Bare Essentials

At the end of this chapter you should be able to

1. List the digits used in base two arithmetic.
2. Give a simple explanation of the term "floating point number".
3. Give two examples of numbers that would need to be stored in floating point format in a computer. What other format is used for numerical data? Give two examples of this other format.
4. Give a definition of "roundoff error".
5. Explain the expression, "There are holes in the floating point number line".
6. Explain why integer arithmetic is "exact"?
7. Describe why floating point arithmetic is not "exact".
8. Give a simple explanation of overflow.
9. Give a simple expression that defines machine precision, ε_m.
10. Give the value of ε_m for exact arithmetic.
11. Write the formulas for computing the relative and absolute errors if α is an exact (scalar) value, and $\hat{\alpha}$ is its floating point approximation.
12. Identify the truncation error of a Taylor series expansion.

A minimal working knowledge of MATLAB requires that you can

13. Name the built in variable that holds the value ε_m.
14. Given the value of ε_m for computations in MATLAB.
15. Write a simple, but carefully coded `if` statement that determines whether two scalar values are close enough to be considered equal.

An Expanded Core of Knowledge

After mastering the bare essentials you should move on to a deeper understanding of the fundamentals. Doing so involves being able to

1. Give a definition and one example of catastrophic cancellation error.
2. Identify (at least) two important differences between symbolic and numeric computations.

3. Write an expression that combines an absolute *and* relative tolerance to compare computed result with a target or exact value. Let α be the exact (scalar) value, and $\hat{\alpha}$ be its floating point approximation.

4. Use an infinite series to give an example of truncation error.

5. Use order notation to express truncation error.

Developing Mastery

A mastery of computational errors requires that you

1. Readily convert between binary and decimal notation

2. Can sketch the floating point number line and label its major features.

3. Explain the difference between generating a denormal and rounding to zero.

4. Be able to distinguish the effects of roundoff and truncation errors in a computed result, for example, by viewing a plot such as Figure (5.4)