Scientific Computing
Chapter II
Basic Numerical Analysis

Jonathan Goodman
Courant Institute of Mathemaical Sciences

Last revised February 11, 2002

1 Introduction

The most basic technique in numerical analysis is manipulation of Taylor series expansions. This is how we find numerical methods for differentiation and integration, for solving differential equations, for optimization, and for many other purposes. The same kind of analysis leads to quantitative understanding of the error in many numerical approximations. These "asymptotic error expansions" are used for validating complex numerical software and for developing efficient adaptive computational algorithms.

This chapter develops the Taylor series method for three specific problems, differentiation, integration, and interpolation. Numerical differentiation is the problem of finding (as accurately as necessary) a derivative of a function, given several values of the function itself. Numerical integration is the problem of computing the integral. Interpolation is the problem of evaluating a function at some value of its arguments given as data function values at other values of the arguments.

Each of these topics is treated from the point view of asymptotic error expansions. For now, this is just a different way of using Taylor series. For example, one Taylor series expansion is

$$e^x = 1 + x + \frac{1}{2}x^2 + \cdots + \frac{1}{n!}x^n + \cdots \quad .$$

One could use this formula to compute e^2 by taking $x = 2$ and adding up enough terms to get the desired accuracy. This is not what we do here. Instead, we fix the number of terms, for example

$$e^x \approx 1 + x + \frac{1}{2}x^2 \ ,$$

and ask about the range of x values for which this approximation is accurate enough. Read Chapter 5 to see how to get all the way to $x = 2$ this way. In

real problems, it is common that we know one or two terms of a Taylor series but not the whole series. More often, the fact that the series exists is all that we need to know.

Errors that arise from taking just a few terms of an infinite Taylor series are called "truncation errors"; we truncate a series. This chapter is about truncation error. Usually truncation error is so much larger than error from inaccurate computer arithmetic, roundoff error, that we neglect roundoff and suppose that all arithmetic is done exactly. We will have to revisit this assumption if our problem is ill posed, if our computational algorithm is unstable, or if the step size is too small.

There is a tradeoff between accuracy and running time on one hand, and program complexity on the other. The considerations used to explain the simplest methods such as rectangle rule integration can be extended to find more sophisticated and accurate methods. For casual computing, this is probably a waste of time. More time is spent optimizing the code than could possibly be saved in a computation that takes the computer less than a second. However, the simple operations discussed here are at teh hearts of computational algorithms whose running times are a serious concern.

The approximations studied here all involve a "step size", h. They become more accurate as h becomes smaller. The rate of improvement is the "order of accuracy". More accurate approximations generally lead to facter computations. For example, in estimating $\int_a^b f(x)dx$, the region of integration, $[a, b]$ is divided into a number of "panels" of size h. The larger h, the more panels and longer running time. If a sophisticated integration method achieves 99% accuracy with $h = .1$ while the simple one requires $h = .01$, the sophisticated method will be, maybe, five times faster – twoce the cost per panel but ten times fewer panels. Whether this factor of five speedup is worth days of extra analysis and programming depends on the situation.

We will often use the phrase "for sufficiently small h" without clarifying how small is small enough. One answer might be that h should be small compared to "natural length scales" in the problem. This issue is too problem dependent to settle in a simple general way. When h is small enough, we say that "h is in the asymptotic range", the range in which an asymptotic expansion gives useful information. It is often clear from the problem roughly how large the asymptotic range is likely to be, and that range is usually large enough to cover practical h values. If a code is unable to produce results consistent with an asymptotic error analysis, it is generally not because the asymptotic range is inaccessible. Look instead for a bug in the code, in the method, or in the error analysis.

2 Taylor series and asymptotic expansions

Taylor series may be thought of in two ways, as a "convergent" formula for a function value, or as an "asymptotic expansion", a sequence of approximations of increasing "order of accuracy". The distinction is explained here. The asymp-

totic expansion viewpoint is central to much elementary numerical analysis. the error in approximating a function by a few terms of its Taylor series is called "truncation error". This is the primary source of error in numerical methods for ordinary or partial differential equations.

Suppose we have a function, $f(x)$, and a "step size", h. The Taylor series for f about x is

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \cdots + \frac{h^n}{n!}f^{(n)}(x) + \cdots \quad . \tag{1}$$

The series (1) "converges" if

$$f(x) + \cdots + \frac{h^n}{n!}f^{(n)}(x) \to f(x+h) \quad \text{as } n \to \infty. \tag{2}$$

Generally speaking, if you can write a formula for $f(x)$ then the expansion will converge for sufficiently small h. The formula (2) suggests that we improve the accuracy of the Taylor series approximation by taking more terms.

The accuracy also increases if we make h smaller; it will become clear later why we are allowed to do this. One way to understand the accuracy of finite Taylor series approximations is to use the "remainder theorem", which is the formula:

$$f(x+h) - \left(f(x) + + \cdots + \frac{h^n}{n!}f^{(n)}(x) \right) = \frac{h^{(n+1)}}{(n+1)!}f^{(n+1)}(\xi) \quad . \tag{3}$$

This ξ depends on x, h, and n in an unknown way, except that ξ is between x and $x+h$. If h is negative, then $x+h$ is to the left of x, but still ξ is between. If we fix a maximum step size, h_0, and set

$$M_{n+1} = \max_{|h| \le h_0} \left| f^{(n+1)}(x+h) \right| / (n+1)! \ , \tag{4}$$

then (3) gives us a bound on how large the Taylor series error can be. This bound is

$$\left| f(x+h) - \left(f(x) + \cdots + \frac{h^{(n-1)}}{(n-1)!}f^{(n-1)}(x) \right) \right| \le M_n \left| h^n \right| \quad . \tag{5}$$

If $M_n h^n \to 0$ as $n \to \infty$, then the series converges. If, for some particular n, M_n is well defined and finite, then (5) holds for that n. In this case the error from an n term Taylor series approximation is bounded by some multiple of h^n.

This error bound is expressed using "big O" notation. Suppose that $E(h)$ is the error in some approximation. We say the approximation is "p^{th} order accurate" and write

$$E(h) = O(h^p) \ , \tag{6}$$

if there is a constant, M_p, and an $h_0 > 0$ so that

$$|E(h)| \le M_p \left| h^p \right| \quad \text{when } |h| \le h_0. \tag{7}$$

We call M_p in (7) the "implied constant" for (6). If M_p were very large, the error bound (7) or (6) would have little practical value, a situation that does arise sometimes.

The "big O" notation may be used in a slightly different way. The notation $O(h^p)$ is meant to represent some function of h that satisfies (7). For example, we write

$$f(x+h) - f(x) - hf'(x) = O(h^2) \ ,$$

or, which means the same thing,

$$f(x+h) = f(x) + hf'(x) + O(h^2) \ .$$

When discussing basic operations and order of accuracy, we usually think of approximations with a higher order of accuracy as actually being more accurate. After all, h^2 is smaller than h by a factor of h. We sometimes refer to powers of h as orders of magnitude, in the way scientists think of powers of 10. For a particular h, a third order approximation may have more error than a second order one. But, given the two approximations, if h is small enough, the higher order one will be more accurate.

When we view a series in powers of h as an asymptotic expansion, we write (using g for the function being approximated)

$$g(h) \sim g_0 + g_1 h + g_2 h^2 + \cdots \ . \tag{8}$$

The $\sim$ rather than $=$ in (8) indicates that we do not necessarily expect the series to converge. In fact, for any particular h there may be a limit to how accurately the left side approximates $g(h)$, no matter how many terms are used. The notation (8) just a shorthand for all the relations

$$\left.\begin{array}{l} g(h) = g_0 + O(h) \\ g(h) = g_0 + g_1 h + O(h^2) \\ \cdots \qquad \cdots \\ g(h) = g_0 + g_1 h + \cdots + g_{n-1} h^{n-1} + O(h^n) \end{array}\right\} \tag{9}$$

Altogether, these mean that taking the first p terms gives a p^{th} order approximation to g. In this chapter and the next, it is the asymptotic approximation properties (9) that matter, not whether the series converges.

If an asymptotic expansion does not converge, it must be because the coefficients g_n grow too rapidly with n. Otherwise, the ratio test from calculus will show that the series has some radius of convergence. An example of this is given by the function

$$g(h) = \int_0^{1/2} \frac{1}{1-t} e^{-t/h} dt \ . \tag{10}$$

The exponential decays very quickly when h is small, so that only values of t close to zero contribute much to the integral. This suggests that a Taylor series of $1/(1-t)$ about $t = 0$ will be useful. This expansion is

$$\frac{1}{1-t} = 1 + t + t^2 + \cdots \ .$$

If you do not remember this Taylor series, you can think of the right side as a geometric series and the left side as the sum, which it is. Integrating this term by term gives

$$\int_0^{1/2} 1e^{-t/h}dt \approx h \ ,$$

and

$$\int_0^{1/2} te^{-t/h}dt \approx h^2 \ ,$$

and so on to the general term

$$\int_0^{1/2} t^n e^{-t/h}dt \approx n!\, h^{n+1} \ ,$$

This gives the asymptotic exapnsion

$$g(h) \sim h + h^2 + 2h^3 + \cdots + n!\, h^{n+1} + \cdots \ .$$

In this example, the coefficients $g_n = (n-1)!$ grow so rapidly that the series does not converge for any positive value of h. However, it is not hard (if you are good at advanced calculus) to show that the relations (9) are all satisfied.

3 Numerical Differentiation

One basic numerical task is estimating the derivative of a function from given function values. 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' by a difference quotient. Several common approximations are

$$\left.\begin{array}{rcll} f'(x) & \approx & \dfrac{f(x+h)-f(x)}{h} & (a) \\ f'(x) & \approx & \dfrac{f(x)-f(x-h)}{h} & (b) \\ f'(x) & \approx & \dfrac{f(x+h)-f(x-h)}{2h} & (c) \\ f'(x) & \approx & \dfrac{-f(x+2h)+4f(x+h)-3f(x)}{2h} & (d) \\ f'(x) & \approx & \dfrac{-f(x+2h)+8f(x+h)-8f(x-h)+f(x+2h)}{12h} & (e) \end{array}\right\} \tag{11}$$

The first three have simple geometric interpretations as the slope of lines connecting nearby points on the graph of $f(x)$. A carefully drawn figure shows that (11c) is more accurate than (11a). We give an analytical explanation of this below. The last two are more technical.

The advantage of the last one over the others is its greater accuracy. If h is small enough, the error in (11e) is likely to be much less than the error in the others. This often (but not always) has the consequence that (11e) achieves a given level of accuracy (say 1%) for a larger h value, a property that leads to greater efficiency when solving differential equations.

To perform a basic error analysis, we use the Taylor series expansion (1). Substituting this into (11a) 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 (11a) by:

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

where

$$E_a(h) = \frac{h}{2}f''(x) + \frac{h^2}{6}f'''(x) + \cdots \quad . \tag{12}$$

If h is small, then h^2 is smaller still; h^2 is smaller than h by a factor of h. Thus, $\frac{h^2}{6}f'''(x)$ will usually be much smaller than $\frac{h}{2}f''(x)$.

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 (11a). 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 (11b). The approximations (11a) and (11b) are called "first order one sided difference approximations" to the derivative.

Taylor series analysis applied to the centered difference approximation (11c) leads to

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

where

$$\begin{aligned} E_c(h) &= \frac{h^2}{6}f'''(x) + \frac{h^4}{24}f^{(5)}(x) + \cdots \\ &\approx \frac{h^2}{6}f'''(x) \quad \text{for } h \text{ small enough.} \end{aligned} \tag{13}$$

In this case, the error depends quadratically on h instead of linearly. Moreover the error from (11c) is much smaller than the error from (11a) or (11b). That is[1],

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

[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 (11d) is second order accurate while (11e) is fourth order. If h is small and the derivatives of f up to fifth order are not extremely large then (11c) and (11d) are more accurate than (11a) or (11b), and (11e) is more accurate than any of the others. The approximation (11c) is called the "second order three point centered difference approximation" to the derivative; (11d) is the "three point one sided second order" difference approximation; (11e) is the "five point centered fourth order" approximation.

A difference approximation may not achieve its expected order of accuracy if the requisite derivatives are infinite or do not exist. As an example of this, 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 (1c) and (1e) 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, but may have a drastic effect on the numerical analysis of the function.

We can use finite differences to approximate higher derivatives such as

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

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

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

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

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

This is called "the standard 5 point discrete Laplacian". It is not necessary to approximate $\triangle f$ by approximating the second partials separately in this way. Another second order accurate approximation is

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

This "rotated 5 point" approximation is rarely preferable to the 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

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

To evaluate the combination

$$\frac{\partial f}{\partial t} - \frac{\partial^2 f}{\partial x^2} \quad , \tag{14}$$

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

$$\frac{\partial f}{\partial t} \approx \frac{f(x, t + \Delta t) - f(x, t)}{\Delta t} \tag{15}$$

and

$$\frac{\partial^2 f}{\partial x^2} \approx \frac{f(x + \Delta x, t) - 2f(x, t) + f(x - \Delta x, t)}{2\Delta x} \quad . \tag{16}$$

The approximation (15) has error roughly $\Delta t \frac{\partial^2 f}{\partial t^2}/2$ which is first order in Δt. The approximation (16) has error roughly $\Delta x^2 \frac{\partial^4 f}{\partial x^4}/6$, which is second order in Δx. If we now take $\Delta x = h$ and $\Delta t = h^2$ then the principal term in the error in the approximation to (14) is is

$$h^2 \left(\frac{1}{2} \frac{\partial^2 f}{\partial t^2} + \frac{1}{6} \frac{\partial^4 f}{\partial x^4} \right) \quad .$$

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.

4 Error Expansions and Richardson Extrapolation

The error expansions (12) and (14) above are instances of a common situation that we now describe more systematically and abstractly.Suppose we are trying to compute a number, A, which is the "Answer" to some computational problem. We have an approximation to A that depends on a step size, h:

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

The approximation becomes increasingly accurate as h becomes small:

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

The error is $E(h) = A(h) - A$.

An "asymptotic error expansion" in powers of h can be expressed either by

$$A(h) \sim A + h^{p_1} A_1 + h^{p_2} A_2 + \cdots \quad , \tag{17}$$

or, equivalently, by

$$E(h) \sim h^{p_1} A_1 + h^{p_2} A_2 + \cdots \quad .$$

The expression (17) does not imply that the series on the right converges to $A(h)$, but rather it is a substitute for the statements that make it an asymptotic series:

$$\left.\begin{array}{lll} \dfrac{A(h)-(A+h^{p_1}A_1)}{h^{p_1}} \to 0 & \text{as } h \to 0, & (a) \\ \dfrac{A(h)-(A+h^{p_1}A_1+h^{p_2}A_2)}{h^{p_2}} \to 0 & \text{as } h \to 0, & (b) \\ \text{and so on.} & \vdots & \end{array}\right\} \qquad (18)$$

The statement (18a) says not only that $A+h^{p_1}$ is a good approximation to $A(h)$, but that the error in the approximation is smaller than h^{p_1} for small enough h. The statement (18b) says that $A + h^{p_1}A_1 + h^{p_2}A_2$ is a better approximation to $A(h)$ in that the error is smaller than h^{p_2}, which is much smaller than h^{p_1} for small h. 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 (11a) and (11b), $p_1 = 1$, $p_2 = 2$, $p_3 = 3$, and so on. For (11c), $p_1 = 2$, $p_2 = 4$, $p_3 = 6$, and so on. Although (11d) has the same order as (11c), $p_1 = 2$ in both cases, 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 of stochastic differential equations. Two methods may have the same order of accuracy but different asymptotic error expansions, as we have seen.

It is possible that an approximation is p^{th} order accurate in the "big O" sense, $E(h) = O(h^p)$, without having an asymptotic error expansion of the form (17). An example of this is given below.

It can be valuable to know the *existence* of an error expansion of the form (17) even if the coefficients, A_1, A_2, ..., 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 to produce a new approximation that has greater order of accuracy than $A(h)$.

4.1 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 (17). In its simplest form, we compute $A(h)$ and $A(2h)$ and then form a linear combination that eliminates the largest error term, $h^{p_1}A_1$.

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

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

$$A^{(1)}(h) = \frac{2^{p_1} A(h) - A(2h)}{2^{p_1} - 1} \tag{19}$$

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 of Richardson extrapolation, 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 (11a). Taking $p_1 = 1$ in (19) 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 (11d). Starting with the second order centered approximation (11c) (with $p_1 = 2$ and $p_2 = 4$) leads to the fourth order approximation (11e).

Richardson extrapolation may also be applied to the output of a complex code. Run it with step size h and $2h$ and apply (19) 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.

4.2 Convergence analysis

It is important to find ways to test whether a code and the algorithm it is based on are correct. A very useful test, "convergence analysis", is based on the asymptotic error expansion (17). It tests not only whether the error is going to zero as $h \to 0$, but whether it does so as predicted by theory. Suppose, for example, we program the formula (11e) but with 3 instead of 4. The resulting approximation will converge to $f'(x)$ as $h \to 0$ but with the wrong order fo accuracy. A convergence analysis would uncover this immediately.

There are two cases, the case where the exact answer is known and the case where it is not known. While we probably would not write a code for a problem to which we know the answer, it is often possible to apply a code to a problem with a known answer for debugging. In fact, a code should be written modularly so that it is easy to apply it to a range or problems broad enough to include a nontrivial problem[2] with a known answer.

If A is known, we can run the code with step size h and $2h$ and, from the resulting approximations, $A(h)$ and $A(2h)$, compute

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

For small h the first term is a good enough approximation so that the ratio should be approximately the characteristic value

$$R(h) = \frac{E(2h)}{E(h)} \cong 2^{p_1}$$

Figure 1 is a computational illustration of this phenomenon. Figure 2 shows what may happen when we apply this convergence analysis to an approximation that is second order accurate in the big O sense without having an asymptotic error expansion. The error gets very small but the error ratio does not have simple behavior as in Figure 1.

Convergence analysis can be applied even when A is not known. In this case we need three approximations, $A(4h)$, $A(2h)$, and $A(h)$. Again assuming the existence of an asymptotic error expansion (11), we get, for small h,

$$R'(h) = \frac{A(4h) - A(2h)}{A(2h) - A(h)} \approx 2^{p_1} \quad .$$

5 Integration

Here the Answer we want is the integral

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

[2] A trivial problem is one that is too simple to test the code fully. For example, if you compute the derivative of a linear function, any of the formulae (11a) – (11e) would give the exact answer. There would be no truncation error for the convergence analysis to measure. The fourth order approximation (11e) gives the exact answer for any polynomial of degree less than five.

Rectangle	$I_k \approx h_k f(x_k)$	1^{st} order
Trapezoid	$I_k \approx \frac{h_k}{2}\left(f(x_k) + f(x_{k+1})\right)$	2^{nd} order
Midpoint	$I_k \approx h_k f(x_{k+1/2})$	2^{nd} order
Simpson	$I_k \approx \frac{h_k}{6}\left(f(x_k) + 4f(x_{k+1/2}) + f(x_{k+1})\right)$	4^{th} order
2 point GQ	$I_k \approx \frac{h_k}{2}\left(f(x_{k+1/2} - \frac{h_k}{2\sqrt{3}}) + f(x_{k+1/2} + \frac{h_k}{2\sqrt{3}})\right)$	4^{th} order
3 point GQ	$I_k \approx \frac{h_k}{18}\left(5f(x_{k+1/2} - \frac{h_k}{2\sqrt{\frac{3}{5}}}) + 8f(x_{k+1/2}) + 5f(x_{k+1/2} + \frac{h_k}{2\sqrt{\frac{3}{5}}})\right)$	6^{th} order

Figure 3: Common panel integration rules

We discuss only "panel methods" here. Other elegant methods such as Gauss quadrature are discussed in Dahlquist and Bjork and other places. In a panel method, the integration interval, $[a, b]$, is divided into n subintervals, or panels, $P_k = [x_k, x_{k+1}]$, where $a = x_0 < x_1 < \cdots < x_n = b$. If the panel P_k is small, we can get an accurate approximation to

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

using a few evaluations of f inside P_k. Adding these approximations over k given an approximation to the integral I. Some of the more common panel integral approximations are given in Figure 3. There $x_{k+1/2} = (x_{k+1} + x_k)/2$ is the midpoint of the panel and $h_k = x_{k+1} - x_k$ is the width.

We begin our error analysis of panel methods assuming that all the panels are the same size

$$h = \Delta x = |P_k| = x_{k+1} - x_k \text{ for all } k.$$

Given this restriction, not every value of h is allowed because $b - a = nh$ and n, the number of panels, is an integer. When we take $h \to 0$, we will assume that h only takes allowed values $h = (b - a)/n$.

The overall integration error is the sum of the integration errors for the individual panels. To focus on a particular panel, suppose that P is a generic interval of length h, that is $P = [x_*, x_* + h]$. For the rectangle rule, we approximate

$$I_P = \int_P f(x)dx = \int_{x_*}^{x_*+h} f(x)dx$$

with the approximate integral, $I_P(h)$, defined by

$$I_P(h) = hf(x_*) \ .$$

To estimate the difference between I_P and $I_P(h)$, we expand f in a Taylor series about x_*:

$$f(x) \approx f(x_*) + (x - x_*)f'(x_*) + \frac{(x - x_*)^2}{2}f''(x_*) + \cdots \ .$$

Integrating this term by term leads to

$$\begin{aligned} I_P(x_*) &= \int_P f(x_*)dx + \int_P (x - x_*)f'(x_*)dx + \cdots \\ &= hf(x_*) + \frac{h^2}{2}f'(x_*) + \frac{h^3}{6}f''(x_*) + \cdots \quad . \end{aligned}$$

The error in integration over this panel then is

$$E(P,h) = I_P(h) - I_P \approx -\frac{h^2}{2}f'(x_*) - \frac{h^3}{6}f''(x_*) - \cdots \quad . \tag{20}$$

From this we see that the error in integration over any particular panel, which is called "local truncation error", is on the order of h^2. To get the "global error", we need to add up the local truncation errors from all the panels. Since there are $n = O(1/h)$ panels, the global truncation error might be of the order $n \cdot h^2 = O(h)$. That is, the global accuracy could be one order lower than the local truncation error.

To see this more precisely and construct the asymptotic error expansion, we sum the local for the panels over all panels to get the total error

$$E_{\rm tot} = \sum_{n=0}^{n-1} E(P_k,h) \approx -\sum_{n=0}^{n-1} \frac{h^2}{2}f'(x_k) - \sum_{n=0}^{n-1} \frac{h^3}{6}f''(x_k) - \cdots \quad . \tag{21}$$

The first term on the right is generally the largest for small h. We can understand it as follows. Since

$$h\sum_{n=0}^{n-1} f(x_k) = \int_a^b f(x)dx + O(h) \quad ,$$

we have

$$h\sum_{n=0}^{n-1} f'(x_k) = \int_a^b f'(x)dx + O(h) = f(b) - f(a) + O(h) \quad .$$

With this approximation, we see that

$$E_{\rm tot} \approx -\frac{f}{2}\left(f(b) - f(a)\right) + O(h^2) \quad . \tag{22}$$

This gives the first term in the asymptotic error expansion. To get the next term, apply (21) to the error itself, i.e.

$$\begin{aligned} h\sum_{n=0}^{n-1} f'(x_k) &= \int_a^b f'(x)dx - \frac{h}{2}\left(f''(b) - f''(a)\right) + O(h^2) \\ &= f(b) - f(a) - \frac{h}{2}\left(f''(b) - f''(a)\right) + O(h^2) \quad . \end{aligned}$$

n	Computed Integral	Error	Error/h	$(E - A_1 h)/h^2$	$(E - A_1 h - A_2 h^2)/h^3$
10	3.2271	-0.2546	-1.6973	0.2900	-0.7250
20	3.3528	-0.1289	-1.7191	0.2901	-0.3626
40	3.4168	-0.0649	-1.7300	0.2901	-0.1813
80	3.4492	-0.0325	-1.7354	0.2901	-0.0907
160	3.4654	-0.0163	-1.7381	0.2901	-0.0453

Figure 4: Computational experiment illustrating the asymptotic error expansion for rectangle rule integration

In the same way, we find that

$$\frac{h^3}{6} \sum_{n=0}^{n-1} f''(x_k) = \frac{h^2}{6} \left(f'(b) - f'(a) \right) + O(h^3) \quad .$$

Combing all these gives the first two terms in the total error expansion

$$E_{\text{tot}} \approx -\frac{h}{2} \left(f(b) - f(a) \right) + \frac{h^2}{12} \left(f'(b) - f'(a) \right) + \cdots \quad . \tag{23}$$

It is clear that this procedure can be used to continue the expansion as far as we want, but you would have to be very determined to compute, for example, A_4,the coefficient of h^4. An elegant and much more systematic discussion of this error expansion is carried out in the book of Dahlquist and Bjork. The resulting error expansion is called the Euler McLaurin formula. The coefficients $1/2$, $1/12$, and so on, are related to the so called Bernoulli numbers.

The error expansion (23) will not be valid if the integrand, f, has singularities inside the domain of integration. Suppose, for example, that $f(x) = 0$ for $x \leq 1/\sqrt{2}$ and $f(x) = \sqrt{x - 1/\sqrt{2}}$ for $x \geq 0$. In this case the error expansion for the rectangle rule approximation to $\int_0^1 f(x)dx$ has one valid term only. This is illustrated in Figure 5. The "Error/h" column shows that the first coefficient, A_1, exists. Moreover, A_1 is given by the formula (23). The numbers is the last column do not tend to a limit. This shows that the coefficient A_2 does not exist. The error expansion does not exist beyond the first term.

The analysis of the higher order integration methods listed in Table 1 will be easier if we use a more symmetric basic panel. From now on, the panel of length h will have x_* in the center, rather at the left end, that is

$$P = [x_* - h/2, x_* + h/2] \quad .$$

If we now expand $f(x)$ in a Taylor series about x_* and integrate term by term, we get

$$\int_P f(x)dx = \int_{x=x_*-\frac{h}{2}}^{x_*+\frac{h}{2}} f(x)dx \approx hf(x_*) + \frac{f''(x_*)}{16}h^3 + \frac{f^{(4)}(x_*)}{384}h^5 + \cdots \quad .$$

n	Computed Integral	Error	Error/h	$(E - A_1 h)/h^2$
10	7.4398e-02	-3.1277e-02	-3.1277e-01	-4.2173e-01
20	9.1097e-02	-1.4578e-02	-2.9156e-01	-4.1926e-01
40	9.8844e-02	-6.8314e-03	-2.7326e-01	-1.0635e-01
80	1.0241e-01	-3.2605e-03	-2.6084e-01	7.8070e-01
160	1.0393e-01	-1.7446e-03	-2.7914e-01	-1.3670e+00
320	1.0482e-01	-8.5085e-04	-2.7227e-01	-5.3609e-01
640	1.0526e-01	-4.1805e-04	-2.6755e-01	1.9508e+00
1280	1.0546e-01	-2.1442e-04	-2.7446e-01	-4.9470e+00
2560	1.0557e-01	-1.0631e-04	-2.7214e-01	-3.9497e+00
5120	1.0562e-01	-5.2795e-05	-2.7031e-01	1.4700e+00

Figure 5: Computational experiment illustrating the breakdown of the asymptotic expansion.

For the midpoint rule, this leads to a global error expansion in even powers of h, $E \approx A_1 h^2 + A_2 h^4 + \cdots$, with $A_1 = (f'(b) - f'(a))/16$. Each of the remaining panel methods is symmetric about the center of the panel. This implies that each of them has local truncation error containing only odd powers of h and global error, E_{tot}, containing only even powers of h.

For the remaining methods, we will not compute the coefficients in the error expansion, but only the leading power of h, the order of accuracy. This can be determined by a simple observation: the order of the local truncation error is one more than the degree of the lowest monomial that is not integrated exactly by the panel method. For example, the rectangle rule integrates $f(x) = x^0 \equiv 1$ exactly but gets $f(x) = x^1 \equiv x$ wrong. The order of the lowest monomial not integrated exactly is 1 so the local truncation error is of the order of h^2. The midpoint rule integrates x^0 and x^1 correctly but gets x^2 wrong. The order of the lowest monomial not integrated exactly is 2 so the local truncation error is of the order of h^3. If the generic panel has x_* in the center, then

$$\int_P (x - x_*)^n \, dx$$

is always done exactly if n is odd. This is because both the exact integral and its panel method approximation are zero by symmetry. The exact integral because the panel is symmetric about x_* and the discrete approximation to it because the evaluation points and weights are also symmetric about x_*. This is not true of the rectangle rule.

To understand why this rule works, think of the Taylor expansion of a general function, $f(x)$ about the midpoint, x_*. This is the same as writing f as the sum of a series of monomials. Applying the panel integral approximation to f is the same as applying the approximation to each monomial and summing the results. Moreover, the integral of a monomial $(x - x_*)^n$ over P is proportional to h^{n+1}, as is the panel method approximation to it, regardless of whether the panel method is exact or not. The first monomial that is not integrated exactly

contributes something proportional to h^{n+1} to the error.

Using this rule it is easy to determine the accuracy of the approximations in Table 1. The trapezoid rule integrates constants and linear functions exactly, but it gets quadratics wrong. This makes the local truncation error third order and the global error second order. The Simpson's rule coefficients $1/6$ and $2/3$ are designed exactly to integrate constants and quadratics exactly, which they do. Simpson's rule integrates cubics exactly (by symmetry) but quartics are gotten wrong. This makes Simpson's rule have fourth order global accuracy. The two point Gauss quadrature also does constants and quadratics correctly but quartics wrong (check this!). The three point Gauss quadrature rule does constants, quadratics, and quartics correctly but gets $(x - x_*)^6$ wrong. That makes it sixth order accurate. The theory of Gauss quadrature is discussed by Dahlquist and Bjork. I think it is the most beautiful part of numerical analysis.

6 The method of undetermined coefficients

The method of undetermined coefficients is a general way to find approximation formulae of a desired type. Suppose we want to estimate some A in terms of given data $g_1(h)$, $g_2(h)$, ... The method is simply to assume a linear estimation formula of the form

$$A \approx A(h) = a_1(h)g_1(h) + a_2(h)g_2(h) + \cdots \ , \tag{24}$$

then determine the unknown coefficients by matching Taylor series up to the highest possible order. The coefficients will probably take the form of a constant times some power of h: $a_k(h) = a_k h^{p_k}$. The algebra will be simpler if we guess or figure out the powers sooner. The estimator is "consistent" if $A(h) - A \to 0$ as $h \to \infty$. Generally, being consistent is the same as being at least first order accurate (but not always). At the end of our calculations, we may discover that there is no consistent estimator of the desired type.

The method is illustrated in some examples. To simplify the writing, I will use notation that leaves out x whenever possible. I write f for $f(x)$, f' for $f'(x)$, etc.

Example 1: Estimate $f'(x)$ from $f(x)$ and $f(x+h)$ Solution: The estimator is (dropping the x argument)

$$f' \approx a_1(h)f + a_2(h)f(x+h) \ .$$

Now expand in Taylor series:

$$f(x+h) = f + f'h + \frac{1}{2}f'' + \cdots \ .$$

The estimator is

$$f' \approx a_1(h)f + a_2(h)f + a_2(h)f'h + a_2(h)f''h^2 + \cdots \ . \tag{25}$$

Looking at the right hand side, we see various coefficients, f, f', and so on. Since the relation is supposed to work whatever f, f', etc. values we use, we might want to set the coefficient of f to zero, etc. We also see various powers of h. At first we do not know what the powers are. We could guess them, particularly when we have more experience. For now, we will match the coefficients of f and its derivatives up to whatever order is possible. Since there are two coefficients, the first two conditions will determine them, hopefully. Equating the coefficient of f on both sides of (25) gives

$$0 = a_1(h) + a_2(h) \quad .$$

Next we equate the coefficients of f' from both sides:

$$1 = a_2(h)h \; .$$

Solving these gives

$$a_2 = \frac{1}{h} \quad , \quad a_1 = \frac{-1}{h} \; .$$

The estimate, then, is

$$\begin{aligned} f'(x) &\approx \frac{-1}{h} f(x) + \frac{1}{h} f(x+h) \\ &= \frac{f(x+h) - f(x)}{h} \quad . \end{aligned}$$

This is the first order one sided difference approximation we saw earlier.
Example 2: Estimate $f'(x)$ from $f(x)$, $f(x-h)$, $f(x+h)$, $f(x+2h)$. Note: this is not centered nor is it completely one sided, but it is biased to one side. This approximation has proven useful in high accuracy wave simulations. Solution: This time we guess that all the coefficients have a power $1/h$. Thus we make the guess:

$$f'(x) = \frac{1}{h} \left(a_{-1} f(x-h) + a_0 f + a_1 f(x+h) + a_2 f(x+2h) \right) \quad .$$

The Taylor series expansions are

$$\begin{array}{rcccccccccc} f(x-h) &=& f &-& f'h &+& \frac{f''}{2}h^2 &-& \frac{f'''}{6}h^3 &+& \frac{f^{(4)}}{24}h^4 + \cdots \\ f(x+h) &=& f &+& f'h &+& \frac{f''}{2}h^2 &+& \frac{f'''}{6}h^3 &+& \frac{f^{(4)}}{24}h^4 + \cdots \\ f(x+2h) &=& f &+& 2f'h &+& 2f''h^2 &+& \frac{4f'''}{3}h^3 &+& \frac{2f^{(4)}}{3}h^4 + \cdots \end{array}$$

Equating powers of h, which turns out to be the same as equating the coefficients of f, f', etc. from both sides gives, in order

$$\begin{array}{rll} f\,, \;\; O(h^{-1}) : & \quad & 0 = a_{-1} + a_0 + a_1 + a_2 \\ f'\,, \;\; O(h^0) : & & 1 = -a_{-1} + a_1 + 2a_2 \\ f''\,, \;\; O(h^1) : & & 0 = \frac{1}{2}a_{-1} + \frac{1}{2}a_1 + 2a_2 \\ f'''\,, \;\; O(h^2) : & & 0 = \frac{-1}{6}a_{-1} + \frac{1}{6}a_1 + \frac{4}{3}a_2 \end{array}$$

We could compute the $O(h^3)$ equation but already we have four equations for the four unknown coefficients. If we would use the $O(h^3)$ equation in place of the $O(h^2)$ equation, we loose an order of accuracy in the resulting approximation.

These are a system of 4 linear equations in the four unknowns a_{-1} through a_2. Rather than going through a general solution procedure, we can preceed in an ad hoc way. Notice that the combination $b = -a_{-1} + a_1$ appears in the second and fourth equations. If we substitute b, these equations are

$$1 = b + 2a_2 \ ,$$
$$0 = \frac{1}{6}b + \frac{4}{3}a_2 \ .$$

which implies that $b = -8a_2$ and then that $a_2 = -\frac{1}{6}$ and $b = \frac{4}{3}$. Then, since $-4a_2 = \frac{2}{3}$, the fourth equation gives $a_{-1} + a_1 = \frac{2}{3}$. Since $b = \frac{4}{3}$ is known, we get two equations for a_{-1} and a_1:

$$a_1 - a_{-1} = \frac{4}{3} \ ,$$
$$a_1 + a_{-1} = \frac{2}{3} \ .$$

The solution is $a_1 = 1$ and $a_{-1} = \frac{-1}{3}$. With these, the first equation leads to $a_0 = \frac{-1}{2}$. Finally, our approximation is

$$f'(x) = \left(\frac{-1}{3} f(x-h) - \frac{1}{2} f(x) + f(x+h) - \frac{1}{6} f(x+2h) \right) + O(h^3) \ .$$

Often these methods may be found by making the estimator exact on polynomials up to some degree. This might make the algebra simpler.
Example 3: Estimate $f''(x)$ as accurately as possible from $f(x)$, $f(x+h)$, $f'(x)$ and $f'(x+h)$. Solution: The estimator we seek has the form

$$f''(x) \approx af(x) + bf(x+h) + cf'(x) + df'(x+h) \ .$$

We can determine the four unknown coefficients a, b, c, and d by requiring the approximation to be exact on constants, linears, quadratics, and cubics. It does not matter what x value we use, so let us take $x = 0$. This gives, respectively, the four equations:

$$\begin{aligned} 0 &= a + b && \text{(constants, } f = 1) \ , \\ 0 &= bh + c + d && \text{(linears, } f = x) \ , \\ 1 &= b\tfrac{h^2}{2} + dh && \text{(quadratics, } f = x^2/2) \ , \\ 0 &= b\tfrac{h^3}{6} + d\tfrac{h^2}{2} && \text{(cubics, } f = x^3/6) \ , \end{aligned}$$

Solving these gives

$$a = \frac{-6}{h^2} \ , \quad b = \frac{6}{h^2} \ , \quad c = \frac{-4}{h} \ , \quad d = \frac{-2}{h} \ .$$

and the approximation

$$f''(x) \approx \frac{6}{h^2} \left(-f(x) + f(x+h) \right) - \frac{2}{h} \left(2f'(x) + f'(x+h) \right) \ .$$

A Taylor series calculation shows that this is second order accurate.

7 Adaptive parameter estimation

In most real computations, the computational strategy is not fixed in advance, but is adjusted as the computation proceeds. These are called "adaptive" algorithms. At this point, we discuss adaptive methods for finding appropriate parameters such as the step size h. It is preferable to have the computer rather than the user choose h. The user may not know or wish to know enough about the algorithm to choose h appropriately. Even if he or she did, finding h might be a time consuming trial and error process we would willingly avoid, particularly if the operation must be done repeatedly as part of a larger computation. A simple example of this situation is in the homework, which asks the student to make a plot of $f(x)$, which is defined as an integral in which x is a parameter.

We want a program that takes our function, and a desired level of accuracy, ϵ, and returns an approximation, $\widehat{A}$ with $\left|\widehat{A} - A\right| \leq \epsilon$ with a high degree of confidence. In many cases, this may be done using Richardson extrapolation ideas discussed above. Suppose, for example, that we seek A and have a second order approximation $A(h)$. If h is small enough we can estimate the error as

$$E(h) = A(h) - A \approx \frac{4}{3}\left(A(h) - A(\frac{h}{2})\right) \ .$$

Thus, if we calculate the two approximations $A(h)$ and $A(h/2)$, we may estimate the error. The simplest adaptive algorithm using this idea would be to repeatedly reduce h by a factor of two until the error estimate satisfies the error tolerence we seek. More formally

$$\begin{array}{l} \texttt{while} \quad \left(\left|\frac{4}{3}\left(A(h) - A(\frac{h}{2})\right)\right| \geq \epsilon\right) \\ \quad h = h/2 \ ; \\ \texttt{return} \quad A(h) \ ; \end{array} \tag{26}$$

While this algorithm is based on manipulations with error expansions, essintially the same algorithm can be based on simpler ideas. Assuming that the approximations $A(h)$ will converge to A as $h \to 0$, we just compare two approximations with different h values. That is we compute $|A(h) - A(h/2)|$. If this quantity is small enough, we assume that h is small enough. In other words, the algorithm (26) may be viewed as repeatedly reducing h by a factor of two until the answer stops changing. The Richardson calculation, and the factor $4/3$, just tells us is that when we do this, our answer is likely to be nearly within ϵ of A.

We can base a reasonably reliable piece of software on refinements of the basic strategy (26). The main drawbacks of (26) are that

(i) It may be an infinite loop.

(ii) It might terminate early if the initial h is too large to be in the "asymptotic range".

(iii) If the function is such that the approximation $A(h)$ does not have an asymptotic error expansion, the program will not detect this.

(iv) It does not return the best possible estimate of A.

To address point (i) we need some criterion for giving up. When it gives up it should return an error flag. In software like this it is also a good idea to print an error message with information about the failure. Under no circumstances will we tolerate software that fails silently. Exactly what the criteria should be may depend on the problem but here are some ideas.

Suppose our application is numerical differentiation, that is, $A = f'(x)$. Then we may ask for, or probably should expect, an initial guess from the user of a plausable step size, h_0. This need not be very accurate, just an indication of the relevent length scales of the problem. For example, in atomic physics 10^{-10} meters might be a natural length scale. It might happen that a good step size for differentiation is 1000 times smaller than that, but not 10^{20} times smaller. The revised program might look like

```
h = h0;
hMin = 10*epsMach*h0;
while  ( | 4/3 (A(h) - A(h/2)) | ≥ ε )
  if ( h <= hMin ) {
    Print an error message.
    errorCode = HMIN_REACHED; return 0;}
  h = h/2 ;
return  A(h) ;
```
(27)

If we reach $h < h_{\min}$ we are probably doomed anyway – the subtraction in the finite difference operation will have lost all accuracy. In double precision, this corresponds to 51 trips throught the loop.

If we are estimating an integral, we probably do not want to let h be as anywhere near as small as $\epsilon_{\text{mach}} h_0$, because this would lead to a huge number of panels. We could also take the size of the integration interval $b - a$, as the "natural length scale" for integration. If we define $A(n)$ to be the panel method estimate with n equal size panels, then one strategy might be

```
#define N_MAX 1000000 \* Max number of panels *\
 ...
 ...
while  ( | 4/3 (A(n) - A(2n)) | ≥ ε )
  if ( n >= N_MAX ) {
    errorCode = N_MAX_REACHED; return; }
    Print an error message.
  n = 2 * n ;
return  A(n) ;
```
(28)

We might want a more sophisticated version of this that would warn the user if, say, more than 1000 panels were needed.

To address point (ii), we might want to apply the convergence analysis appropriate to situations where the answer is not known. For this we need three

values, say, $A(h)$, $A(2h)$, and $A(4h)$. We might consider continuing to refine (reduce h) as long as the ratio $(A(2h) - A(h))/(A(4h) - A(2h))$ differs from 2^p (p being the order of accuarcy) by more than a specified tolerence, say .1. It might happen that we become convinced that we have a good approximation to the answer even though the convergence analysis ratio is far from its expected value. This might be due to irregularities in the function, possibly a jump in some derivative, that is severe enough to prevent the asymptotic error analysis from holding but not to prevent convergence. It would be worthwhile returning an error flag in this case. Several commercial numerical software packages do this. The point of checking for the asymptotic convergence rate to be established is that it might happen that two $A(h)$ and $A(2h)$ are nearly equal without either being a good approximation. No amount of checking like this will guarantee that $A(h)$ is close to A. After all, if we know the termination strategy, we can design a function to defeat it. Looking at three rather than two values makes it far less likely to be fooled by accident. This also addresses point (iii).

Point (iv) has two parts. First is a simple observation that underlies a paradox of most adaptive methods: there is never an accurate error estimate. Suppose we had a reliable error estimate, such as (for second order approximations) $\widehat{E}(h) = \frac{4}{3}(A(2h) - A(h))$. If we believed in this estimate, we would add it to our approximate solution to produce a better approximation

$$\widehat{A} = A(h) + \widehat{E}(h) \ .$$

We have gained much accuracy in the approximation but lack any estimate of the size of the remaining error. Adaptive estimations in elementary numerical analysis often use the error for both determining the appropriate h and for correcting the eventual answer. This usually leads to the reported answer having far more accuracy than the ϵ asked for.

There are two common computational situations where this paradox does not hold. In Monte Carlo computation, it is often possible to find statistical "error bars" that essentially say that with high probability the true answer is within a given interval around the estimated answer. The error bars may be made smaller by running the computation longer. Error bar computations do not estimate the error, they only give a bound on it's likely size. Adaptive solution of partial differential equations are often driven by "a-posteriori" error estimates. The term a posteriori means that the refinement information is constructed from a tentative computed approximate solution. The estimates take the form $|E| \leq const \cdot LEE$. Here, as in our earlier discussions, the constant *const* is not known exactly, but we hope it is not too large. The "local error estimate" *LEE* is constructed from the tentative solution in some way. Again we have an bound on the error without knowing the error itself.

8 References and resources

For a review of one variable calculus, I recommend the Schaum outline. The chapter on Taylor series explains the remainder estimate clearly.

The reader wishing to explore the material from this chapter further should look at **Numerical Methods** by G. Dahlquist and Å. Björk. In particular, there is a beautiful "operator calculus" of finite difference operators. Another classic that covers this material in a more pragmatic way is **Numerical Analysis** by E. Isaacson and H. Keller.

Software for numerical integration and interpolation is available from ...

9 Review questions

1. How is Taylor series as a convergent series different from Taylor series as an asymptotic expansion?

2. What is the difference between an error bound and an asymptotic error expansion, say, for a second order accurate approximation?

3. Will Simpson's rule (1.??) give fourth order approximations regardless of the funcion is is applied to? Describe some functions for which Simpson's rule will or will not give fourth order accuracy.

4. How might asymptotic error expansions help us improve the accuracy of a computation even though we cannot compute the coefficients before hand?

5. How might an asymptotic expansion help us validate a piece of computational software? Discuss the situations when an exact answer is or is not known.

6. How might an asymptotic error expansion help us achieve a specified accuracy when the step size needed for this is not known in advance?

7. Why do the coefficients in the asymptotic error expansion for panel method integration depend on ly on the function and its derivatives at the endpoints?