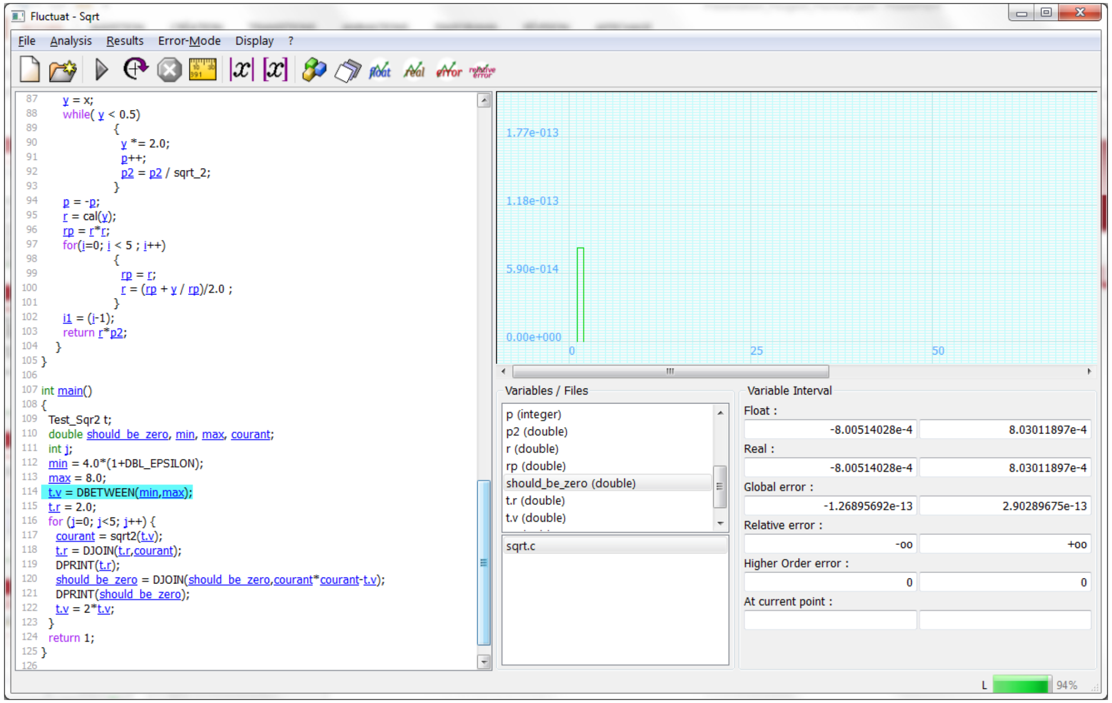
CODE ANALYSES FOR NUMERICAL ACCURACY WITH AFFINE FORMS: FROM DIAGNOSIS TO THE ORIGIN OF THE NUMERICAL ERRORS

Teratec 2017 Forum | Védrine Franck

NUMERICAL CODE ACCURACY WITH FLUCTUAT

- **Compare floating point with ideal computation**
- **Use interval [a, b] and affine forms**
  - **Affine forms** → relationships between variables + error origin
  - For the real domain the floating-point domain and the absolute error



- **Abstract Interpretation based analysis**
  - If Fluctuat provides bounds, then ∀ execution verifying the hypotheses, the results are guaranteed to be in the bounds
  - Fluctuat generates approximations: analysis time / precision of the analysis

- E. Goubault, S. Putot, M. Martel, K. Tekkal, F. Védrine, O. Bouissou, T. Le Gall

FLUCTUAT CASE STUDIES

- **Embedded critical numeric components**
  - 50 to 500 lines of code
  - Provides a bound for the error of output values for the whole input ranges
  - Ex: linear filters, polynomial interpolation and interpolation tables

- **Synchronous systems**
  - 500 to 30 000 lines of code
  - Thin numerical scenarios fin around a test case

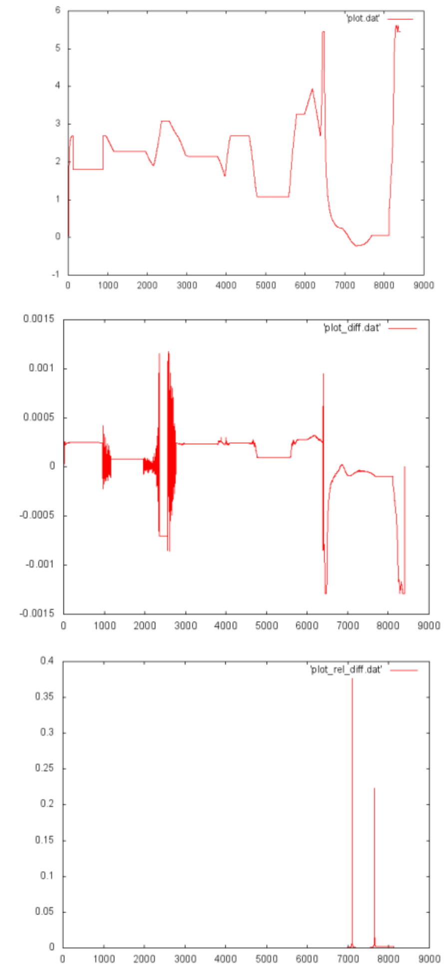
  - Detection of a potential numerical instability around the test case
    - **Expression** with a strong error
      $0 \leq (1 - \cos(x))/x^2 < 1/2$ for values of x close to 0
    - **Progressive accumulation** of errors
      Σ 0.1
    - **Unstable branches**
      if $(x \geq 0)$ **then** $z \leftarrow +1.0$ **else** $z \leftarrow -1.0$
    - **Model error** if the specification is connected with the code

ANALYSIS TIME AND PRECISION

- **Industrial code**
  - Synchronous system of 30 000 lines of code
  - Filter the input sensors, reaction according to a physical model, many parameters
  - No solving libraries like LAPACK

  - Thin numerical scenario of 8400 cycles that extends a test case

- **Results of the analysis for an output variable**
  - Majority of cycles $\Rightarrow$ error $\leq 4\times10^{-2}$
    Proof with a pessimistic accumulation of ½ ulp
    = **developer reasoning** (ulp = unit in the last place)
  - To compare with **double** and **long double** instrumentation on the test case $\Rightarrow$ error $\leq 2\times10^{-3}$
    observation on a sum of rounding errors
  - Analysis time (memory model, number of relations):
    1h by cycle $\Rightarrow$ 100 cycles / 8000 cycles



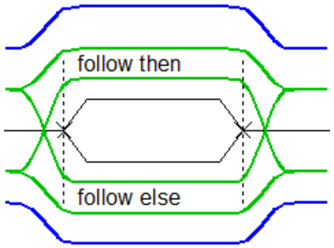
SPECIALIZED VERSION OF FLUCTUAT FOR THIN SCENARIOS

- **Instrumentation library « float_diagnosis » based on affine forms**
- **Can explore all the execution paths of the scenario**

  if $(x \geq 0)$ **then** $z \leftarrow +1.0$ **else** $z \leftarrow -1.0$

  $x \in [-10^{-4}, +10^{-4}]$ with an error $\in [-10^{-8}, +10^{-8}]$

  $\Rightarrow$ 6 execution paths to consider



- **Instrumentation by operator overloading +, -, *, / and redefinition of the types float, double (like CADNA) and recompilation**

- **Differences between instrumentation and Abstract Interpretation (Fluctuat)**
  - Instrumentation: path exploration $\leftrightarrow$ Abstract Interpretation = fixpoint analysis
  - Operations from continuous world (float) $\rightarrow$ to discrete world (int, pointer)
    - Abstract interpretation: interval of int, pointers
    - Instrumentation: enumeration of int + manages unstable branches

ANALYSIS TIME AND PRECISION WITH INSTRUMENTATION

- **Activation of the analysis on the unstable branches**
  - 40 to 60 unstable branches by simulation cycle
    - to compare with 1 unstable branch every 100 cycles = mode comparing **double** and **long double**.
  - The majority of unstable branches $\Rightarrow$ no discontinuity
  - Some false alarms: require better synchronization between float and real

- **Analysis results**
  - Some cycles prove : error $\leq 4\times10^{-2}$
  - Except unstable branches, a majority of cycles $\Rightarrow$ error $\leq 4\times10^{-2}$
  - Analysis time : 1s by cycle

- **Comparison with different instrumentations (exact, interval)**
  - Compare **double** and **long double** : 0.5 ms/cycle = not very stable results
  - Compare with reals in [min, max] and simulated floats
    = too imprecise results: 10ms/cycle

SCIENTIFIC CODE ANALYSIS ?

- **Many challenges for the affine forms**
  - Several millions of lines of code, parallelism
  - May contain finite element libraries
    - dynamically built meshes
  - May contain solving libraries like LAPACK
  - Simulation of several days
  - Strong dependencies to the initial data
  - Code soon adjusted on observed numerical errors:
    observed error ≠ sound accumulation of ½ ulps (developer's reasoning)
    sound results may be prohibitive

- **Analyze only the behavior of kernel code on thin scenarios around a simulation**

EXPECTED RESULTS

- **Try to « catch » numerical instabilities like**
  - discontinuous unstable branches
  - big loss of accuracy in an expression
  - big accumulation of errors
    + chain of instructions involved in the final error
- **If presence of numerical instabilities**
  - provide the means to understand them
- **If absence of numerical instabilities**
  - translate the scenario into a non-regression test

- **Research activity to analyze such code**
  - Automatic placement of synchronization points (unstable branches) – static analysis with Frama-C
  - Limit the size of affine forms but keep the critical relations between the domains and the errors
  - Go beyond affine forms – precision of the analysis

CONCLUSION: OBJECTIVES OF FORMAL METHODS

- **Express an accuracy formula whatever is the execution**

- **Several steps:**
- **« Architecture » of the accuracy formula**
  - Definition of the relationships between the errors and the domains
- **« Adjust » the accuracy formula**
  - Numerical coefficients of the formula obtained by scenario-based analyses
  - Mix of relative accuracy, absolute accuracy
- **« Prove » the accuracy formula**
  - With logical / formal reasoning

- **To formally compare some key algorithms and to go towards a better control of the computed results**

Thanks for your attention

With the support

Fluctuat

float_diagnosis library

Commissariat à l'énergie atomique et aux énergies alternatives
Institut List | CEA SACLAY NANO-INNOV | BAT. 861 – PC142
91191 Gif-sur-Yvette Cedex - FRANCE
www-list.cea.fr

Établissement public à caractère industriel et commercial | RCS Paris B 775 685 019