to simply using it, see [43, 80, 127, 328, 354]. Additional computer exercises and projects can be found in [89, 94, 130, 155, 167, 170, 200, 203, 205, 294, 353, 418].

Review Questions

1.1. True or false: A problem is ill-conditioned if its solution is highly sensitive to small changes in the problem data.

1.2. True or false: Using higher-precision arithmetic will make an ill-conditioned problem better conditioned.

1.3. True or false: The conditioning of a problem depends on the algorithm used to solve it.

1.4. True or false: A good algorithm will produce an accurate solution regardless of the condition of the problem being solved.

1.5. True or false: The choice of algorithm for solving a problem has no effect on the propagated data error.

1.6. True or false: A stable algorithm applied to a well-conditioned problem necessarily produces an accurate solution.

1.7. True or false: If two real numbers are exactly representable as floating-point numbers, then the result of a real arithmetic operation on them will also be representable as a floating-point number.

1.8. True or false: Floating-point numbers are distributed uniformly throughout their range.

1.9. True or false: Floating-point addition is associative but not commutative.

1.10. True or false: In a floating-point number system, the underflow level is the smallest positive number that perturbs the number 1 when added to it.

1.11. True or false: The mantissa in IEEE double precision floating-point arithmetic is exactly twice the length of the mantissa in IEEE single precision.

1.12. What three properties characterize a *well-posed* problem?

1.13. List three sources of error in scientific computation.

1.14. Explain the distinction between truncation (or discretization) and rounding.

1.15. Explain the distinction between absolute error and relative error.

1.16. Explain the distinction between computational error and propagated data error.

1.17. Explain the distinction between precision and accuracy.

1.18. (*a*) What is meant by the *conditioning* of a problem?

(*b*) Is it affected by the algorithm used to solve the problem?

(*c*) Is it affected by the precision of the arithmetic used to solve the problem?

1.19. If a computational problem has a condition number of 1, is this good or bad? Why?

1.20. Explain the distinction between relative condition number and absolute condition number.

1.21. What is an inverse problem? How are the conditioning of a problem and its inverse related?

1.22. (*a*) What is meant by the *backward error* in a computed result?

(*b*) When is an approximate solution to a given problem considered to be good according to backward error analysis?

1.23. Suppose you are solving a given problem using a given algorithm. For each of the following, state whether it is affected by the *stability* of the algorithm, and why.

(*a*) Propagated data error

(*b*) Accuracy of computed result

(*c*) Conditioning of problem

1.24. (*a*) Explain the distinction between forward error and backward error.

(*b*) How are forward error and backward error related to each other quantitatively?

1.25. For a given floating-point number system, describe in words the distribution of machine numbers along the real line.

1.26. In floating-point arithmetic, which is generally more harmful, underflow or overflow? Why?

1.27. In floating-point arithmetic, which of the following operations on two positive floating-point operands can produce an overflow?
(*a*) Addition
(*b*) Subtraction
(*c*) Multiplication
(*d*) Division

1.28. In floating-point arithmetic, which of the following operations on two positive floating-point operands can produce an underflow?
(*a*) Addition
(*b*) Subtraction
(*c*) Multiplication
(*d*) Division

1.29. List two reasons why floating-point number systems are usually normalized.

1.30. In a floating-point system, what quantity determines the maximum relative error in representing a given real number by a machine number?

1.31. (*a*) Explain the difference between the rounding rules "round toward zero" and "round to nearest" in a floating-point system.
(*b*) Which of these two rounding rules is more accurate?
(*c*) What quantitative difference does this make in the unit roundoff ϵ_{mach}?

1.32. In a p-digit binary floating-point system with rounding to nearest, what is the value of the unit roundoff ϵ_{mach}?

1.33. In a floating-point system with gradual underflow (subnormal numbers), is the representation of each number still unique? Why?

1.34. In a floating-point system, is the product of two machine numbers usually exactly representable in the floating-point system? Why?

1.35. In a floating-point system, is the quotient of two nonzero machine numbers always exactly representable in the floating-point system? Why?

1.36. (*a*) Give an example to show that floating-point addition is not necessarily associative.
(*b*) Give an example to show that floating-point multiplication is not necessarily associative.

1.37. (*a*) In what circumstances does *cancellation* occur in a floating-point system?
(*b*) Does the occurrence of cancellation imply that the true result of the specific operation causing it is not exactly representable in the floating-point system? Why?
(*c*) Why is cancellation usually bad?

1.38. Give an example of a number whose decimal representation is finite (i.e., it has only a finite number of nonzero digits) but whose binary representation is not.

1.39. Give examples of floating-point arithmetic operations that would produce each of the exceptional values `Inf` and `NaN`.

1.40. In a floating-point system with base β, precision p, and rounding to nearest, what is the maximum relative error in representing any nonzero real number within the range of the system?

1.41. Explain why the cancellation that occurs when two numbers of similar magnitude are subtracted is often bad even though the result may be exactly correct for the actual operands involved.

1.42. Assume a decimal (base 10) floating-point system having machine precision $\epsilon_{\text{mach}} = 10^{-5}$ and an exponent range of ± 20. What is the result of each of the following floating-point arithmetic operations?
(*a*) $1 + 10^{-7}$
(*b*) $1 + 10^{3}$
(*c*) $1 + 10^{7}$
(*d*) $10^{10} + 10^{3}$
(*e*) $10^{10}/10^{-15}$
(*f*) $10^{-10} \times 10^{-15}$

1.43. In a floating-point number system having an underflow level of $\text{UFL} = 10^{-38}$, which of the following computations will incur an underflow?
(*a*) $a = \sqrt{b^2 + c^2}$, with $b = 1$, $c = 10^{-25}$.
(*b*) $a = \sqrt{b^2 + c^2}$, with $b = c = 10^{-25}$.
(*c*) $u = (v \times w)/(y \times z)$, with $v = 10^{-15}$, $w = 10^{-30}$, $y = 10^{-20}$, and $z = 10^{-25}$.
In each case where underflow occurs, is it reasonable simply to set to zero the quantity that underflows?

1.44. (*a*) Explain in words the difference between the unit roundoff, ϵ_{mach}, and the underflow level, UFL, in a floating-point system.
Of these two quantities,
(*b*) Which one depends only on the number of digits in the mantissa field?

(c) Which one depends only on the number of digits in the exponent field?

(d) Which one does *not* depend on the rounding rule used?

(e) Which one is *not* affected by allowing subnormal numbers?

1.45. Let x_k be a monotonically decreasing, finite sequence of positive numbers (i.e., $x_k > x_{k+1}$ for each k). Assuming it is practical to take the numbers in any order we choose, in what order should the sequence be summed to minimize rounding error?

1.46. Is cancellation an example of rounding error? Why?

1.47. (a) Explain why a divergent infinite series, such as

$$\sum_{n=1}^{\infty} \frac{1}{n},$$

can have a finite sum in floating-point arithmetic.

(b) At what point will the partial sums cease to change?

1.48. In floating-point arithmetic, if you are computing the sum of a convergent infinite series

$$S = \sum_{i=1}^{\infty} x_i$$

of positive terms in the natural order, what stopping criterion would you use to attain the maximum possible accuracy using the smallest number of terms?

1.49. Explain why an infinite series with alternating signs, such as

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

for $x < 0$, is difficult to evaluate accurately in floating-point arithmetic.

1.50. If f is a real-valued function of a real variable, the truncation error of the finite difference approximation to the derivative

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

goes to zero as $h \to 0$. If we use floating-point arithmetic, list two factors that limit how small a value of h we can use in practice.

1.51. List at least two ways in which evaluation of the quadratic formula

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

may suffer numerical difficulties in floating-point arithmetic.

Exercises

1.1. The average normal human body temperature is usually quoted as 98.6 degrees Fahrenheit, which might be presumed to have been determined by computing the average over a large population and then rounding to three significant digits. In fact, however, 98.6 is simply the Fahrenheit equivalent of 37 degrees Celsius, which is accurate to only two significant digits.

(a) What is the maximum relative error in the accepted value, assuming it is accurate to within $\pm 0.05°$ F?

(b) What is the maximum relative error in the accepted value, assuming it is accurate to within $\pm 0.5°$ C?

1.2. What are the approximate absolute and relative errors in approximating π by each of the following quantities?

(a) 3

(b) 3.14

(c) 22/7

1.3. If a is an approximate value for a quantity whose true value is t, and a has relative error r, prove from the definitions of these terms that $a = t(1 + r)$.

1.4. Consider the problem of evaluating the function $\sin(x)$, in particular, the propagated data error, i.e., the error in the function value due to a perturbation h in the argument x.

(a) Estimate the absolute error in evaluating $\sin(x)$.

(b) Estimate the relative error in evaluating