Chapter 1: Basic Numerical Analysis*

Jonathan Goodman

October 2, 1995

1 Introduction

This chapter discusses the most basic ideas in numerical analysis and explains how they are applied to validating complex numerical software and to developing efficient adaptive computational algorithms. The main idea is to manipulate asymptotic error expansions that are derived from Taylor series expansions of the functions being operated on.

We will be concerned with efficiency. This means finding the derivative accurately using a large step size, finding an integral accurately using a small number function evaluations, etc. As we said in the introduction, for casual computation such concern is probably a waste of time; more time is spent optimizing the code than could possibly be saved in a computation that takes less than a second on the computer. However, we will see that these simple operations are at the heart of algorithms that solve partial differential equations, algorithms whose running time and accuracy are of serious concern.

2 Numerical Differentiation

We start with the simplest case. Suppose we have a smooth function, $f(x)$, of a single variable, x. If we can evaluate f but not $f' = \frac{df}{dx}$, we could approximate f' be a difference quotient. Several common approximations are

$$
\begin{aligned}
(a) \qquad & f'(x) \approx \frac{f(x+h)-f(x)}{h} \\
(b) \qquad & f'(x) \approx \frac{f(x)-f(x-h)}{h} \\
(c) \qquad & f'(x) \approx \frac{f(x+h)-f(x-h)}{2h} \\
(d) \qquad & f'(x) \approx \frac{-f(x+2h)+4f(x+h)-3f(x)}{2h} \\
(e) \qquad & f'(x) \approx \frac{-f(x+2h)+4f(x+h)-4f(x-h)+f(x+2h)}{4h}
\end{aligned}
$$

*These are course notes for Computational Methods in Finance, given fall 1995 at the Courant Institute of Mathematical Sciences at New York University by Jonathan Goodman. Send email to goodman@cims.nyu.edu.

The first three have simple geometric interpretations as the slope of lines connecting nearby points on the graph of $f(x)$. The last two are more technical. Their advantages over the first three will become clear shortly.

To perform a basic error analysis, we expand $f(x+h)$ about $h=0$ using Taylor series.

$$f(x+h) = f(x) + hf'(x) + h^2\frac{f''(x)}{2} + \cdots + h^n\frac{f^{(n)}}{n!} + \cdots \quad .$$

Substituting this into (a) gives

$$\frac{f(x+h)-f(x)}{h} = f'(x) + h\frac{f''(x)}{2} + h^2\frac{f'''(x)}{6} + \cdots \quad .$$

Now we can express the error in formula (a) by

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

where

$$E_a(h) = \frac{h}{2}f''(x) + \frac{h^2}{6}f'''(x) + \cdots \quad .$$

If h is small, then h^2 is smaller still; h^2 is smaller than h bt a factor of h. Thus, $\frac{h^2}{6}f'''(x)$ will be much smaller than $\frac{h}{2}f''(x)$ unless $f'''(x)$ is much larger than $f''(x)$. When h is small enough, we may approximate the error by

$$E_a(h) \approx \frac{h}{2}f''(x) \quad .$$

This tells the story about approximation (a). The error depends roughly linearly on h (for small enough h). Cutting h in half reduces the error roughly in half. The same is true of the related approximation (b).

Taylor series analysis applied to the centered difference approximation (c) leads to

$$f'(x) = \frac{f(x+h)-f(x-h)}{2h} + E_c(h)$$

where

$$\begin{aligned} E_c(h) &= \frac{h^2}{3}f'''(x) + \frac{h^4}{12}f^{(5)}(x) + \cdots \\ &\approx \frac{h^2}{3}f'''(x) \quad \text{for } h \text{ small enough.} \end{aligned}$$

In this case, the error depends quadratically on h instead of linearly. Moreover the error from (c) is much smaller than the error from (a) or (b). That is[1],

$$\frac{h^2}{3}f'''(x) \ll \frac{h}{2}f''(x) \quad \text{i.e.} \quad E_c(h) \ll E_a(h) \quad .$$

[1]The notation $A \ll B$ is read "A is much smaller than B".

When the error is on the order of h for small h we say that the approximation is first order accurate. When it is of the order of h^2 we call it second order accurate, and similarly for third, fourth, and higher powers of h. A Taylor series analysis shows that (d) is second order accurate while (e) is fourth order. If h is small and the derivatives of f up to fifth order are not extremely large then (c) and (d) are more accurate than (a) or (b), and (e) is more accurate than any of the others. Warning: mathematicians use the phrase "order of magnitude" to refer to powers of h rather than to powers of ten. We might say that $E_a(h)$ is of the order of h even if $f''(x) = 100$. We are referring to scaling behavior, not to specific numbers.

It is important to bear in mind that a difference approximation may not achieve its expected order of accuracy if the requisite derivatives are infinite or do not exist. An example of this is the theoretical (Black Scholes) value of an American put option on a stock. The curve is linear when the price is below the early exercise point and has (as we shall see) a continuous first derivative at the early exercise point. To model this phenomenon, let $f(x)$ be the function

$$f(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ x^2 & \text{if } x \geq 0 \end{cases} .$$

If we want $f'(0)$, the formulas (c) and (e) are only first order accurate despite their higher accuracy for smoother functions. This f has a mild singularity, a discontinuity in its second derivative. Such a singularity is hard to spot on a graph. Nevertheless, it has a drastic effect on the numerical analysis of the function.

We can use finite difference approximate higher derivatives such as

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + \frac{h^2}{12} f^{(4)} + \cdots \quad ,$$

and to estimate partial derivatives of functions depending on several variables, such as

$$\frac{\partial}{\partial x} f(x,y) = \frac{f(x+h,y) - f(x-h,y)}{2h} + \frac{h^2}{3}\frac{\partial^3 f}{\partial x^3}(x,y) + \cdots \quad .$$

We can approximate sums of partial derivatives by adding the separate difference approximations. For example

$$\triangle f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \approx \frac{f(x+h,y) + f(x-h,y) + f(x,y+h) + f(x,y-h) - 4f(x,y)}{h^2} \quad .$$

It is not necessary to approximate $\triangle f$ by approximating the second partials separately in this way. Another second order accurate approximation is

$$\triangle f(x,y) \approx \frac{f(x+h,y+h) + f(x+h,y-h) + f(x-h,y+h) + f(x-h,y-h) - 4f(x,y)}{2h^2} \quad .$$

This approximation is rarely preferable to the previous, standard, one.

A multidimensional complication is that we might use different step sizes in different coordinate directions. An important example of this arises when we compute a function, $f(x,t)$ that satisfies a partial differential equation such as

$$\frac{\partial f}{\partial t} = \frac{\partial^2 f}{\partial x^2} \quad .$$

To evaluate the combination

$$\frac{\partial f}{\partial t} - \frac{\partial^2 f}{\partial x^2} \ , \tag{1}$$

we need a "time step", Δt, and a "space step", Δx. For example, we might use

$$\frac{\partial f}{\partial t} \approx \frac{f(x, t + \Delta t) - f(x,t)}{\Delta t} \tag{2}$$

and

$$\frac{\partial^2 f}{\partial x^2} \approx \frac{f(x+\Delta x, t) - 2f(x,t) + f(x - \Delta x, t)}{2\Delta x} \ . \tag{3}$$

The approximation (2) has error roughly $\Delta t \frac{\partial^2 f}{\partial t^2}/2$ which is first order in Δt. The approximation (3) has error roughly $\Delta x^2 \frac{\partial^3 f}{\partial x^3}/3$, which is second order in Δx. If we now take $\Delta x = h$ and $\Delta t = h^2$ then the principle term in the error in the approximation to (1) is is

$$h^2 \left(\frac{1}{2} \frac{\partial^2 f}{\partial t^2} + \frac{1}{3} \frac{\partial^3 f}{\partial x^3} \right) \ .$$

This example illustrates the point that the overall order of accuracy depends not only on the order of accuracy of the individual approximations, but also on the choice of step sizes in various directions.

3 Error Expansions and Richardson Extrapolation

The foregoing discussion of finite difference approximations is one instance of a common situation that we now describe more abstractly. Suppose we are trying to compute a number, A. We have an approximation to A that depends on a step size, h:

$$A \approx A(h) \ .$$

We may write $A(0)$ or A_0 for the exact answer, A. The approximation becomes increasingly accurate as h becomes small:

$$A(h) \to A \ \text{ as } \ h \to 0 \ .$$

The error, $E(h) = A(h) - A$, can be written in the form of an "asymptotic expansion" in powers of h:

$$A(h) = A + h^{p_1} A_1 + h^{p_2} A_2 + \cdots \ . \tag{4}$$

It goes without saying that $0 < p_1 < p_2 < \cdots$. In many cases, the powers and coefficients are found by Taylor series manipulations. For the approximations (a) and (b), $p_1 = 1$, $p_2 = 2$, $p_3 = 3$, and so on. For (c), $p_1 = 2$, $p_2 = 4$, $p_3 = 6$, and so on. Although (d) has the same order as (c), $p_1 = 2$ in both cases, but its asymptotic error expansion, $p_2 = 3$, $p_3 = 4$, etc. is different (work this out!). In all cases the principal error term is dominant when h is small enough. How small is small enough depends on the coefficients, A_1, A_2, etc. The leading power, p_1, determines the order of accuracy, as we have seen. This order need not be an integer. Methods with fractional order of accuracy sometimes arise, particularly in simulation

h	Error: $E(h)$	Ratio: $E(h)/E(h/2)$
.1	4.8756e-04	3.7339e+00
.05	1.3058e-04	6.4103e+00
.025	2.0370e-05	7.3018e+00
.0125	2.7898e-06	7.6717e+00
6.2500e-03	3.6364e-07	7.8407e+00
3.1250e-03	4.6379e-08	7.9215e+00
1.5625e-03	5.8547e-09	7.9611e+00
7.8125e-04	7.3542e-10	———-

Figure 1: Convergence Study for a third order accurate approximation

h	Error: $E(h)$	Ratio: $E(h)/E(h/2)$
.1	1.9041e-02	2.4014e+00
.05	7.9289e-03	1.4958e+01
.025	5.3008e-04	-1.5112e+00
.0125	-3.5075e-04	3.0145e+00
6.2500e-03	-1.1635e-04	1.9880e+01
3.1250e-03	-5.8529e-06	-8.9173e-01
1.5625e-03	6.5635e-06	2.8250e+00
7.8125e-04	2.3233e-06	———-

Figure 2: Convergence Study for a method that has no asymptotic expansion

of stochastic differential equations. Two methods may have the same order of accuracy but different asymptotic error expansions, as we have seen.

We sometimes use the "big O" notation. We say $E(h) = O(h^p)$ if there is some constant, C so that

$$|E(h)| \leq Ch^p \quad \text{whenever } |h| \leq h_0. \tag{5}$$

In this case we may say that the method is p^{th} order accurate. It may happen that a method is p^{th} order accurate in the sense (5) without having an asymptotic error expansion of the form (4). An example of this is comming.

In numerical analysis, it can be valuable to know the existence of an error expansion of the form (4) even if the coefficients, A_1, A_2, $\ldots$, cannot be determined. The two main applications are **convergence analysis** and **Richardson extrapolation**. In convergence analysis, we verify empirically the supposed order of accuracy of numbers produced by a computer code that may be large, unknown, and contain bugs. In Richardson extrapolation, we combine approximations for several values of h .....

3.1 Richardson extrapolation

Richardson extrapolation is a method that increases the order of accuracy of an approximation provided that the approximation has an asymptotic error expansion of the form (4). In its simplest form, we compute $A(h)$ and $A(2h)$ and then form a linear combination that eliminates the largest error term, $h^{p_1} A_1$. Since

$$\begin{aligned} A(2h) &= A + (2h)^{p_1} A_1 + (2h)^{p_2} A_2 + \cdots \\ &= A + 2^{p_1} h^{p_1} A_1 + 2^{p_2} h^{p_2} A_2 + \cdots \quad , \end{aligned}$$

we find that

$$\frac{2^{p_1} A(h) - A(2h)}{2^{p_1} - 1} = A + \frac{2^{p_2} - 1}{2^{p_1} - 1} h^{p_2} A_2 + \frac{2^{p_3} - 1}{2^{p_1} - 1} h^{p_3} A_3 + \cdots \quad .$$

In other words, the extrapolated approximation

$$A^{(1)}(h) = \frac{2^{p_1} A(h) - A(2h)}{2^{p_1} - 1} \tag{6}$$

has order of accuracy $p_2 > p_1$ and asymptotic error expansion

$$A^{(1)}(h) = A + h^{p_2} A_2^{(1)} + h^{p_3} A_3^{(1)} + \cdots \quad ,$$

where $A_2^{(1)} = \dfrac{2^{p_2} - 1}{2^{p_1} - 1} A_2$, and so on.

Richardson extrapolation can be repeated to remove more asymptotic error terms. For example,

$$A^{(2)}(h) = \frac{2^{p_2} A^{(1)}(h) - A^{(1)}(2h)}{2^{p_2} - 1}$$

has order p_3. Since $A^{(1)}$ depends on $A(h)$ and $A(2h)$, $A^{(2)}$ depends on $A(h)$, $A(2h)$, and $A(4h)$. It is not necessary to use powers of 2, but this is natural in many applications. Richardson extrapolation *will not work* if the underlying approximation, $A(h)$, has accuracy of order h^p in the $O(h^p)$ sense without at least one term of an asymptotic expansion.

As an example, we derive higher order difference approximations from low order ones. Start, for example, with the first order one sided approximation to $f'(x)$ given by (a). Taking $p_1 = 1$ in (6) leads to the second order approximation

$$\begin{aligned} f'(x) &\cong 2 \cdot \frac{f(x+h) - f(x)}{h} - \frac{f(x+2h) - f(x)}{2h} \\ &= \frac{-f(x+2h) + 4f(x+h) - 3f(x)}{2h} \quad , \end{aligned}$$

which is the second order three point one sided difference approximation (d). Starting with the second order centered approximation (c) (with $p_1 = 2$ and $p_2 = 4$) leads to the fourth order approximation (e).

Richardson extrapolation may also be applied to the output of a complex code. Run it with step size h and $2h$ and apply (6) to the output. This is sometimes applied to stochastic differential equations as an alternative to making up high order schemes from scratch, which can be time consuming and intricate.

When Richardson extrapolation is applied to integration using the trapezoid rule, the resulting integration method is called Romberg integration.

Rectangle	$I_k \approx hf(x_k)$	1^{st} order
Trapezoid	$I_k \approx \frac{h}{2}\left(f(x_k) + f(x_{k+1})\right)$	2^{nd} order
Midpoint	$I_k \approx hf(x_{k+1/2})$	2^{nd} order
Simpson	$I_k \approx \frac{h}{6}\Big(f(x_k) + f(x_{k+1/2}) + f(x_{k+1})\Big)$	4^{th} order
2 point GQ	$I_k \approx \frac{h}{2}\left(f(x_{k+1/2} + \frac{h}{2\sqrt{3}}) + f(x_{k+1/2} - \frac{h}{2\sqrt{3}})\right)$	4^{th} order
3 point GQ	$I_k \approx \frac{h}{2}\left(f(x_{k+1/2} + \frac{h}{2\sqrt{3}}) + f(x_{k+1/2} - \frac{h}{2\sqrt{3}})\right)$	6^{th} order

Figure 3: Common panel integration rules

4 Integration

We have a smooth function, $f(x)$, and we want to evaluate

$$I = \int_a^b f(x)dx \quad .$$

We discuss only "panel methods" even though more elegant methods such as Gauss quadrature exist. In a panel method, the interval $[a, b]$ is divided into n small panels and the approximation to the integral, I is the sum of approximations to the integral of f over the individual panels. We suppose for now that the panels are of equal size, $h = \frac{b-a}{n}$. Then we write

$$I = I_0 + \cdots + I_{n-1}$$

where

$$I_k = \int_{x_k}^{x_{k+1}} f(x)dx \quad ,$$

where $x_k = a + kh$. We can add up approximations to the I_k to get an approximation for I. A specific panel method is defined by the approximation used for I_k. Some common examples are given in the table below. There $x_{k+1/2} = a + (k + 1/2)h$ is the midpoint of the k^{th} panel.

5 Adaptive Methods

Adaptive methods are computational algorithms that attempt to fit the numerical parameters or even numerical methods to the particular problem being solved. For example, we might adjust the resolution, h, to fir the features of a function, f, being studied. We might also try to choose an appropriate method: high order for smooth functions where very accurate solutions are called for, or low order for rough functions where formal high order accuracy would be irrelevant and robustness is more important.

Adaptive algorithms are typically more complex than simple fixed ones. Moreover, they may be heuristic and seemingly sell robust than fixed algorithms. However, adaptivity is used in most state of the art codes and often makes otherwise intractable problems tractable.

Some features of adaptive algorithms are illustrated in the simple problem of choosing a step size, h, for numerical differentiation. Suppose you want to write a program that computes $f'(x)$ to within ϵ using the one sided formula, (a). We assume initially that $f(x)$ can be evaluated at any x with arbitrary accuracy and that such evaluations are the only way to get information about f. We will also suppose that f has two continuous derivatives, but we should design an algorithm that will not fail completely if f is rougher than that.

With a fixed strategy, we would specify h rather than ϵ. This has the disadvantage that the user may not know enough about f or about numerical analysis (nor have the time) to make an intelligent choice of h. Moreover, there would be no indication whether the user chose wisely; no warning bell would go off if h was too large. In this sense an adaptive method may be more robust than a fixed method, being less likely to give the wrong answer.

With slightly more generality, we consider the abstract problem of finding h so that

$$|A(h) - A| \leq \epsilon \ , \tag{7}$$

where $A(h)$ is an approximation of order p to A. We want to specify ϵ and let the computer find an h that satisfies (8). Not knowing A, we can never be completely sure we have succeeded. It makes sense to ask the user for a rough initial guess for h since we may not know the order of magnitude of a "small" perturbation. For some applications, a million dollar perturbation might be considered small while in others (my family budget) it would be on the large side.

The simplest strategy would be as follows:

```
                    start, have A, h0, ε.
                    compute A1 = A(h0).
                    set h = h0.
refinement loop:    h = h/2 (compute the refined approximation)
                    A2 = A(h)
      (*)           if |A2 − A1| ≤ ε, (stop if the h and h/2 approximations agree.)
                        return A2 and stop.
                    otherwise, (replace h be h/2 and try again.)
                       set A1 = A2,
                       test for an infinite loop and possibly report failure
                       go to refinement loop
```

This algorithm repeatedly cuts h in half until $A(h)$ and $A(h/2)$ differ by at most ϵ. Th then returns $A(h/2)$, which is probably the more accurate approximation. To be robust, it should have some protection against falling into an infinite loop. For example, if h is less than roundoff error ($|h/x| < 10^{-7}$ for single precision if you need to distinguish $x + h$ from x) you cannot get a meaningful answer.

There are more sophisticated convergence criteria that use the order of accuracy of $A(h)$. From the asymptotic expansion (4), you can solve for the main error term from $A(h)$ and $A(h/2)$. This is more or less the same as Richardson extrapolation. On the one hand we have

$$A(h) - A(h/2) \cong \left(1 - \frac{1}{2^p}\right) h^p A_1 \ ,$$

and on the other hand,

$$E(h) = A(h) - A \cong h^p A_1 \quad .$$

Therefore, we can form an error estimate, $\overline{E}(h)$, by

$$E(h) \cong \overline{E}(h) = \frac{A(h) - A(h/2)}{1 - \frac{1}{2^p}} \quad .$$

For example, for a first order method, we have the error estimate

$$\overline{E}(h) = 2\left(A(h) - A(h/2)\right) \quad .$$

This suggests replacing the stopping test (*) above by $\left|\overline{E}(h)\right| \leq \epsilon$. It also suggests that we should return $A(h) + \overline{E}(h)$, which is the Richardson extrapolated estimate of the answer. For sufficiently small h, this should be much more accurate than either $A(h)$ or $A(h/2)$.

Values of h small enough for the asymptotic expansion (4) to apply are said to be in the "asymptotic range". We want to protect ourselves from false conclusions that could come from applying (4) outside that range. One possible way to do this is to compare the error estimates $\overline{E}(h)$ and $\overline{E}(2h)$. These numbers should be fairly close in the asymptotic range but there is no particular reason for them to be close otherwise. We might consider a strategy that reports the extrapolated estimate $A(h) + \overline{E}(h)$ if $\overline{E}(h)$ and $\overline{E}(2h)$ agree but just reports $A(h)$ otherwise.

This discussion brings out a paradox about adaptive methods that rely on error estimates. *There is no such thing as an accurate error estimate.* If you had an accurate error estimate, you would surely add it in in order to report an answer as accurate as possible. There seems to be little alternative to the very conservative strategy of using $\overline{E}(h)$ both in a termination criterion such as (*) and as a correction to the answer. If you want an error that is reliably less than ϵ, you will probably have an error much less than ϵ most of the time.

These adaptive strategies can be used in more complex problems such as integration, interpolation, and the solution of partial differential equations. However, in all these problems there is the possibility of more sophisticated adaptive algorithms that locally adjust the level of resolution in different regions of space. This is because the function being integrated of approximated may have features on several length scales, come large and some much smaller. In such cases, a uniform mesh with a resolution necessary for the smallest features would be unnecessarily fine over most of the computational domain. Consider, for example, the problem of integrating the function $f(x) = e^{-x^2/2} + e^{-20x^2}$. For most of space, a panel size of $\Delta x = .1$ would give reasonable accuracy. However, near the origin, a much finer mesh is called for. Methods that apply a non uniform level of resolution are called "local adaptive" methods.

A key ingredient in a local adaptive method is a local error estimate. For example, in panel method integration, we might estimate the integration error in each panel. Panels in which the error is too large could be refined while panels with small error could be left alone. This would result in panels of various sizes, possibly a wide range. Again we have the paradox that there is no accurate error estimate; we would use the estimate of the integration error over a panel (the local error estimate) both in a local refinement criterion and as a correction.

In adaptive integration there is an additional complication. The global error is the sum of local errors over the "active" (i.e. as yet unrefined) panels of various sizes. Given error estimates for each of these active panels, we need to select the worst ones for refinement. A commonly used hueristic principle used to guide this choice is the equidistribution principle: the error should be roughly equidistributed among the panels. Panels for which the local error is extremely small have been unnecessarily refined. It would be a waste of computer resources to apply the equidistribution principle too rigidly, it is, after all, only a heuristic.

An adaptive integration strategy could use a binary branching tree data structure. Each coarsest level panel would be the root of a tree. Each time a panel is refined, we would add the two sub panels as leaves attached to the panes being refined. The leaves of the trees are the active panels some of which need further refinement. We might choose, for example, to refine the active panel with the largest local error estimate until the sum of the local error estimates is less than ϵ.

It should be clear from this that adaptive methods often require more complex programming and more understanding of computer science than fixed algorithms. For that reason, people trained in mathematics or the sciences have not used them as aggressively as they should. This gives people who are able to program adaptive methods an enormous and cheap competitive advantage over those who cannot or will not.