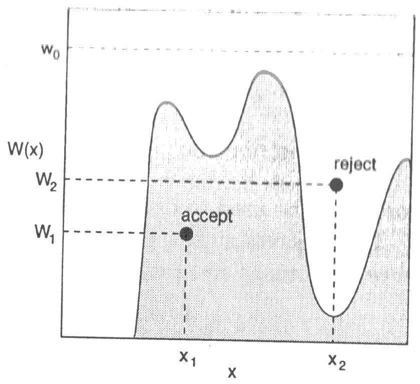

Fig. 7.5 The von Neumann rejection technique for generating random points with weight $w(x)$.

and apply a variation of the rejection technique known as the *Metropolis algorithm.* That algorithm has now become the cornerstone of computation thermodynamics.

7.20 ASSESSMENT⊙

Use the von Neumann rejection technique to generate a normal distribution of standard deviation 1, and compare to the preceding *Box–Muller* method.

8 Differentiation

8.1 PROBLEM 1: NUMERICAL LIMITS

A particle is moving through space, and you record its position as a function of time $x(t)$ in a table. Your **problem** is to determine its velocity $v(t) = dx/dt$ when all you have is this table of x versus t.

8.2 METHOD: NUMERIC

You probably did rather well in your first calculus course and feel competent at taking derivatives. However, you probably did not take derivatives of a table of numbers using the elementary definition:

$$\frac{df(x)}{dx} \overset{\text{def}}{=} \lim_{h\to 0} \frac{f(x+h)-f(x)}{h}. \tag{8.1}$$

In fact, even a computer runs into errors with this kind of limit because it is wrought with subtractive cancellation; the computer's finite word length causes the numerator to fluctuate between 0 and the machine precision ϵ_m as the denominator approaches zero.

8.2.1 Method: Forward Difference

The most direct method for numerical differentiation of a function starts by expanding it in a Taylor series. This series advances the function one small

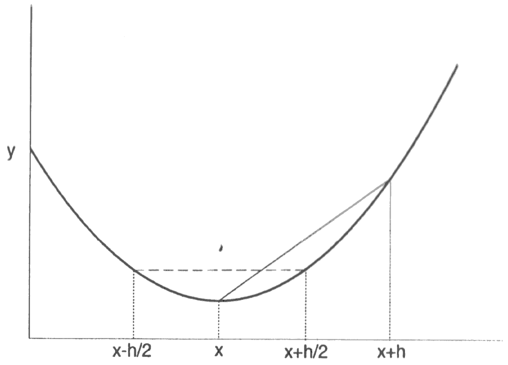

Fig. 8.1 Forward difference (solid line) and central-difference (dashed line) methods for numerical first derivative.

step forward:

$$f(x+h) = f(x) + hf'(x) + \tfrac{h^2}{2}f''(x) + \tfrac{h^3}{6}f^{(3)}(x) + \ldots, \tag{8.2}$$

where h is the *step size.* This is illustrated in Fig. 8.1. We obtain the *forward-difference* derivative algorithm by solving (8.2) for $f'(x)$:

$$f'_c(x) \simeq \frac{f(x+h)-f(x)}{h}, \tag{8.3}$$

$$\simeq f'(x) + \frac{h}{2}f''(x) + \cdots, \tag{8.4}$$

where the subscript c denotes a computed expression. You can think of this approximation as using two points to represent the function by a straight line in the interval from x to $x+h$.

The approximation (8.3) has an error proportional to h (unless the "ultra-user" looks down kindly upon you and makes f'' vanish). We can make the approximation error smaller and smaller by making h smaller and smaller. For too small an h, however, all precision will be lost through the subtractive cancellation on the LHS of (8.4), and the decreased approximation error becomes irrelevant.

For example, consider a case where $f(x)$ is a simple, analytic polynomial:

$$f(x) = a + bx^2. \tag{8.5}$$

The exact derivative is

$$f' = 2bx. \tag{8.6}$$

The computed derivative is

$$f'_c(x) \approx \frac{f(x+h)-f(x)}{h} = 2bx + bh. \tag{8.7}$$

This clearly becomes a good approximation only for small h ($h \ll 2x$).

8.2.2 Method: Central Difference

An improved approximation to the derivative starts with the basic definition (8.1). Rather than making a single step of h forward, we form a *central difference* by stepping forward by $h/2$ and backward by $h/2$:

$$\boxed{f'_c(x) \approx \frac{f(x+h/2)-f(x-h/2)}{h} \overset{\text{def}}{=} D_cf(x,h).} \tag{8.8}$$

Here D_c is the symbol for the central difference. The central-difference approximation is illustrated in Fig. 8.1.

When the Taylor series for $f(x \pm h/2)$ are substituted into (8.8), we obtain

$$f'_c(x) \simeq f'(x) + \tfrac{1}{24}h^2f^{(3)}(x) + \cdots. \tag{8.9}$$

The important difference from (8.3) is that when $f(x-h/2)$ is subtracted from $f(x+h/2)$, all terms containing an odd power of h in the Taylor series cancel. Therefore, the central-difference algorithm becomes accurate to one order higher in h; that is, h^2. If the function is well behaved; that is, if $(f^{(3)}h^2)/24 \ll (f^{(2)}h)/2$, then you can expect the error with the central-difference method to be smaller than with the forward difference (8.4).

If we now return to our polynomial example (8.5), we find that for a parabola, the central difference gives the exact answer regardless of the size of h:

$$f'_c(x) \approx \frac{f(x+h/2)-f(x-h/2)}{h} = 2bx. \tag{8.10}$$

8.2.3 Method: Extrapolated Difference

Because the Taylor series provides an analytic expression for the error, we can be even more clever. While the central difference (8.8) makes the error term proportional to h vanish, we can also make the term proportional to h^2 vanish by algebraically *extrapolating* from relatively large h (and, because of this, small roundoff error) to $h \to 0$:

$$f'_c(x) \simeq \lim_{h\to 0} D_cf(x,h). \tag{8.11}$$

We introduce the additional information by forming the central difference with step size $h/2$:

$$D_c f(x, h/2) \stackrel{\text{def}}{=} \frac{f(x+h/4) - f(x-h/4)}{h/2} \tag{8.12}$$

$$\approx f'(x) + \frac{h^2 f^{(3)}(x)}{96} + \cdots . \tag{8.13}$$

We now eliminate the quadratic error term as well as the linear error term in (8.9) with the combination

$$f'_c(x) \approx \frac{4D_c f(x,h/2) - D_c f(x,h)}{3} \tag{8.14}$$

$$\approx f'(x) - \frac{h^4 f^{(3)}(x)}{1920 \times 4 \times 16} + \cdots . \tag{8.15}$$

If $h = 0.4$, there is only one place of roundoff error, and if $f^{(3)} = O(1)$, the truncation error in (8.15) is now the same size as machine precision ϵ_m, which is the best you can hope for.

A good way of computing (8.14) is to group the terms as

$$f'_c(x) = \tfrac{1}{3h}\left\{8\left[f(x+\tfrac{h}{4}) - f(x-\tfrac{h}{4})\right] - \left[f(x+\tfrac{h}{2}) - f(x-\tfrac{h}{2})\right]\right\}. \tag{8.16}$$

The advantage to (8.16) is that it reduces the loss of precision that occurs when large and small numbers are added together, only to be subtracted from other large numbers; it is better to first subtract the large numbers from each other and then add the difference to the small numbers.

When working with these and similar higher-order methods, it is important to remember that while they may work very well for well-behaved functions, they may fail badly for computed or measured functions containing noise. In these difficult cases it may be better to first fit the data with some analytic function using the techniques of Chapter 5, *Data Fitting*, and then differentiate the fit.

But regardless of the algorithm, you may remember that evaluating the derivative of $f(x)$ at x requires you to know the values of f surrounding x. We shall use this same idea in Chapter 9, *Differential Equations and Oscillations.*

8.3 ASSESSMENT: ERROR ANALYSIS

The approximation errors in numerical differentiation decrease with decreasing step size h while roundoff errors increase with a smaller step size (you have to take more steps and do more calculations). We know from our discussion in Chapter 3, *Errors and Uncertainties in Computations*, that the best approximation occurs for an h that makes the total error $\epsilon_{\text{approx}} + \epsilon_{\text{ro}}$ a minimum.

As a rough guide, this occurs when $\epsilon_{\text{ro}} \approx \epsilon_{\text{approx}}$.

Because differentiation subtracts two numbers very close in value, the limit of roundoff error is essentially machine precision:

$$f' \approx \frac{f(x+h) - f(x)}{h} \approx \frac{\epsilon_m}{h}, \tag{8.17}$$

$$\Rightarrow \quad \epsilon_{\text{ro}} \approx \frac{\epsilon_m}{h}. \tag{8.18}$$

The approximation error with the forward-difference algorithm (8.4) is an $\mathcal{O}(h)$ term, while that with the central-difference algorithm (8.9) is an $\mathcal{O}(h^2)$ term:

$$\epsilon^{\text{fd}}_{\text{approx}} \approx \frac{f^{(2)}h}{2}, \tag{8.19}$$

$$\epsilon^{\text{cd}}_{\text{approx}} \approx \frac{f^{(3)}h^2}{24}. \tag{8.20}$$

The h value for which roundoff and approximation errors are equal is therefore

$$\frac{\epsilon_m}{h} \approx \epsilon^{\text{fd}}_{\text{approx}} = \frac{f^{(2)}h}{2}, \tag{8.21}$$

$$\frac{\epsilon_m}{h} \approx \epsilon^{\text{cd}}_{\text{approx}} = \frac{f^{(3)}h^2}{24}, \tag{8.22}$$

$$\Rightarrow \quad h^2_{\text{fd}} = \frac{2\epsilon_m}{f^{(2)}}, \qquad h^3_{\text{cd}} = \frac{24\epsilon_m}{f^{(3)}}. \tag{8.23}$$

We take $f' \approx f^{(2)} \approx f^{(3)}$ (which is crude, although not really so crude for e^x or $\cos x$), and assume a single precision calculation $\epsilon_m \approx 10^{-7}$. In this case we obtain

$$h_{\text{fd}} \approx 0.0005, \tag{8.24}$$

$$h_{\text{cd}} \approx 0.01. \tag{8.25}$$

This may seem backward because the better algorithm leads to a larger h value. It is not. The ability to use a large h means that the error in the central-difference method is some 20 times smaller than the error in the forward-difference method. For the forward-difference algorithm, approximately half of the machine precision is lost.

8.4 IMPLEMENTATION: DIFFERENTIATION, DIFF.F (.C)

8.5 ASSESSMENT: ERROR ANALYSIS, NUMERICAL

1. Differentiate the functions $\cos x$ and e^x at $x = 0.1$, $1.$, and 100 using single-precision forward-, central-, and extrapolated-difference algorithms.

   (a) Print out the derivative and its relative error $\mathcal{E}$ as a function of h. Reduce the step size h until it equals machine precision $h \approx \epsilon_m$.

   (b) Plot $\log_{10} |\mathcal{E}|$ versus $\log_{10} h$ and check whether the number of decimal places obtained agrees with the estimates in the text.

   (c) See if you can identify truncation error at large h and roundoff error at small h in your plot. Do the slopes agree with our predictions?

8.6 PROBLEM 2: SECOND DERIVATIVES

Let's say that you have measured the position versus time $x(t)$ for a particle. Your **problem** is to determine the force on the particle.

8.7 THEORY: NEWTON II

Newton's second law tells us that the force and acceleration are linearly related:

$$F = ma, \tag{8.26}$$

where F is the force, m is the particle's mass, and a is the acceleration. So if we can determine the acceleration $a(t) = d^2x/dt^2$ from the $x(t)$ table, we can determine the force.

8.8 METHOD: NUMERICAL SECOND DERIVATIVES

The concern about errors we expressed for first derivatives is even more of a concern for second derivatives because the additional subtractions lead to more cancellations. Let's go back to the central-difference method:

$$f'(x) \simeq \frac{f(x+h/2) - f(x-h/2)}{h}. \tag{8.27}$$

This algorithm gives the derivative at x by moving forward and backward from x by $h/2$. The second derivative $f^{(2)}(x)$ is taken to be the central difference of the first derivative:

$$f^{(2)}(x) \simeq \frac{f'(x+h/2) - f'(x-h/2)}{h}, \tag{8.28}$$

$$f^{(2)}(x) \simeq \frac{[f(x+h) - f(x)] - [f(x) - f(x-h)]}{h^2}. \tag{8.29}$$

As was true for first derivatives, by evaluating a function in the region surrounding x, it is possible to determine the second derivative at x. But some care is required to preserve the grouping in (8.29) and not convert it to the neater form:

$$f^{(2)}(x) \simeq \frac{f(x+h) + f(x-h) - 2f(x)}{h^2}. \tag{8.30}$$

This latter form increases subtractive cancellation because the computer first stores the "large" number $f(x+h) + f(x-h)$ [well, if f does not change sign, then the sum is larger than either $f(x+h)$ or $f(x-h)$] only to subtract another large number $2f(x)$ from it.

8.9 ASSESSMENT: NUMERICAL SECOND DERIVATIVES

Write a program to calculate the second derivative of $\cos x$ using the central-difference algorithm (8.29). Test it over four cycles. Start with $h \approx \pi/10$ and keep reducing h until you reach machine precision.