Department of Computing Science and Mathematics
University of Stirling

CSC9V4 Systems I - Tutorial 3

Week beginning 26 February 2018

1. Explain why a simple (apparently exact) floating-point number like 1.2f (say) is represented on a computer by an approximation. What are the consequences of this when two (or three, or a hundred) floating-point numbers are added?

2. What is meant by underflow, in respect of floating-point arithmetic? Write down an expression that might result in underflow

3. What is meant by overflow, in respect of floating-point arithmetic? Write down an expression that might result in (i) positive overflow and (ii) negative overflow.

4. 23 bits are used for the fractional part and 8 bits are used for the exponent in the IEEE 754 Standard for single-precision floating-point numbers. Using the standard "calculate" the largest number that can be represented. What is the smallest normalised number? What range of values does the denormalised form of the standard give you? Show how you calculated the values.

5. Consider (i) multiplying two floating-point numbers (ii) dividing one floating-point number by another. For what sorts of calculations are these operations likely to produce (relatively) large inaccuracies? Can you describe conditions where these inaccuracies might mount up? How might one avoid this?

6. You want to calculate x/a – y/a where you know that x and y may be very close to each other, and a is a large number. How would you do this in such a way as to minimise the rounding error?