Hes·so // FRIBOURG FREIBURG
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz

iCoSys
Institute of Complex Systems

COJAC: Climbing over Java Arithmetic Capabilities

Smart tools to boost the power of numbers in your existing Java code

Realization

HES-SO Fribourg

Prof. Frédéric Bapst
frederic.bapst@hefr.ch

Prof. Richard Baltensperger

+many HEIA-FR students
(Lucy Linder, Baptiste Wicht...)

Keywords

- Diagnostic tool
- Integer overflow
- Numerical analysis
- Code instrumentation
- Software defects
- High precision numbers
- Automatic differentiation
- Symbolic computing
- Chebfun

Our skills

- Java bytecode on-the-fly instrumentation
- Numerical instability detection
- Overflow detection
- Automatic differentiation
- (Semi-)Symbolic methods

Valorization

Free tool to analyze and enrich the arithmetic behavior of Java programs, bridging the gap between numerical analysts and ordinary programmers.

Partnership

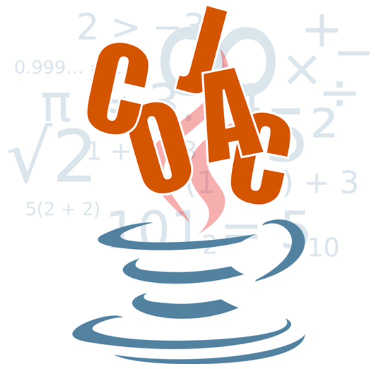
-

Funding

-

Schedule

11/2015 – 11/2035 ☺



github.com/Cojac

Programmers tend to underestimate the possible anomalies that are creeping wherever there are numbers and computations. Numbers in programming do not behave like numbers in mathematics. Every programmer knows that, but nearly all are writing code that contains related defects, because it can be truly subtle.

The risk of letting the poison spread into the code is higher than expected. The consequences range from an incoherent program outcome to critical security breaches.

Many powerful number models are available as alternatives to float/double, but using them generally means strong programming efforts. Why not program with the standard types, and simply switch to a better model at runtime?

For Java developers, we have designed, implemented and published a complete tool that provides several levels of assistance, to diagnose how reasonably the numbers and computations behave, as well as to boost their power and bring dramatically new capabilities. The tool is free and open-source.

COJAC principles:

- no source code modification (!!), no recompilation
- write code with standard numbers, boost them at runtime
- just add a flag when you launch your existing code
- numbers have much to tell you, and they can be powerful

COJAC main features:

- detection of every integer overflow
- signaling floating point anomalies (NaN or ∞, cancellation, absorption, underflow, risky comparison, unnatural casting…)
- interval computation
- arbitrary precision numbers
- automatic differentiation
- symbolic expressions simplification/transformation
- symbolic functions and Chebfun-like functions
- location-dependent behavior and delta-debugging
- double-as-float debugging trick + delta-debugging
- changing the rounding mode

https://github.com/Cojac