Numerical reproducibility of high-performance computations using floating-point or interval arithmetic

Nathalie Revol, INRIA

Experimental mathematical computations usually require intensive and numerically guaranteed computations. We identify problems encountered when these computations use floating-point or interval arithmetic, on high-performance environments. Depending mostly on the architecture, floating-point computations can be performed using different precisions, classically single, double or long double, and this can yield different results. The order in which operations are performed can also impact the result: mathematical operations such as the addition or the multiplication are associative, but their floating-point counterparts are not. The consequence is that the result of a floating-point program can vary from one execution to another. This issue is known as numerical reproducibility.

Interval computations return intervals which enclose the exact results. Often, interval arithmetic is implemented using floating-point arithmetic. Interval computations thus suffer from the same problems of numerical reproducibility as floating-point computations, namely variable computing precision and variable order of operations. Furthermore, the guarantee offered by interval arithmetic holds only if directed rounding modes are used. The respect of directed rounding modes is often an issue, especially for HPC. For each problem, recommendations will be given to preserve the guarantee offered by interval arithmetic and to alleviate the impact of this problem on the quality (that is, the width) of the results.