CMSC/AMSC 460 Fall 2007
Homework 1
Due Tuesday, September 18, before class begins
10 points

Under fixed point arithmetic, if we add $n$ numbers, then either we get the exact answer or we get an "overflow" error.

Under floating point arithmetic, this is not true. We probably get an approximation to the exact answer, although we could possibly get the exact answer or overflow.

Consider the three examples in the program `hw1.m`. You might want to play with the program, changing $n$ or $h$ or computing the difference between the computed sums and the true ones in order to understand what is happening.

(a) (1) Is sum1 equal to the true value? If not, why not? (Be specific: it is not enough to say, "Round-off caused the error." If sum1 is not exact, in what part of the calculation did the error arise?)

(b) (3) Are sum2 and sum3 equal to the true value? If not, why not? And why aren't they equal to each other?

(c) (3) Are sum4 and sum5 equal to the true value? If not, why not? And why aren't they equal to each other?

(d) (3) Given $n$ numbers to add, what ordering do you advise using to keep round-off error small?