Estimation of numerical reproducibility on CPU and GPU

{ Fabienne.Jezequel, Jean-Luc.Lamotte & Issam.Said }@lip6.fr

Problem

Results of numerical simulations may be **different from one architecture to another**, or even **inside the same architecture**.

In sequential or parallel environments, different orders in the sequence of floating-point operations may lead to differences in rounding error propagation and therefore to reproducibility failures.

How to identify the cause of differences: **rounding errors or bug?**

Reproducibility failures in a wave propagation code

The 3D acoustic wave equation $\frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} - \sum_{b \in x,y,z} \frac{\partial^2}{\partial b^2} u = 0$ where u is the acoustic pressure, c is the wave velocity and t is the time is solved using a finite difference scheme with time order 2 and space order 8.

Differences in the results are observed from one architecture to another, from one execution to another inside a GPU, and from one implementation of the finite difference scheme to another.

In *binary32*, for $64 \times 64 \times 64$ space steps and 1000 time iterations any two results at the same space coordinates have **0 to 7 common digits** and the average number of common digits is about 4.

Accuracy estimation with CADNA

The CADNA library [1, 2] estimates rounding errors using **Discrete Stochastic Arithmetic** [3, 4]. Each arithmetical operation is executed 3 times with the random rounding mode: each result is rounded up or down with the probability 1/2.

CADNA provides new numerical types which consist of 3 floating point variables and an integer variable to store the accuracy. All operators and mathematical functions are redefined for these types.

⇒ **CADNA requires only a few modifications in user programs**, mainly changes in type declarations.

Example: an execution with CADNA

Let $f(x, y) = 9x^4 - y^4 + 2y^2$. $f(10864, 18817)$ and $f(\frac{1}{3}, \frac{2}{3})$ are computed with CADNA.

CADNA prints results with only their **digits not affected by rounding errors** and **detects numerical instabilities**:

```
f(10864,18817) = @.0 (no correct digit)
f(1/3,2/3) = 0.802469135802469E+000
There are 2 numerical instabilities
2 LOSS(ES) OF ACCURACY DUE TO CANCELLATION(S)
```

Deployment of CADNA on GPU in CUDA language

Rounding mode change:

- On CPU, the rounding mode is frequently switched from $+\infty$ to $-\infty$, or from $-\infty$ to $+\infty$.
- On GPU, an arithmetic operation can be performed with a specified rounding mode: e.g. `fmul_ru` for multiplication rounded to $+\infty$.

Instability detection:

- On CPU, dedicated counters are incremented.
- On GPU, such counters would need a lot of atomic operations.
  ⇒ An unsigned char is associated with each result (each bit is associated with a type of instability).

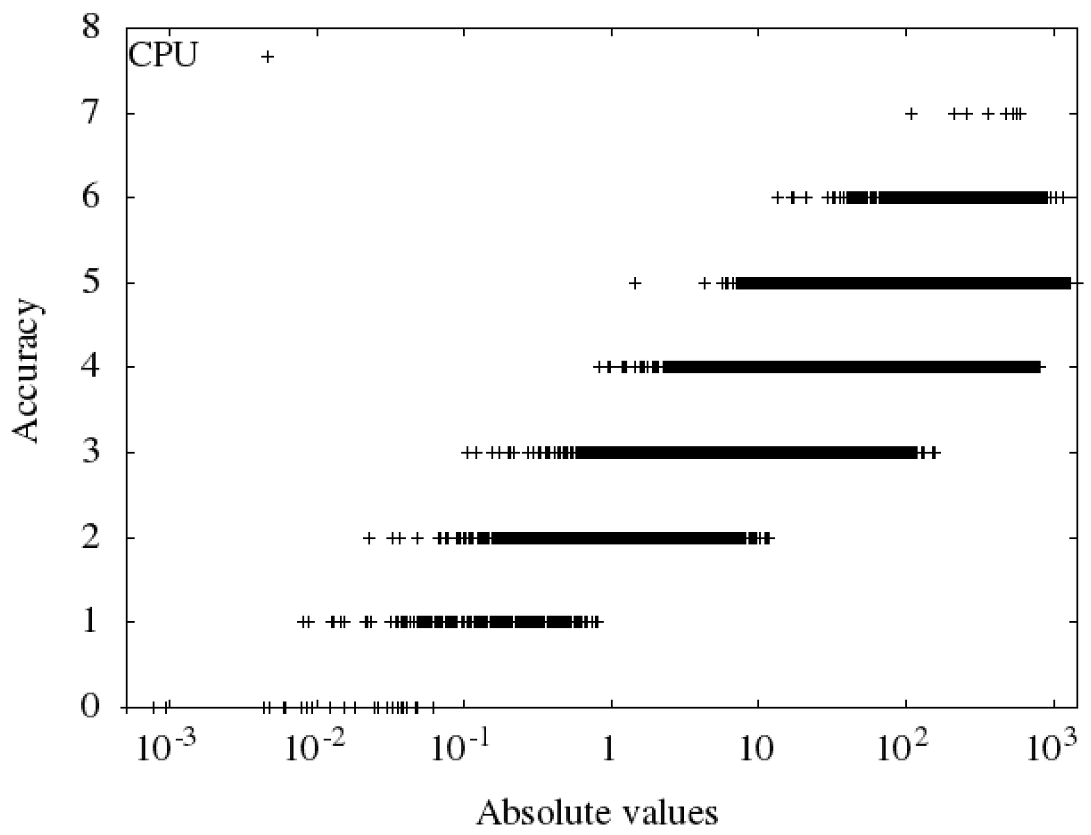
The acoustic wave propagation code with CADNA

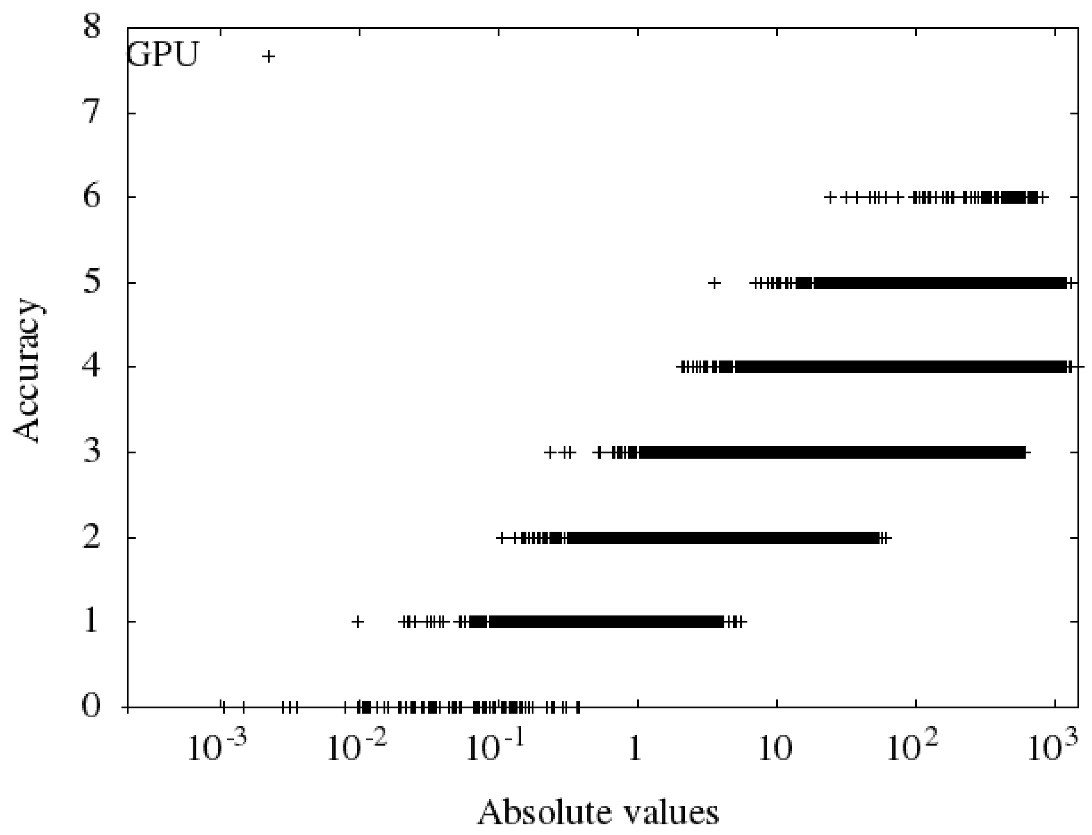
The number of exact digits estimated by CADNA on an AMD Opteron 6168 CPU and an NVIDIA C2050 GPU **varies from 0 to 7**. Its mean value is 4.1 on CPU and 3.5 on GPU.
⇒ This is consistent with our previous observations.

Results computed at 3 different points in the space domain:

	$p_1 = (0, 19, 62)$	$p_2 = (50, 12, 2)$	$p_3 = (20, 1, 46)$
IEEE CPU	-1.110479E+0	5.454238E+1	6.141038E+2
IEEE GPU	-1.110204E+0	5.454224E+1	6.141046E+2
CADNA CPU	-1.1E+0	5.454E+1	6.14104E+2
CADNA GPU	-1.11E+0	5.45E+1	6.1410E+2
Reference	-1.108603879E+0	5.454034021E+1	6.141041156E+2

Despite differences in the estimated accuracy, **the same trend can be observed on CPU and on GPU**: highest rounding errors impact negligible results and highest results are impacted by low rounding errors.





Conclusion

Discrete Stochastic Arithmetic can estimate which digits are affected by **rounding errors** and **possibly explain reproducibility failures**.

Related works:

- taking advantage of SIMD instructions (SSE, AVX, Xeon Phi) [5]
- CADNA for MPI codes
- CADNA for OpenMP codes.

References

[1] The CADNA library `http://www.lip6.fr/cadna`

[2] F. Jézéquel and J.-M. Chesneaux, *CADNA: a library for estimating round-off error propagation*, Computer Physics Communications, vol. 178, no. 12, pp. 933-955, 2008.

[3] J. Vignes, *A stochastic arithmetic for reliable scientific computation*, Mathematics and Computers in Simulation, vol. 35, pp. 233-261, 1993.

[4] J. Vignes, *Discrete Stochastic Arithmetic for validating results of numerical software*, Numerical Algorithms, vol. 37, no. 1-4, pp. 377-390, 2004.

[5] P. Eberhart, J. Brajard, P. Fortin, and F. Jézéquel, *Towards high performance stochastic arithmetic*, in The 16th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics (SCAN'14), Würzburg, Germany, Sep. 2014.