7 Floating-Point Computation (DJG)

The following two functions are algorithms for exponentiation where \( x \) is a single-precision floating-point value and \( n \) is an integer,

\[
\text{fun power1}(x, n) = \begin{cases} 
1.0 & \text{if } n=0 \\
 x \times \text{power1}(x, n-1) & \text{else} 
\end{cases}
\]

\[
\text{fun power2}(x, n) = \begin{cases} 
1.0 & \text{if } n=0 \\
 x \times \text{power2}(x, n-1) & \text{else if } \text{even } n \\
 x \times \text{power2}(x, n \div 2) & \text{else} 
\end{cases}
\]

(a) What is, roughly, the largest value of \( n \) that can be used without overflow when \( x = 10.0 \)? [1 mark]

(b) Suppose \( x \) is close to 1.0.

(i) What is the worst possible relative error to expect in the answer from \( \text{power1} \) when \( n = 100 \)? [3 marks]

(ii) Can we say anything useful about the absolute error in part (b)(i)? [1 mark]

(iii) What is the expected value of the relative error in results from \( \text{power1} \)? [1 mark]

(c) Sometimes the expected magnitude of error can be estimated as the result of a random walk.

(i) Under what conditions is this appropriate? [2 marks]

(ii) What is the random walk estimate for the relative error in part (b)(i)? [3 marks]

(d) If \( x \) is again close to 1.0, what is the worst possible relative error to expect from \( \text{power2} \) when \( n = 100 \)? [6 marks]

(e) For what range or class of \( x \) values will \( \text{power2} \) with \( n = 100 \) give a result with no error? [3 marks]