The solution in such cases is to use an alternative Clenshaw recurrence that incorporates c_k's in an upward direction. The relevant equations are

$$y_{-2} = y_{-1} = 0 \tag{5.5.25}$$

$$y_k = \frac{1}{\beta(k+1,x)}[y_{k-2} - \alpha(k,x)y_{k-1} - c_k], \qquad (k = 0, 1, \ldots, N-1) \tag{5.5.26}$$

$$f(x) = c_N F_N(x) - \beta(N,x)F_{N-1}(x)y_{N-1} - F_N(x)y_{N-2} \tag{5.5.27}$$

The rare case where equations (5.5.25)–(5.5.27) should be used instead of equations (5.5.21) and (5.5.23) can be detected automatically by testing whether the operands in the first sum in (5.5.23) are opposite in sign and nearly equal in magnitude. Other than in this special case, Clenshaw's recurrence is always stable, independent of whether the recurrence for the functions F_k is stable in the upward or downward direction.

CITED REFERENCES AND FURTHER READING:

Abramowitz, M., and Stegun, I.A. 1964, *Handbook of Mathematical Functions*, Applied Mathematics Series, Volume 55 (Washington: National Bureau of Standards; reprinted 1968 by Dover Publications, New York), pp. xiii, 697. [1]

Gautschi, W. 1967, *SIAM Review*, vol. 9, pp. 24–82. [2]

Lakshmikantham, V., and Trigiante, D. 1988, *Theory of Difference Equations: Numerical Methods and Applications* (San Diego: Academic Press). [3]

Acton, F.S. 1970, *Numerical Methods That Work*; 1990, corrected edition (Washington: Mathematical Association of America), pp. 20ff. [4]

Clenshaw, C.W. 1962, *Mathematical Tables*, vol. 5, National Physical Laboratory (London: H.M. Stationery Office). [5]

Dahlquist, G., and Bjorck, A. 1974, *Numerical Methods* (Englewood Cliffs, NJ: Prentice-Hall), §4.4.3, p. 111.

Goodwin, E.T. (ed.) 1961, *Modern Computing Methods*, 2nd ed. (New York: Philosophical Library), p. 76.

5.6 Quadratic and Cubic Equations

The roots of simple algebraic equations can be viewed as being functions of the equations' coefficients. We are taught these functions in elementary algebra. Yet, surprisingly many people don't know the right way to solve a quadratic equation with two real roots, or to obtain the roots of a cubic equation.

There are two ways to write the solution of the *quadratic equation*

$$ax^2 + bx + c = 0 \tag{5.6.1}$$

with real coefficients a, b, c, namely

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{5.6.2}$$

and

$$x = \frac{2c}{-b \pm \sqrt{b^2 - 4ac}} \tag{5.6.3}$$

If you use *either* (5.6.2) *or* (5.6.3) to get the two roots, you are asking for trouble: If either a or c (or both) are small, then one of the roots will involve the subtraction of b from a very nearly equal quantity (the discriminant); you will get that root very inaccurately. The correct way to compute the roots is

$$q \equiv -\frac{1}{2}\left[b + \text{sgn}(b)\sqrt{b^2 - 4ac}\right] \tag{5.6.4}$$

Then the two roots are

$$x_1 = \frac{q}{a} \qquad \text{and} \qquad x_2 = \frac{c}{q} \tag{5.6.5}$$

If the coefficients a, b, c, are complex rather than real, then the above formulas still hold, except that in equation (5.6.4) the sign of the square root should be chosen so as to make

$$\text{Re}(b^* \sqrt{b^2 - 4ac}) \geq 0 \tag{5.6.6}$$

where Re denotes the real part and asterisk denotes complex conjugation.

Apropos of quadratic equations, this seems a convenient place to recall that the inverse hyperbolic functions $\sinh^{-1}$ and $\cosh^{-1}$ are in fact just logarithms of solutions to such equations,

$$\sinh^{-1}(x) = \ln\left(x + \sqrt{x^2 + 1}\right) \tag{5.6.7}$$

$$\cosh^{-1}(x) = \pm \ln\left(x + \sqrt{x^2 - 1}\right) \tag{5.6.8}$$

Equation (5.6.7) is numerically robust for $x \geq 0$. For negative x, use the symmetry $\sinh^{-1}(-x) = -\sinh^{-1}(x)$. Equation (5.6.8) is of course valid only for $x \geq 1$.

For the *cubic equation*

$$x^3 + ax^2 + bx + c = 0 \tag{5.6.9}$$

with real or complex coefficients a, b, c, first compute

$$Q \equiv \frac{a^2 - 3b}{9} \qquad \text{and} \qquad R \equiv \frac{2a^3 - 9ab + 27c}{54} \tag{5.6.10}$$

If Q and R are real (always true when a, b, c are real) *and* $R^2 < Q^3$, then the cubic equation has three real roots. Find them by computing

$$\theta = \arccos(R/\sqrt{Q^3}) \tag{5.6.11}$$

in terms of which the three roots are

$$
\begin{aligned}
x_1 &= -2\sqrt{Q}\cos\left(\frac{\theta}{3}\right) - \frac{a}{3} \\
x_2 &= -2\sqrt{Q}\cos\left(\frac{\theta+2\pi}{3}\right) - \frac{a}{3} \\
x_3 &= -2\sqrt{Q}\cos\left(\frac{\theta-2\pi}{3}\right) - \frac{a}{3}
\end{aligned}
\tag{5.6.12}
$$

(This equation first appears in Chapter VI of François Viète's treatise "De emendatione," published in 1615!)

Otherwise, compute

$$A = -\left[R + \sqrt{R^2 - Q^3}\right]^{1/3} \tag{5.6.13}$$

where the sign of the square root is chosen to make

$$\mathrm{Re}(R^* \sqrt{R^2 - Q^3}) \geq 0 \tag{5.6.14}$$

(asterisk again denoting complex conjugation). If Q and R are both real, equations (5.6.13)–(5.6.14) are equivalent to

$$A = -\mathrm{sgn}(R)\left[|R| + \sqrt{R^2 - Q^3}\right]^{1/3} \tag{5.6.15}$$

where the positive square root is assumed. Next compute

$$B = \begin{cases} Q/A & (A \neq 0) \\ 0 & (A = 0) \end{cases} \tag{5.6.16}$$

in terms of which the three roots are

$$x_1 = (A + B) - \frac{a}{3} \tag{5.6.17}$$

(the single real root when a, b, c are real) and

$$
\begin{aligned}
x_2 &= -\frac{1}{2}(A+B) - \frac{a}{3} + i\frac{\sqrt{3}}{2}(A-B) \\
x_3 &= -\frac{1}{2}(A+B) - \frac{a}{3} - i\frac{\sqrt{3}}{2}(A-B)
\end{aligned}
\tag{5.6.18}
$$

(in that same case, a complex conjugate pair). Equations (5.6.13)–(5.6.16) are arranged both to minimize roundoff error, and also (as pointed out by A.J. Glassman) to ensure that no choice of branch for the complex cube root can result in the spurious loss of a distinct root.

If you need to solve many cubic equations with only slightly different coefficients, it is more efficient to use Newton's method (§9.4).

CITED REFERENCES AND FURTHER READING:

Weast, R.C. (ed.) 1967, *Handbook of Tables for Mathematics*, 3rd ed. (Cleveland: The Chemical Rubber Co.), pp. 130–133.

Pachner, J. 1983, *Handbook of Numerical Analysis Applications* (New York: McGraw-Hill), §6.1.

McKelvey, J.P. 1984, *American Journal of Physics*, vol. 52, pp. 269–270; see also vol. 53, p. 775, and vol. 55, pp. 374–375.

5.7 Numerical Derivatives

Imagine that you have a procedure which computes a function $f(x)$, and now you want to compute its derivative $f'(x)$. Easy, right? The definition of the derivative, the limit as $h \to 0$ of

$$f'(x) \approx \frac{f(x+h) - f(x)}{h} \tag{5.7.1}$$

practically suggests the program: Pick a small value h; evaluate $f(x+h)$; you probably have $f(x)$ already evaluated, but if not, do it too; finally apply equation (5.7.1). What more needs to be said?

Quite a lot, actually. Applied uncritically, the above procedure is almost guaranteed to produce inaccurate results. Applied properly, it can be the right way to compute a derivative only when the function f is *fiercely* expensive to compute, when you already have invested in computing $f(x)$, and when, therefore, you want to get the derivative in no more than a single additional function evaluation. In such a situation, the remaining issue is to choose h properly, an issue we now discuss:

There are two sources of error in equation (5.7.1), truncation error and roundoff error. The truncation error comes from higher terms in the Taylor series expansion,

$$f(x+h) = f(x) + hf'(x) + \frac{1}{2}h^2 f''(x) + \frac{1}{6}h^3 f'''(x) + \cdots \tag{5.7.2}$$

whence

$$\frac{f(x+h) - f(x)}{h} = f' + \frac{1}{2}hf'' + \cdots \tag{5.7.3}$$

The roundoff error has various contributions. First there is roundoff error in h: Suppose, by way of an example, that you are at a point $x = 10.3$ and you blindly choose $h = 0.0001$. Neither $x = 10.3$ nor $x + h = 10.30001$ is a number with an exact representation in binary; each is therefore represented with some fractional error characteristic of the machine's floating-point format, ϵ_m, whose value in single precision may be $\sim 10^{-7}$. The error in the *effective* value of h, namely the difference between $x + h$ and x as represented in the machine, is therefore on the order of $\epsilon_m x$, which implies a fractional error in h of order $\sim \epsilon_m x/h \sim 10^{-2}$! By equation (5.7.1) this immediately implies at least the same large fractional error in the derivative.

We arrive at Lesson 1: Always choose h so that $x + h$ and x differ by an exactly representable number. This can usually be accomplished by the program steps

$$\begin{aligned} \texttt{temp} &= x + h \\ h &= \texttt{temp} - x \end{aligned} \tag{5.7.4}$$

Some optimizing compilers, and some computers whose floating-point chips have higher internal accuracy than is stored externally, can foil this trick; if so, it is usually enough to declare `temp` as `volatile`, or else to call a dummy function `donothing(temp)` between the two equations (5.7.4). This forces `temp` into and out of addressable memory.