Joey Arcangeletti

MEE 381

Assignment 1

Dr. Coller

Question 1)   
Which is faster: adding two floats together, or two doubles? To do this one, you’re going tohave to create two separate arrays: one with floats, and the other with doubles. To do a faircomparison, both arrays will need to have (essentially) the same sets of numbers that are addedtogether. First create the array of random doubles. Then cast those doubles into floats to fill thesame-size array of floats. You’ll probably need two different functions which computes the sum

For this question I have found that it is usually faster to use floats for calculations rather than using doubles. The problem with using floats is that it is not as high precision as doubles are. So, it is rather common to use floats for usual programming and simulations. But if it is required to have a high precision then doubles should be used. The reason that the floats are faster is because floats use a 32 bit memory. Doubles on the other hand use a 64-bit memory. Because the floats use a smaller memory the CPU can compute 32-Bit faster than the 64-Bit. This is why it is faster to add two floats together versus two doubles together.

Question 2)

Which is faster: computing the square of a double via direct multiplication, or computing thesquare using the built in function Math.Pow function? Also compare these to the built-insquare root function: Math.Sqrt function.

For this question, when comparing the two methods we can first look at computing the square of a double via direct multiplication. Because this method should be a more straightforward approach, we would assume that this is the more fast method. The Math.Pow function is used for more general calculations; it is slower than the straightforward multiplication but requires more computing power. The Math.Sqrt function is a function to calculate square roots. This function requires the most computing power as it is the most intensive to calculate. When we compare all of them together, we can start with the fastest being computing the square via direct multiplication. This one is also the simplest so perhaps that is a trend we are seeing. It is a very simple operation and therefore computes the fastest. The Math.Pow function is second on the list as it takes more computational energy to do. Lastly the Math.Sqrt function is the slowest as it takes the most computing power compared to squaring and direct multiplication. Depending on the hardware and software being used could alter these results but in our case we find that squaring a double via direct multiplication is the fastest method.