Numerical Reproducibility in open TELEMAC: A Case Study within the Tomawac Library

Rafife Nheili, Philippe Langlois, Christophe Denis

HAL Id: lirmm-01101715

https://hal-lirmm.ccsd.cnrs.fr/lirmm-01101715

Submitted on 9 Jan 2015

Numerical Reproducibility in open TELEMAC: A Case Study within the Tomawac Library

Rafife Nheili[*1], Philippe Langlois[1], and Christophe Denis[2]

[1] Univ. Perpignan Via Domitia, Digits, Architectures et Logiciels Informatiques, F-66860 Perpignan. Univ. Montpellier II, Laboratoire d'Informatique, Robotique et de Microélectronique de Montpellier, UMR 5506, F-34095 Montpellier. CNRS. France.
[2] EDF R&D, F-92141 Clamart. CMLA-ENS Cachan, F-94235 Cachan. France.

Abstract. The lack of numerical reproducibility for parallel floating-point computations is a stringent problem for HPC numerical simulations. Reproducibility is indeed desirable for code verification, testing, debugging and even for certification authority agreement. We describe and analyse strategies towards numerically reproducible simulations with a well known open source software for computational fluid dynamics.

1 Introduction

Numerical reproducibility consists in getting the same result when running a parallel HPC software with a varying number of processing units. The lack of numerical reproducibility mainly comes from the non-associativity of the floating-point addition: the non-deterministic parallel reductions lead to different results, as for instance when summing with the MPI reduction.

We present a case study focusing on open TELEMAC-MASCARET, a well known modelling tool developed by EDF and associates since 20 years [4]. It implements the finite element method to approximate the solutions of numerous applications of free surface fluids in both river and maritime hydraulics [2]. In our scope, the main step of this resolution is the assembly process: i) Quantities are computed for each element and stored in its local nodes. ii) These local quantities are accumulated to the global points of the mesh, via a connectivity table which links local nodes and global points. For the parallel computation, the elements are distributed between the processors thanks to a sub-domain decomposition. So the assembly process is first executed "at the processor level", then the processors that share the interface nodes communicate to exchange and add the missing contributions. This accumulation is order dependant in floating-point arithmetic. As soon as one interface node is shared between more than 2 processors, a different computing order occurs compared to the sequential one. Hence this interface assembly step lacks of numerical reproducibility when varying the number of computing units, *i.e.* the number of sub-domains. Others sources of non-reproducibility as the dot products and the segment assembly step will not be discussed in the following case study.

We focus on the Tomawac library which is the TELEMAC unit to solve transport equations with the method of characteristics. This method introduces no dot product. So its reproducibility only depends on the previously described parallel summation within the finite element assembly step. In this paper, we improve this computation to provide the numerical reproducibility of Tomawac.

---

2   How to recover associativity for parallel FP addition

First we exhibit cases of non-reproducibility for Tomawac when we compute the assembly step with the classical floating-point reduction FPSum. Fig.1(Left) shows the maximum relative errors between the sequential result and the parallel ones for several number of processors $p = 2, 4, 8, 16$ (x-axis is the simulation time scale). Reproducible results should yield a unique plot at every time step when p is varying. It is not the case.

   We analyse three approaches to achieve reproducibility.

IntSum converts FPSum to the sum of 8 byte integers [4]. The finite element and the interface node assembly steps are computed with integer recoding and integer sums. This choice is reasonable here because every interface node belongs to less than ten neighbouring elements. Fig.1(Right) now displays a unique plot that illustrates the expected numerical reproducibility of IntSum. The parallel IntSum accuracy compared to the sequential FPSum one is of the order of 10^{-15}. Reproducibility and accuracy are different software quality goals. Indeed IntSum is more accurate than FPSum.

CompSum improves FPSum with a twice more accurate compensated sum [3]. It consists in accumulating the round-off errors of intermediate sums to later compensate the computed result. The accumulated errors of each global point are added to its final value. Fig.1(Right) exhibits the CompSum reproducibility and the same good properties already mentioned for IntSum.

ReprodSum modifies FPSum with the reproducible sum from [1]. It uses a pre-rounding technique that rounds the input values to a common threshold that depends on the maximum and on the size of the entry vector. It does not depend on the computation order and is almost as accurate as FPSum.
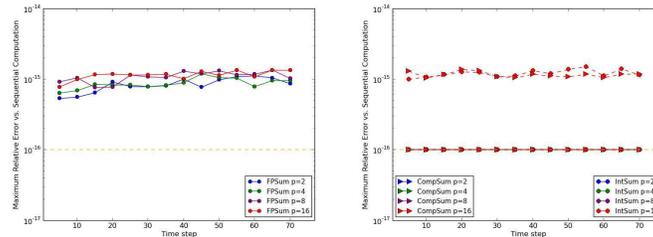

Fig. 1: Reproducibility and accuracy improvement for the mean wave frequency (Nice test case, Tomawac). Left: Parallel vs. sequential FPSum. Right: IntSum, CompSum and their accuracy compared to sequential FPSum (dashed).

References

1. Demmel, J.W., Nguyen, H.D.: Fast reproducible floating-point summation. In: Proc. 21th IEEE Symposium on Computer Arithmetic. Austin, Texas, USA (2013)
2. Hervouet, J.M.: Hydrodynamics of free surface flows: Modelling with the finite element method. John Wiley & Sons (2007)
3. Ogita, T., Rump, S.M., Oishi, S.: Accurate sum and dot product. SIAM J. Sci. Comput. 26(6), 1955–1988 (2005)
4. open TELEMAC-MASCARET: www.opentelemac.org, v.7.0, Release notes (2014)