



HAL
open science

Repeatability, Reproducibility, Computer Science and High Performance Computing: Stochastic simulations can be reproducible too...

David R.C. Hill

► **To cite this version:**

David R.C. Hill. Repeatability, Reproducibility, Computer Science and High Performance Computing: Stochastic simulations can be reproducible too.... 2019 International Conference on High Performance Computing & Simulation (HPCS), Jul 2019, Dublin, Ireland. pp.322-323, 10.1109/HPCS48598.2019.9188157 . hal-04060904

HAL Id: hal-04060904

<https://uca.hal.science/hal-04060904v1>

Submitted on 6 Apr 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Repeatability, Reproducibility, Computer Science and High Performance Computing

Stochastic simulations can be reproducible too...

David R.C. HILL

Université Clermont Auvergne, CNRS, Clermont Auvergne INP, Mines

Saint-Etienne, LIMOS, 63000 Clermont-Ferrand, France

david.hill@uca.fr

INVITED SPEECH

IEEE International Conference on High Performance Computing & Simulation

HPCS 2019, 15-19 July 2019, Dublin,

Many of us know the important work of Karl Popper, as philosopher of Science, specialized in modeling and simulation. Karl Popper is generally regarded as one of the greatest philosophers of Science of the 20th century. Among the criteria of the scientific method, which are intangible principles, we find refutability, non-contradiction and reproducibility. The latter is one of the conditions on which Popper distinguishes between the scientific or pseudo-scientific character of a study. Indeed, Science works by drawing "laws" or "principles" from reproducible observations whose main property is to be true as long as no observation has proved otherwise. Scientific conclusions can only be drawn from a well observed and described "result/event", which has appeared several times, and has been observed by different people and/or studies. This criterion eliminates what seriously distorts the results as well as errors in judgment or even manipulations by scientists.

Traditionally we have 2 main branches of the scientific method: the deductive branch with Mathematics and formal logic proofs and the empirical branch with statistical analysis of controlled experiments. We now see claims or hopes that Large Scale Simulation could be a 3rd branch and data intensive or data driven computer Science would be a 4th branch. The problem is that 'our' Computer Science is a young Science that do not currently meet the standards of branches 1 and 2.

Computer Science rely on programs, that humans have to debug. Therefore, what is first required is even more stringent than reproducibility, it is repeatability. This means that with exactly the same input and execution environment, we expect exactly the same results. This is essential at least for debugging. Repeatability with single-core CPUs was relatively straightforward. With high performance computing and the hardware developments of last decades it is becoming more tough, if not impossible to obtain repeatability, often named “bitwise reproducibility” by computer Scientists. In recent computer Science papers we often see the term reproducibility but in fact it means the possibility to repeat the experiment and obtain the same result when the scientific computation is run several times with the same input data and the same execution context [2]. Replicability avoids changes and is required for debugging, reproducibility requires changes in the general sense. Reproducing computer experiments can be achieved by different studies, with different methods and hardware, but they will produce the same scientific conclusion.

The problem with High Performance Computing (HPC) is that we are losing repeatability. It started with parallel systems where we realized that we could not obtain run to run repeatability, even on the same machine [1], and it became harder and harder, if not impossible to reach this repeatability goal with the numerical experiment run on different machines, or with different numbers of processing units, different types of processors/accelerators with different architectures, execution environments, computational loads, etc. Even our processor manufacturers started to introduce the loss of repeatability with non-determinism at the level of CPU cores (with out-of-order execution – see Intel documentation on this point¹). This optimization sometimes impacts the numerical results because floating point operations are non-associative. Most computer scientists are aware of that (not all), but the majority of HPC end-users do not realize this and they expect repeatability in sensitive area. Hybrid computing combining multicore CPUs and compute accelerators like GPUs, many-core processors like Intel Xeon Phi, or even dedicated FPGAs is amplifying this lack of repeatability. The main problem of this fact is that it is becoming particularly tough to setup debugged parallel programs, even at the level of a single computing node which already embed a lot of parallelism. When we have to track and understand a specific event, like particle tracking, the lack of run to run repeatability can be a problem. The quest for repeatability is not limited to debugging.

The trend in numerical simulation is to build more complex models, sometimes with strict accuracy requirements. This often implies the use of high performance computing platforms tuned to obtain the results in an acceptable time (precise climate simulation is an example). For extreme cases in optimization problems, stochastic methods are used to produce interesting approximate solutions much faster than what deterministic

¹ Run-to-Run Reproducibility of Floating-Point Calculations for Applications on Intel® Xeon Phi™ Coprocessors (and Intel® Xeon® Processors) – by Martin Cordel, <https://software.intel.com/en-us/articles/run-to-run-reproducibility-of-floating-point-calculations-for-applications-on-intel-xeon>

algorithms would produce. In the case of complex numerical integration on high dimensional domains, the deterministic methods are often slow and thus the use of Monte Carlo methods is required. In this context, parallel stochastic simulations are too often presented as “non-reproducible” since they have built-in randomness. But in fact, Monte Carlo methods rely on pseudo-random numbers on purpose. When we use pseudo-random numbers for stochastic models, we are running deterministic experiments since pseudo-random number generators have been carefully crafted to be repeatable. This is essential for debugging. In addition, a careful and thorough parallelization of the pseudo-random number generators has to be seriously considered to avoid introducing strange biases in the statistics of your stochastic simulation [3].

However, anyone wishing to produce a scientific work of quality must pay attention to the numerical reproducibility of simulation results [4]. In this talk we will see that significant differences are observed in the results of parallel stochastic simulations if the practitioner fails to apply best practices. Remembering that pseudo-random number generators are deterministic, it is possible to reproduce the same numerical results for parallel stochastic simulations by implementing a rigorous method. A test application running with a billion threads will show that the method enables checking the parallel results with their sequential counterpart before scaling. This possible comparison, at small scale, increases confidence in the proposed models. This talk will present this method in the current context of numerical reproducibility which include soft errors impacting all top 500 machines, including the future Exascale machines that are coming.

Keywords: Repeatability, Reproducibility, High Performance Computing, Stochastic Simulation.

BIOGRAPHY

DAVID R.C. HILL is full Professor of Computer Science at Clermont Auvergne University (Past Blaise Pascal University). He is currently Deputy Director of the Computer Science & Modeling Institute in this University (ISIMA). Professor Hill is doing his research in Modeling and Simulation at the French CNRS (UMR 6158 LIMOS Optimization & Modeling Laboratory). He has authored or co-authored more than 250 papers and he also published several text books. His email and web addresses are respectively david.hill@uca.fr and <https://fc.isima.fr/~hill/>.

REFERENCES

- [1] S. Krishnamurthi and J. Vitek, “The Real Software Crisis: Repeatability as a Core Value. Sharing experiences running artifact evaluation committees for five major conferences.”, Communications Of The Acm, 2015, vol. 58, n°3, pp. 34-36.

- [2] K. Diethelm, “The Limits of Reproducibility in Numerical Simulation”, IEEE Computing in Science and Engineering, January/February, 2012, pp. 64–71.
- [3] D.R.C. Hill, J. Passerat-Palmbach, C. MAZEL and M.K. TRAORE, “Distribution of Random Streams for Simulation Practitioners”, Concurrency and Computation: Practice and Experience, June 2013, vol. 25, n°10, pp. 1427–1442.
- [4] D.R.C. Hill, “Parallel Random Numbers, Simulation and Reproducible Research”, IEEE Computing in Science and Engineering, July/August, 2015, pp. 66–71.