Numerical Verification and Reproducibility of Digital Simulations

Floating Point Adverse Effects on the Numerical Verification and Reproducibility of Digital Simulations
Seminar at The Maison de la Simulation

Christophe DENIS

Research Fellow (HDR), CMLA, ENS Paris-Saclay
also Research Engineer at EDF R&D
christophe.denis@cmla.ens-cachan.fr

"The scientific community should always be able to verify that a published program will produce correct results, or that a published calculation is correct, in the same way that it can check the truth of scientific theorems or experimental measurements", K. Roberts. The publication of scientific fortran programs. Comput. Phys. Comm., 1, 1969

13 mars 2017
Outline of the talk

1. Introduction
2. Recovering numerical reproducibility on numerical codes
3. Estimation of the numerical accuracy by using Monte Carlo Arithmetic (MCA)
4. Verificarlo: a toolbox for automatic and transparent MCA computations
5. VerifiShadow: a new prototype tool using the Intel Pin tool
6. On-going academic work: Reducing data storage in weather forecasts
7. On-going industrial work: Checking the FP accuracy of non regression tests with Verificarlo
8. Concluding remarks and future works
Acknowledgements

- Verificarlo is developed under a close scientific collaboration with Eric Petit and Pablo de Oliveira Castro from Université de Versailles Saint-Quentin-en-Yvelines
- The work on Verificarlo has been partially supported by a public grant as part of Investissement d’avenir, référence ANR-11-LABX-0056-LMH, LabEx LMH
- Philippe Langlois and Raffifie Nheili (UPVD, numerical reproducibility of the Telemac-2D software)
- Lin Guo (CMLA)
- Olivier Jamond (CEA, on EPX)
- Marc Andreewsky, Thomas De Soza and Josselin Delmas (EDF R&D)
- Fenwick Cooper, Peter Duben, Andrew Dawson (University of Oxford)
My scientific topics

1. recover the numerical reproducibility for debugging purposes
2. analyze the effect of the floating point arithmetic on the precision of the results
3. take into account all sources of uncertainties
4. manage the compromise between the performance, the precision and the electrical consumption of large scale simulations
Today, presentation of topic 1 and 2:

1. recover the numerical reproducibility for debugging purposes
2. analyze the effect of the floating point arithmetic on the precision of the results
3. take into account all sources of uncertainties
4. manage the compromise between the performance, the precision and the electrical consumption of large scale simulations
Numerical Verification and Reproducibility of Digital Simulations

Introduction

When FP meets parallel computing

Numerical simulation of a code using the same version of numerical libraries

Same input data:

Simulation 1

Simulation 2

These two simulations could be not exactly the same if using different:

- number of processors
- compilers (eg. gcc or icc)
- compiler option (eg. -O3 –fast-math or -O0)
- programming languages (eg Fortran 90 or C++)
- computing resources (eg. CPU or CPU+GPU)
- version of numerical libraries (eg. version 1.2 or 1.2.6)
Illustration: from 1 to ... 2 processors

- Performed by the software Telemac-2D to simulate free-surface flows [Hervouet, 2007]
- Comparison between the serial and the parallel simulation (2 processors) on $H$ [Langlois et al., 2016]
  - Blank finite elements contained non reproducible results

If no bug in the CFD code, what is the hidden source of approximation?
Floating point computation: the IEEE-754 standard

- **Real number**: value that represents a quantity along an infinite continuous line.
- In the computer, floating point (FP) numbers approximate real numbers: discrete and finite set of numbers.
  - Prof. William Kahan (UCLA), the primary architect behind the IEEE 754-1985 standard for floating-point computation.
  - Received the Turing Award in 1989 for "his fundamental contributions to numerical analysis".
- **Representation and encoding in memory**:
  - Single precision (32 bits)
  - Double precision (64 bits)
- **Four rounding modes**
  - Rounding to nearest, rounding toward $+\infty$, rounding toward $-\infty$, rounding toward zero
Floating point computation: some adverse effects

- Even if the IEEE-754 provides the best trade-off between range and precision, there are some cons:
  - Representation error of real numbers, for example the decimal value 0.1 is not exactly representable (see the Patriot Missile Software Problem)
    - $t = t + 0.1$ (accumulation des erreurs) unlike $t = i \times 0.1$, $i = 1..., n$
  - Loss of arithmetical properties (for example the floating point summation is not associative)
    - In single precision, $10^{32} - 10^{32} + 0.01 = 0.01 \neq 10^{32} + 0.01 - 10^{32} = 0.0$
  - A floating point computation $fl(a \circ b)$ is a model of an exact computing $a \circ b$
    - $fl(a \circ b) = (a \circ b)(1 + \epsilon)$ (1)
  - Cancellation error
    - The FP result is zero while the exact value is not
How to improve the numerical reproducibility of hydrodynamics simulations

- PhD Student: R. Nheili, Supervisors P. Langlois (UPVD) and C. Denis, in collaboration with EDF R&D
- Aim of the PhD-Thesis
  1. identify the sources of the non-reproducibility in Telemac
  2. propose, implement and analyze efficient methods that eliminate this drawback.

One source of non-reproducibility: the finite element assembly

The whole domain

\[ V(i) = \hat{a} \]

The domain decomposition into two sub-domains

\[ V^{d_1}(i) = \hat{b} \quad V^{d_2}(i) = \hat{c} \]

Interface point assembly

\[ V(i) = \hat{b} + \hat{c} \neq \hat{a} \]

source: PhD-Thesis of R. Nheili
Recovering numerical reproducibility on numerical codes

Compensated summation has been the most efficient method to recover numerical reproducibility

- Key method: Error free transformation

\[
a \ op_{\text{exact}} \ b = x \ op \ y, \ \text{avec} \ a, b \in \mathbb{R} \ et \ x, y \in \mathbb{F}
\]

- Algorithm 2sum (Knuth, 1965)

\[
[x, y] = \text{fonction } 2\text{Sum}(a, b)
\]

\[
x = a + b
\]
\[
z = x - a
\]
\[
y = (a - (x - z)) + (b - z)
\]
Results obtained in the improvement of the numerical reproducibility of hydrodynamics simulations

- **acc** is the difference between the original code and the version using compensated summation (not the numerical accuracy ...)
- Manual intrusive works in the code
- Development at CMLA of a tool for automatically improving the numerical reproducibility of a code
But numerical precision $\neq$ numerical reproducibility

- High precision and high reproducibility
- Low precision and high reproducibility
Analysis of the numerical precision

Exascale simulations will be able to produce results that one will not be able to validate the computations with experimental measurements ...

- Analysis of the numerical precision of the FP results is required to:
  - perform a rigorous $V\&V$ procedure;
  - estimate the digits of a computation not affected by the round-off error propagation;
  - find the best compromise between performance and accuracy.

- Objectives of our work:
  - democratize the assessment of the numerical accuracy of full-scale real-life applications without the assistance of an expert
  - automatically use in a transparent way for the end-user.
  - provide automatically statistical results for non-experts to understand and to interpret the output of the analysis.
An example provided by W. Kahan [Kahan, 1966]

Example from W. Kahan (UCLA)

\[
\begin{pmatrix}
0.2161 & 0.1441 \\
1.2969 & 0.8648
\end{pmatrix}
\begin{pmatrix}
x \\
x
\end{pmatrix}
=
\begin{pmatrix}
0.1440 \\
0.8642
\end{pmatrix},
\begin{pmatrix}
x_{\text{exact}}
\end{pmatrix}
=
\begin{pmatrix}
2 \\
-2
\end{pmatrix}
\tag{2}
\]

Results using LAPACK routines using IEEE-754 single and double precision FP numbers:

\[
\begin{pmatrix}
x_{\text{single}}
\end{pmatrix}
=
\begin{pmatrix}
1.33317912 \\
-1.00000000
\end{pmatrix},
\begin{pmatrix}
x_{\text{double}}
\end{pmatrix}
=
\begin{pmatrix}
2.00000000240030218 \\
-2.00000000359962060
\end{pmatrix}
\]

\(s(x)\) : number of common digit between the exact and the FP computation \(x\)

- \(s(x_{\text{single}}(1)) = 0, s(x_{\text{double}}(1)) = 9\)
- \(s(x_{\text{single}}(2)) = 0, s(x_{\text{double}}(2)) = 9\)

How to estimate \(s\) if the value of \(x_{\text{exact}}\) is not known on a whole scientific code?

- without modifying its source code (to capture the effect of the compiler optimisations)
- without before requiring the correction of some numerical instabilities
Estimation of the numerical accuracy by using the Monte Carlo Arithmetic (MCA) [Parker, 2011]

- Stochastically simulate rounding and catastrophic cancellation errors
- Introduce a uniformly-distributed error at a virtual precision $t$

\[
inexact(x, t, \xi) = x + 2^{e_x-t} \xi = (-1)^{s_x} (m_x + 2^{-t} \xi) 2^{e_x} \quad (3)\]

- $e_x$ exponent of $x$, $m_x$ significand of $x$, $s_x$ sign of $x$
- $\xi$ uniform random variable in $[-\frac{1}{2}, \frac{1}{2}]$

Each floating point operation is transformed in a MCA operation:

\[
x \circ y \rightarrow round(inexact(inexact(x) \circ inexact(y)))
\]

- Distribution of the round-off errors is estimated using $N$ Monte Carlo sampling $x$
- $\hat{s}(x)$ : estimation of $s$ computed as follows :

\[
\hat{s}(x) = -\log_{10} \frac{\hat{\sigma}(x)}{\hat{\mu}(x)}
\]
Verificarlo: a toolbox for automatic and transparent MCA computations [Denis et al., 2016] [Denis, 2016]

- LLVM-based transparent MCA instrumentation
  - Support MCA analysis of large code-bases (eg. LAPACK) without any source code modification
  - Multi-language support, per function analysis
  - Instrumentation occurs just before code generation
  - Enables analyzing precision loss due to the LLVM compiler optimizations unlike tools instrumenting the source code
Numerical Verification and Reproducibility of Digital Simulations

Example provided by W. Kahan

- Demo
- Recall: computations using IEEE-754 FP numbers

<table>
<thead>
<tr>
<th>Precision</th>
<th>Result</th>
<th>$s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP</td>
<td>$x(1) = 1.33317912$</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>$x(2) = -1.00000000$</td>
<td>0</td>
</tr>
<tr>
<td>DP</td>
<td>$x(1) = 2.000000000240030218$</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>$x(2) = 2.000000000359962060$</td>
<td>9</td>
</tr>
</tbody>
</table>

Computation performed with Verificarlo

- Automatically implementation of MCA on the whole LAPACK and BLAS libraries without any modification of their source codes.

<table>
<thead>
<tr>
<th>Precision</th>
<th>$\hat{\mu}$</th>
<th>$\hat{\sigma}$</th>
<th>$\hat{s}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP</td>
<td>$\hat{\mu}(x(1)) = 1.02463705$</td>
<td>$\hat{\sigma}(x(1)) = 6.4...$</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>$\hat{\mu}(x(2)) = 6.46717332$</td>
<td>$\hat{\sigma}(x(2)) = 9.6...$</td>
<td>0.0</td>
</tr>
<tr>
<td>DP</td>
<td>$\hat{\mu}(x(1)) = 1.9999999992$</td>
<td>$\hat{\sigma}(x(1)) = 8.4... \times 10^{-9}$</td>
<td>8.3</td>
</tr>
<tr>
<td></td>
<td>$\hat{\mu}(x(2)) = -1.9999999988$</td>
<td>$\hat{\sigma}(x(2)) = 1.2... \times 10^{-8}$</td>
<td>8.2</td>
</tr>
</tbody>
</table>
On the number of MCA samples

**Figure** – Convergence of the $s$ value using double FP numbers: (a) on the left, 200 samples, (b) on the right focus on the first 9 samples

- A small number of samples (<5) for this case gives no reliable estimation
- Using confidence interval to find a good compromise on the number of samples
Confidence interval on $s$

- Normality test (Shapiro–Wilk test) OK from 8 MCA samples
- Confidence test on $s$ built from the work of Miller [Miller, 1991] on the relative standard deviation

**Figure** – Confidence interval at 99% by using the Miller method among MCA samples

- The Sharma- Krishna [Sharma and Krishna, 1994] method is used when the Shapiro–Wilk test fails
VerifiShadow : a new prototype tool using the Intel Pin tool

- Aim of the VerifiShadow software developed at CMLA in the context of a scientific collaboration:
  - Implement verification methods (MCA, Interval Arithmetic,...) directly in the executable as it is compiled (for example in production)
  - Compare the statistical results obtained on Verificarlo and VerifiShadow
  - Impact of the compiler and the numerical libraries on the numerical precision
  - Pinpoint the exact operation that is to blame for a precision loss

- First version of VerifiShadow: Implementation of MCA for double precision numbers
Proof of concept of VerifiShadow on the example provided by W. Kahan

**Figure** – Convergence of the $s$ value using double FP numbers: (a) on the left, with Verificarlo, (b) on right, with VerifiShadow

- **Future work**
  - Implementation of MCA for single precision and further proof of concepts
  - Validation of this software on a whole physics atomis code
On-going work: Reducing data storage in weather forecasts.

- Scientific collaboration with the Department of Physics of the University of Oxford
- Lorenz Chaotic models used in weather forecasts
- Objective: Determine the best precision of FP numbers to trade data storage and accuracy (e.g., FPGA)
- Verificarlo is used to compute the accuracy of the results for different virtual precision of the FP numbers
On-going work: Reducing data storage in weather forecasts (cont'd)

- Comparison for 3 reduced precisions \( \text{prec} = \{15, 20, 25\} \) between the value of the forecast error \( fe \) and the number of significant bits given by Verificarlo for the Lorenz'95 model.

![Graph showing forecast error (fe) and number of significant bits (s) vs time in model time units.](image)

- Future work: test this approach with more sophisticated models (two-level Lorenz model, shallow water model, spectral dynamical core.)
On-going industrial work: Checking the FP accuracy of non regression tests with Verificarlo

- Context: on-going scientific collaboration between EDF R&D and CMLA
  - Originally developed by EDF and written in Fortran, C, Python
  - 1,500,000 lines of source code and 2,000 of tests
- Objective of this scientific collaboration
  - Analysis of the effect of the FP arithmetic of non reproducible test case
    - E.g., if the numerical dispersion estimated by Verificarlo << variability of the results ⇒ strong suspicion of a bug
  - Challenge the use of Verificarlo on a full real application
  - MCA has been implemented automatically in Code_Aster without modifying a line of source code!
Verificarlo is used to estimate the number of significant digits of \( f_{\text{min}} \) in the band \( B = [1.0\, \text{Hz}, 200.0\, \text{Hz}] \) (70 samples)

For 69 samples, the number of frequencies computed in the band \( B \) is 5, sorted as:
- 25.392.. Hz / 52.856.. Hz / 69.607.. Hz / 137.40.. Hz / 153.21.. Hz

For one single sample, the number of frequencies computed in the band \( B \) is 6, sorted as:
- 6.990.. Hz / 25.392.. Hz / 52.856.. Hz / 69.607.. Hz / 137.40.. Hz / 153.21.. Hz
On-going industrial work: Checking the FP accuracy of non regression tests with Verificarlo

**Code_Aster, test case 1**

- The number of frequency is sensitive to the floating point arithmetic (3 or 4 samples would be not sufficient to underline that)
- Estimation of the numerical precision of $f_{min}$ (by taking the frequency number 2 for the sample having 6 frequencies)
This test case showed significant variability on the displacement field on several computers.

The software development team eliminated this problem by correcting an unstable test.

Number of significant digits before and after the correction provided by Verificarlo

- Before the correction ($\sim 75\%$ of failure)
- After the correction (0% of failure)

Verificarlo has assessed the correction of the unstable test.
On-going industrial work: Checking the numerical reproducibility of Europlexus software

- Context: on-going scientific collaboration between CEA, CMLA and UVSQ
- Application: the software Europlexus
  - fast transient dynamic simulation software
  - co-developed by CEA (CEA), the Joint Research Center (JRC) of the European Commission, and other industrial and academic partners
    - about 1 million lines of Fortran 77 and Fortran 90.
- Automatic implementation of MCA with Verificarlo during the compilation of Europlexus
Numerical Verification and Reproducibility of Digital Simulations

On-going industrial work: Checking the FP accuracy of non regression tests with Verificarlo

Results obtained by O. Jamond, P. de Oliveira Castro and E. Petit

- Europlexus buckling simulation of two doubly clamped column
  - a Normal column
  - a Fragilized column for which the element labeled in the figure has been weakened to create a breaking point.

- the buckling direction for the Normal column is completely dominated by the small numerical error introduced.
The assessment of the numerical accuracy of scientific codes becomes crucial

- to explain the non numerical reproducibility in HPC
- to control the floating point computation when porting a scientific code on another programming language or on different computing resources
- to find the best compromise between the performance and the precision

The current versions of Verificarlo and Verifishadow are the first step toward fully automatic tools without the assistance of an expert
Future work

- Development of a tool built on Verificarlo measuring the numerical efficiency of a parallel architecture (in collaboration with IDRIS, R. Carpentier)

- Development of a user-friendly environment around Verificarlo (easy to install and use) : support from the CNRS DIRE.

- Epistemic uncertainties around round-off error models

- First complete version of Verifishadow

- Statistical post-treatment toolbox to go beyond the standard deviation analysis

- Numerical verification of full-scale real-life applications (Applications and collaborations are welcome!)

- Management of the compromise between performance and precision
Some Informations to go further

  - In the program, there is a lecture on MCA and a lab session on Verificarlo. Participants could come with their applications.

- (in french) Journées national du développement logiciel (JDEV 2017 Marseille, 4 au 7 juillet 2017)
  - Dans le programme : Containers : prise en compte des effets de l’arithmétique flottante sur leurs reproductibilités et précisions numériques

THANK YOU FOR YOUR ATTENTION
References I


References III


Principle of compensated summation algorithms

- First step

- Second step

- Last step
Principle of compensated summation algorithms

- The result:

\[ S = \sum_{i=1}^{n-1} e_i + s_n \]

At the end of this process, two possibilities

1. Restart the process on this array
2. or compute the compensated summation \( S \):

Figure extraite de : J-M Chesneau et al., Round-off errors, Round-off errors Encyclopedia of Computer Science and Engineering, volume 4, Wiley, 2009
An increasing computing power is required ...

For helping both the scientist to solve problems and companies to help their decision making on their numerous financial, technological and concurrential challenges. This computing power is used to:

- Decrease the gap between the result of a numerical simulation and the studied physical phenomenon
- Study accurately local physical phenomenon
- Take into account all the uncertainties
- Set up more efficient computing schemes (model reduction, deep learning, metamodel etc.)
- To compensate the loss of efficiency of numerical codes (the majority of the codes indeed use a low part of the allocated resources)
An increasing continuous power computing has been provided by alternatively these two following steps:

1. Increase of the performance of the processors (step “tick” from the Intel)
2. Add in the architecture more processors - and accelerators - having the same performance as the previous step “tick” (step “tick” from the Intel)

The past decades, a relatively cheap power computing was provided mainly thanks to the constant reduction of the fine engraving (Free Lunch)
Free lunch is over

- Technical difficulty to decrease the fine engraving the processor, thermal dissipation
- "Tick-tick" cycle time increases, stagnation of the processor speed.

⇒ The parallel architecture has to be more complex and heterogeneous

« Although we do not know the detail of the hardware that will be available, we can be certain that the level of parallelism will increase significantly, that machines will be more complex and heterogeneous [...]. Thus, in common with our colleagues in the US and Japan, we recognize that considerably more effort and manpower will be required to even begin to address this complexity ... . »

Some scientific challenges to be solved

1. Limitation of the electrical consumption
2. Fault tolerance
3. Performance prediction and optimisation
4. Uncertainties quantification
5. Numerical reproducibility and verification
Numerical Verification and Reproducibility of Digital Simulations

Proof of concept on the industrial code TELEMAC

- Contains several codes TELEMAC-2D to simulate free-surface flows Modelisation of the free surface flows [Hervouet, 2007]
  - Developed by EDF R&D LHNE through a scientific consortium since 1987
  - Open source distribution
  - More than 300k lines of source code
  - Telemac-2D
    - Simulate 2D free-surface flows
    - Shallow water equations, finite element modelisation
    - Sequential or parallel version (based on a domain decomposition)
Proof of concept on the industrial code TELEMAC (cont’d)

- Automatically implementation of MCA without any modification of their source codes
- Overhead of a MCA computation ($16 \times$)
  - Estimation of the extracted power of a tidal turbine farms (in collaboration with Marc Andreewsky, research-Engineer at EDF R&D)
  - from the simulation of the computation of the flows velocities