perspectives

- VERROU provides an easy floating point error estimation ;
- VERROU provides error localization ;
- VERROU is used with industrial code ;
- VERROU is open-source and available: `https://github.com/edf-hpc/verrou`

## Perspectives

- `verrou_dd_back` optimizations:
  - data structure improvments (need feedback about usual backtrace size, and search space size).
  - reimplement and specialize `VG_(get_StackTrace)` function.
- Performance improvements (instrumentation, vectorization, delta-debug parallelism ...)
- Localization improvements (delta-debug search space, amplification detection ...)


Merci
Questions?

# Performance

| type | double | | float | |
|---|---|---|---|---|
| compilation option | O0 | O3 | O0 | O3 |
| tool_none | x6.3 | x6.4 | x7.0 | x8.4 |
| nearest | x10.0 | x22.3 | x10.6 | x27.9 |
| nearest-nc | x10.1 | x21.8 | x10.4 | x27.2 |
| random | x15.4 | x41.2 | x17.7 | x54.1 |
| nearness | x17.4 | x47.5 | x21.0 | x66.0 |
| random_det | x16.6 | x46.0 | x19.8 | x64.7 |
| random_comdet | x16.8 | x47.2 | x20.2 | x66.8 |
| random_scomdet | x19.4 | x56.2 | x23.4 | x79.8 |
| nearness_det | x19.2 | x53.1 | x23.5 | x77.5 |
| nearness_comdet | x20.2 | x56.5 | x24.9 | x85.3 |
| nearness_scomdet | x20.0 | x56.7 | x25.3 | x85.4 |
| sr_monotonic | x20.4 | x57.4 | x25.0 | x81.5 |
| sr_smonotonic | x20.5 | x57.0 | x25.3 | x83.1 |

# Number of samples

| Confidence level $1-\alpha$ | Probability $p$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 0.66 | 0.75 | 0.8 | 0.85 | 0.9 | 0.95 | 0.99 | 0.995 | 0.999 |
| 0.66 | 3 | 4 | 5 | 7 | 11 | 22 | 108 | 216 | 1079 |
| 0.75 | 4 | 5 | 7 | 9 | 14 | 28 | 138 | 277 | 1386 |
| 0.8 | 4 | 6 | 8 | 10 | 16 | 32 | 161 | 322 | 1609 |
| 0.85 | 5 | 7 | 9 | 12 | 19 | 37 | 189 | 379 | 1897 |
| 0.9 | 6 | 9 | 11 | 15 | 22 | 45 | 230 | 460 | 2302 |
| 0.95 | 8 | 11 | 14 | 19 | 29 | 59 | 299 | 598 | 2995 |
| 0.99 | 12 | 17 | 21 | 29 | 44 | 90 | 459 | 919 | 4603 |
| 0.995 | 13 | 19 | 24 | 33 | 51 | 104 | 528 | 1058 | 5296 |
| 0.999 | 17 | 25 | 31 | 43 | 66 | 135 | 688 | 1379 | 6905 |

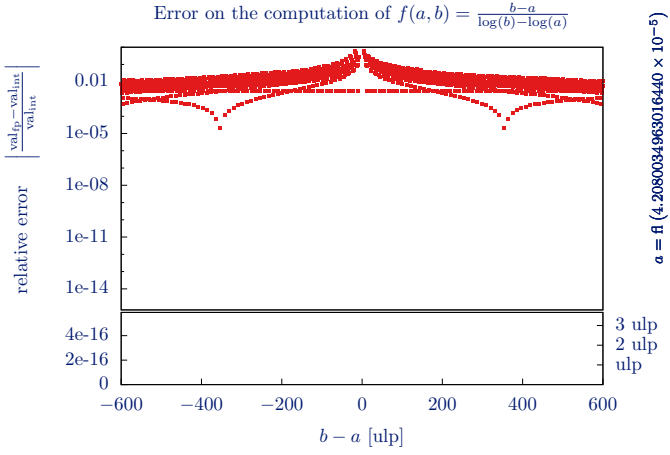
**Confidence Intervals for Stochastic Arithmetic**, Devan Sohier, Pablo De Oliveira Castro, François Févotte, Bruno Lathuilière, Eric Petit, Olivier Jamond

# Bug fix example (1/3)

$$f(a,b) = \begin{cases} a & \text{if } a = b \\ \frac{b-a}{\log(b)-\log(a)} & \text{if not} \end{cases}$$

## Empirical study

- outside the code
- around the problematic
- reference = interval arithmetic



# Bug fix example (1/3)

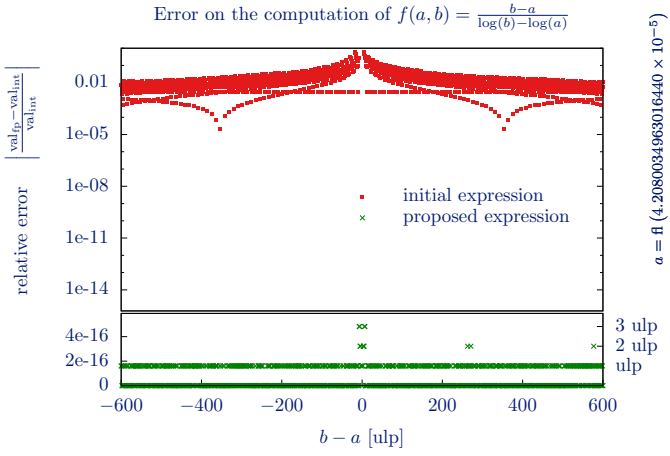
$$f(a,b) = \begin{cases} a & \text{if } a = b \\ \frac{b-a}{\log(b)-\log(a)} & \text{if not} \end{cases}$$

rewritting
manual

$$f(a,b) = \begin{cases} a & \text{if } a = b \\ a\,\frac{\frac{b}{a}-1}{\log\left(\frac{b}{a}\right)} & \text{if not} \end{cases}$$

## Empirical study

- outside the code
- around the problematic
- reference = interval arithmetic



# Bug fix example (1/3)

$$f(a,b) = \begin{cases} a & \text{if } a = b \\ \frac{b-a}{\log(b)-\log(a)} & \text{if not} \end{cases}$$

rewritting
manual

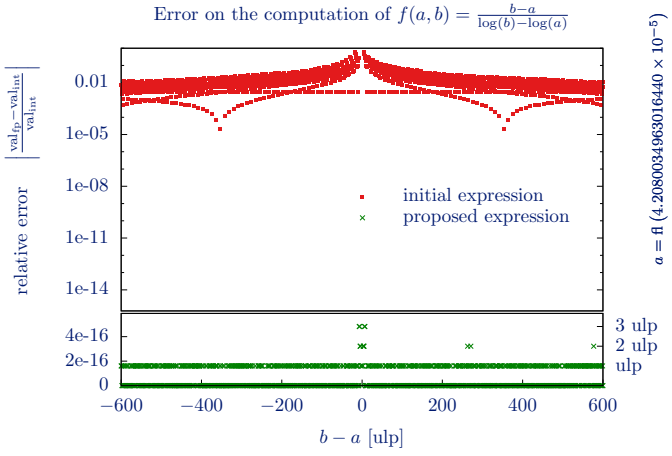
$$f(a,b) = \begin{cases} a & \text{if } a = b \\ a\,\frac{\frac{b}{a}-1}{\log\left(\frac{b}{a}\right)} & \text{if not} \end{cases}$$

## Empirical study

- outside the code
- around the problematic
- reference = interval arithmetic

## Proof

- error bounded by 10 ulps

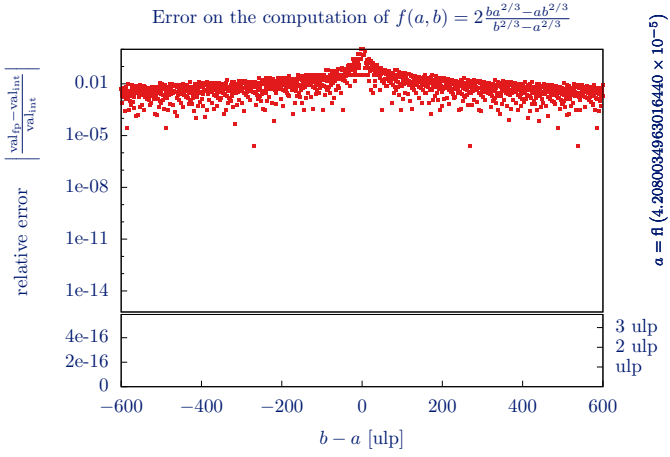


# Bug fix example (2/3)

$$f(a,b) = \begin{cases} a & \text{if } a = b \\ 2\frac{ba^{2/3} - ab^{2/3}}{b^{2/3} - a^{2/3}} & \text{if not} \end{cases}$$

## Empirical study

- outside the code
- around the problematic
- reference = interval arithmetic



# Bug fix example (2/3)

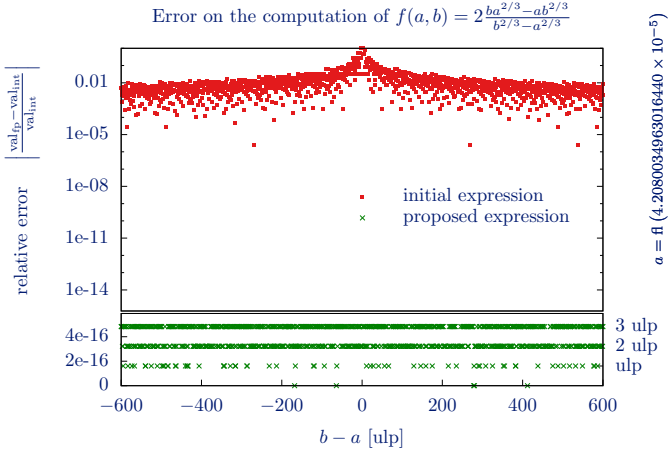
$$f(a,b) = \begin{cases} a & \text{if } a = b \\ 2\frac{ba^{2/3} - ab^{2/3}}{b^{2/3} - a^{2/3}} & \text{if not} \end{cases}$$

wolfram alpha

$$f(a,b) = 2\,\frac{a^{2/3}b^{2/3}}{a^{1/3} + b^{1/3}}$$

## Empirical study

- outside the code
- around the problematic
- reference = interval arithmetic



# Bug fix example (2/3)

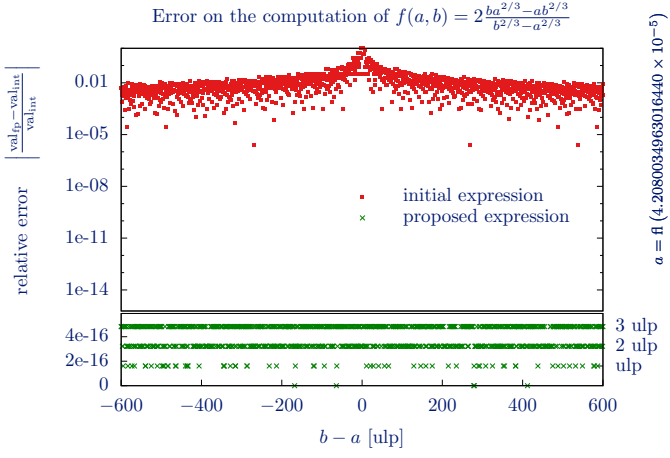
$$f(a,b) = \begin{cases} a & \text{if } a = b \\ 2\frac{ba^{2/3} - ab^{2/3}}{b^{2/3} - a^{2/3}} & \text{if not} \end{cases}$$

wolfram alpha

$$f(a,b) = 2\,\frac{a^{2/3}b^{2/3}}{a^{1/3} + b^{1/3}}$$

## Empirical study

- outside the code
- around the problematic
- reference = interval arithmetic



# Bug fix example (3/3)

$$f_n(a,b) = \begin{cases} a & \text{if } a = b \\ (n-1)\dfrac{b^{\frac{1}{n}} - a^{\frac{1}{n}}}{a^{\frac{1}{n}-1} - b^{\frac{1}{n}-1}} & \text{if not} \end{cases} \xrightarrow[\text{rewriting}]{\text{manual}} f_n(a,b) = \frac{n-1}{\sum_{i=1}^{n-1} a^{\frac{i-n}{n}} b^{\frac{-i}{n}}}$$