# Quick Addition Checker Program Description

Run a series of Java DP ADDITION tests to validate accuracy along estimate execution timing.

A raw test sequence is a sequence of natural numbers that fit in a Java double-precision floating-point value without loss of significance (see [Oracle Docs](https://docs.oracle.com/javase/specs/jvms/se6/html/Overview.doc.html).) Java DP values have 53 bits of significance and hold a natural number exactly up to 9,007,199,254,740,991. This is a little more than 15 digits.

When computing sums of natural numbers, exact DP calculations limit the sequence to 134,164,078 with an integer summation value of 8,999,999,979,877,081. This is our "*Natural*" test sequence of numbers that our *adder* implementations will sum.

We also create a "*scaled*" test sequence by dividing a natural number test sequence by a prime number; we do this to force round off error into the sequence, and therefore error into the computed sum (e.g., the prime *7919*.)

Given an initial sequence of natural numbers, and fractions from those natural numbers (scaled by a prime), we can then randomly order those numbers with respect to addition. Order is important because naive addition performs best when summed smallest-to-largest, and worst when ordered largest-to-smallest. Randomized sums are considered an average case and typically have errors between the other two orders. Finally, we can employ Java parallelized addition, which has the effect of performing separate sums on smaller sequences and adding together the intermediate sums to create the final sum. We also test an accuracy enhanced summation technique (Kahn Addition.)

Please note that our tests have sizes are increasing in a roughly log-linear sequence. We also time the overall tests for performance analysis.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sequence** | **Order** | **Distribution** | **Label** |
| Natural | Smallest first | Sequential | NAT-SML-SEQ |
| Natural | Largest first | Sequential | NAT-LRG-SEQ |
| Natural | Random | Parallel | NAT-RND-PAR |
| Scaled | Smallest first | Sequential | SCL-SML-SEQ |
| Scaled | Largest first | Sequential | SCL-LRG-SEQ |
| Scaled | Random | Sequential | SCL-RND-SEQ |
| Scaled | Random | Parallel | SCL-RND-PAR |
| Scaled-Kahan | Random | Sequential | SCLK-RND-SEQ |

Test output is in CSV format to allow Excel analysis. The columns in the report are:

* *n* - number of values to sum
* *expected* - the expected or true sum
* *actual* - the observed or computed sum
* *delta* - the difference between expected and actual (expected / actual)
* *relative* - the relative error: (expected - actual) / expected; multiplied by 10\*\*9
* *sigd* - the number of significant digits [computed as log10(0.5) - log10(abs(relative))]
* *elapsed* - summation time in milliseconds
* *label* - test identification (taken from table above)

**Author:**

Donald Trummell

## Java Parallelization References:

* Java 8 parallelism selector: <http://gee.cs.oswego.edu/dl/html/StreamParallelGuidance.html>.
* General Java parallelism benchmark: <https://dzone.com/articles/forkjoin-framework-vs-parallel>.