Math 563 Lecture Notes
Differentiation, Richardson Extrapolation

Spring 2020

The point: Finite difference methods for derivatives (the standard tools) are introduced, making use of Taylor series and polynomial interpolation. Some general principles - rounding vs. truncation error, Richardson extrapolation and error estimates - are also illustrated here. Later, we'll see another (more involved) approach using the Fourier transform.

Related reading: Quarteroni, Chapter 10 (the section on differentiation; 10.10.1)

1 Finite difference formulas

The goal here is to construct a formula to estimate the derivative $f'(x)$ (or $f''(x)$, etc.) at a point x using a set of sample points, called a **finite difference formula**. We are either given the function $f(x)$ or the values at a set of points; like interpolation, both will use the same methods.

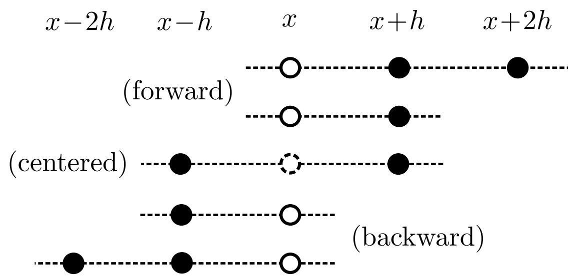
Terminology: In important property of some approximations is that they are exact for all polynomials of degree up to k (but not any higher degree). For example, the formula

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

is an approximation of degree 1 since the error is zero for linear polynomials (for which $f'' = 0$ so the error term vanishes) but not for quadratics ($f'' \neq 0$). We sometimes say this approximation 'has **degree** of accuracy k'.

We say the approximation has **order** (of accuracy) p if the error scales like $O(h^p)$ as $h \to 0$, and this is the best possible $O(\cdots)$ in general (so the error is $\Theta(h^p)$). The forward difference above has order 1 (a 'first-order' formula).

The **stencil** is the set of points used in the approximation, illustrated in a diagram (useful for more complicated problems like in 2d). Some stencils for equally spaced points are shown below (the point x where $f'(x)$ is to be estismated is in white):



Formulas that only use points $\geq x$ / $\leq x$ are called **forward** / **backward**, respectively.

1.1 The interpolation approach

A straightforward way to compute the derivative is to use the interpolant in Lagrange form. For reasons to be made clear, assume the evaluation point x_k is one of the nodes $x_0, \cdots, x_n$.[1]

Let $p_n(x)$ be the polynomial interpolant for these nodes. It satisfies

$$f(x) = \underbrace{\sum_{i=0}^{n} f_i \ell_i(x)}_{p_n(x)} + \frac{f^{(n+1)}(\eta_x)}{(n+1)!}\omega_n(x), \qquad \omega_n(x) := \prod_{j=0}^{n}(x - x_j).$$

Differentiate this once to get

$$f'(x) = \sum_{i=0}^{n} f_i \ell_i'(x) + \underbrace{\frac{1}{(n+1)!}\left(f^{(n+1)}(\eta_x)\omega_n'(x) + (\cdots)\cdot\omega_n(x)\right)}_{E(x)}.$$

The term in the ellipsis is not nice in general (note that η_x is a function of x!). However, ω_n vanishes at x_k - the reason we required this point to be a node. Plugging it in leaves

$$E(x_k) = \frac{f^{(n+1)}(\eta)}{(n+1)!}\omega_n'(x_k) = \frac{f^{(n+1)}(\eta)}{(n+1)!}\prod_{j=0,\, j\neq k}(x_k - x_j)$$

for some η between x_0 and x_n. Thus, we have a formula of the form

$$f'(x_k) = \sum_{i=0}^{n} f_i \underbrace{\ell_i'(x_k)}_{\text{constants}} + \frac{f^{(n+1)}(\eta)}{(n+1)!}\underbrace{\textstyle\prod_{j=0,\, j\neq k}(x_k - x_j)}_{\text{constant}} \tag{1}$$

which is a linear combination of the f_i's. That is,

$$f'(x_k) = \sum_{i=0}^{n} c_i f_i + C\frac{f^{(n+1)}(\eta)}{(n+1)!}$$

where the constants c_i and C depend on the nodes $x_0, \cdots, x_n$.

[1] If given data but not $f(x)$, and estimating $f'(x)$ at a value x in beteen nodes, then we may not make such an assumption. The construction still works, but the error term is not as nice.

Example: We derive the 'centered difference' formula for $f'(0)$ given points $-h, 0$ and h. Choose interpolation nodes

$$x_0 = -h, \quad x_1 = 0, \quad x_2 = h$$

with p' to be evaluated at $x_1 = 0$. The Lagrange basis polynomials are

$$\ell_0 = \frac{x(x-h)}{2h^2}, \quad \ell_1 = -\frac{(x+h)(x-h)}{h^2}, \quad \ell_2 = \frac{(x+h)x}{2h^2}.$$

Aside (calculation trick): Note that differentiating and evaluating at $x = 0$ is not too bad using the general rule

$$((x-a)g(x))'\big|_{x=a} = \big((x-a)g'(x) + g(x)\big)\big|_{x=a} = g(a)$$

(i.e. because $x - a$ is a factor, one term in the product rule vanishes at $x = a$).

Now use the box to compute the derivatives of ℓ_0 and ℓ_2:

$$\ell_0'(0) = \frac{x-h}{2h^2}\Big|_{x=0} = -\frac{1}{2h}, \quad \ell_2'(0) = \frac{x+h}{2h^2}\Big|_{x=0} = \frac{1}{2h}.$$

Finally, compute directly for ℓ_1 (no $x - x_1$ factor, so the box doesn't apply):

$$\ell_1'(0) = \left(\frac{h^2 - x^2}{h^2}\right)'\Big|_{x=0} = 0.$$

The error is given by (again using the general rule, just remove the $x - x_1$ factor)

$$E = \left(\frac{f^{(3)}(\eta_x)}{6}\prod_{j=0}^{2}(x - x_j)\right)'\Big|_{x=0} = \frac{f^{(3)}(\xi)}{6}(0+h)(0-h) = -\frac{f^{(3)}(\xi)}{6}h^2.$$

We have now derived the order 2 **centered difference** for $f'(x)$,

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{f^{(3)}(\xi)}{6}h^2.$$

Note that even though there are only two points used in the final approximation (which should be good only up to lines), the formula is actually exact for quadratics too.

Observations: The formula (1) we derived suggests some patterns. Suppose the spacing between each point is a multiple of h, so $x_j - x_0 = n_j h$. There are n factors in the product, so the formula has the form

$$f'(x_k) = \sum_{i=0}^{n} c_i f_i + \frac{C f^{(n+1)}(\eta)}{(n+1)!}h^n.$$

(e.g. $C = -1/6$ in the example above). We conclude that:

- An approximation built from an interpolant with $n+1$ points and spacing of 'size' h has an error $O(h^n)$ - one less factor of h than for interpolating $f(x)$ itself.
- This approximation is exact for polynomials of degree $\leq n$ due to the $f^{(n+1)}$ factor.
- The more centered a formula is, the better the constant in front of the h^n. This is because the product in the constant of the error term,

$$C = \prod_{j \neq k} (x_k - x_j)$$

  is smallest when the evaluation point x_k is placed symmetrically (vs. to one side).
- It is possible for a coefficient to be zero (as in the centered formula), in which case **the number of function evaluations may seem to violate the number of points rule**. The '$n+1$ points means $O(h^n)$' counts coefficients that may be zero.

1.2 Taylor series approach (the nicer way)

The interpolation method has the disadvantage that it does not generalize well to higher order derivatives. Instead, Taylor series expansion will do the trick, and is easier to work with.

The finite difference formula will have the form

$$\frac{1}{h^m} \sum_{i=0}^{n} c_i f(x + a_i h) = f^m(x) - \text{error}$$

where m is the order of the derivative. The strategy is simple:

- (Approximate) Expand $f(x + a_i h)$'s in a Taylor series to at least order $m+1$
- (Solve for coeffs.) Choose c_i's so that the terms up to order m leave only $f^{(m)}$
- (Find the error) Try to cancel out terms past the m-th; the first term that is non-zero will be the leading term of the error.

This process is best illustrated by example. We compute what is perhaps the most used finite difference approximation, the centered difference (for f'') of the form

$$\delta^2 f = \frac{af(x-h) + bf(x) + cf(x+h)}{h^2} = f''(x) - E.$$

The choice of $1/h^2$ is made since the m-th derivative should have $1/h^m$. Expand out the f's:

$$f(x \pm h) = f(x) \pm hf'(x) + \frac{h^2}{2} f''(x) \pm \frac{h^3}{6} f'''(x) + \frac{h^4}{24} f^{(4)}(\eta_\pm)$$

where the choice to stop at h^4 vs. h^3 will be clear later. Now plug this in to find that

$$\delta^2 f = \frac{(a+b+c)}{h^2} f(x) + \frac{(c-a)}{h} f'(x) + \cdots$$

and immediately observe that $c = a$ to cancel out the $f'(x)$ term. Then **all terms of odd order cancel**, so for unknown points η_- and η_+, which leaves

$$\delta^2 f = \frac{(a+b+a)}{h^2} f(x) + \frac{a+a}{2} f''(x) + \frac{h^2}{24}(af^{(4)}(\eta_-) + af^{(4)}(\eta_+)).$$

It follows that $a = 1$ and $b = -2$ from the first two terms. The formula is then

$$\frac{f(x-h) - 2f(x) + f(x+h)}{h^2} = f''(x) + \frac{h^2}{12} f^{(4)}(\xi) \quad \text{for some } \xi \text{ in the interval.}$$

after using the intermediate value theorem to simplify the error (see remark below).

Key observation: Note that the symmetry of the centered formula means that every other term in the error series will cancel. This provides one extra order of accuracy for free.

In general, a formula for $f''(x)$ using three points will have an error $O(h)$ - we have three coefficients to satisfy three terms (f, f', f'' in the Taylor series). The symmetry makes the **next** term also vanish, producing one extra order.

This simple fact enables numerical codes to be quite efficient - e.g. for solving second-order ODEs and PDEs, since we get a second-order accurate formula using only three points.

Simplifying the error (IVT trick): The intermediate value theorem can be used to combine $f^{(k)}(\eta)$ terms together when the η's are different. The relevant version is the following:

Lemma (IVT) Let $f(x)$ be continuous in an interval $[a, b]$ and let $c_1, \cdots c_n$ be **positive** constants with $\sum c_i = 1$. If $\xi_1, \cdots, \xi_n$ are points in $[a, b]$ then

$$\sum_i c_i f(\xi_i) = f(\xi)$$

for a single value $\xi \in [a, b]$.

Proof. Let $m = \min_{x\in[a,b]} f(x)$ and $M = \max_{x\in[a,b]} f(x)$. Then, since the c_i's are positive,

$$m = m\sum_i c_i \leq \sum_i c_i f(\xi_i) \leq M \sum_i c_i = M$$

Thus the sum is between m and M (in the range of f), so by the intermediate value theorem, there is a point ξ such that it equals $f(\xi)$ as desired. □

This, for instance, allows us to simplify terms like the following (factor out a 3/2 here):

$$\frac{3}{4} f^{(k)}(\xi_1) + \frac{3}{4} f^{(k)}(\xi_2) \to \frac{3}{2} f^{(k)}(\xi).$$

The shortcut: Note that the terms simplify **as if the ξ's were equal**. In fact, we can argue this 'shortcut' of setting all the ξ_i's to one ξ must be valid. From the Lagrange approach, the error must be a single term. Then pick a polynomial of degree such that the $f^{(k)}$ in the error formula is constant. The shortcut is correct for such a polynomial and determines the only unknown constant.

Thus, one obtains the shortcut that

the error term can be obtained by setting all ξ's to be the same value.

This rule holds even if the terms have opposite signs, where the IVT trick cannot be applied. The underlying theory is that of the **Peano kernel**, which provides a rigorous guarantee that the shortcut is valid if the approximation is exact for polynomials up to some degree.

1.3 Another typical example

Suppose the first-order backward difference

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

is to be improved in accuracy. We can add a point at $x - 2h$ to increase the order by one, and derive the three point (or $O(h^2)$) backward difference formula

$$D(h) = \frac{af(x) + bf(x-h) + cf(x-2h)}{h}.$$

There are three coefficients and no symmetry, so we need three terms plus an error term. Expand the f's in a Taylor series:

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2} f''(x) - \frac{h^3}{6} f'''(\xi_1)$$
$$f(x-2h) = f(x) - 2hf'(x) + 2h^2 f''(x) - \frac{4h^3}{3} f'''(\xi_2).$$

Plugging in to the formula,

$$D(h) = \frac{(a+b+c)}{h} f(x) - (b+2c)f'(x) + h(\frac{b}{2} + 2c)f''(x) - h^2\left(\frac{b}{6} f'''(\xi_1) + \frac{4c}{3} f'''(\xi_2)\right)$$

which gives $a + b + c = 0$ and

$$1 = b + 2c, \quad 0 = b/2 + 2c.$$

Solving for the coefficients, we obtain the formula

$$\frac{-3f(x) + 4f(x-h) - f(x-2h)}{2h} = f'(x) - h^2\left(\frac{1}{3} f'''(\xi_1) - \frac{2}{3} f'''(\xi_2)\right).$$

Using (without proof) the rule that the ξ_i's can be taken to be the same,

$$\frac{-3f(x) + 4f(x-h) - f(x-2h)}{2h} = f'(x) + \frac{h^2}{3} f'''(\xi). \tag{2}$$

Note that the 'IVT trick' cannot be used here because the coefficients have opposite signs; we must appeal to the shortcut. The error is $O(h^2)$ - adding a point improves the order and degree of accuracy by one.

1.4 Rounding error

Recalling the motivating example from class, let us see why one must be careful when calculating derivatives numerically. Let's consider the centered difference for $f''(x)$,

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

The truncation error E_T satisfies

$$|E_T| = \frac{h^2}{12} |f^{(4)}(\xi)|.$$

Suppose, to be optimistic, that f is computed to machine precision (an error $\leq \mathtt{u}_m \approx 1.2 \times 10^{-16}$). Then the rounding error has the bound

$$|E_r| \leq \frac{\mathtt{u}_m + 2\mathtt{u}_m + \mathtt{u}_m}{h^2} = \frac{4\mathtt{u}_m}{h^2}.$$

Now let $M = \max |f^{(4)}(x)|$ over $[x-h, x+h]$. Then the total error is bounded as follows:

$$|E(h)| \leq \frac{Mh^2}{12} + \frac{4\mathtt{u}_m}{h^2}.$$

Minimizing the right hand side, we find that

$$\text{bound minimized when } Mh/6 = 8\mathtt{u}_m/h^3 \implies h^4 = 48\mathtt{u}_m/M$$

$$\implies h^* = (48\mathtt{u}_m/M)^{1/4}.$$

If $M \approx 1$, then this value and the associated error bound is

$$h^* \approx (48 \cdot 10^{-16})^{1/4} \approx 2.5 \times 10^{-4}, \quad |E(h^)| \leq 1.3 \times 10^{-8}.$$

The best possible error has a magnitude of about 10^{-8}, and it is achieved when h is of size 10^{-4}, which is not that small. Thus, one should be careful when approximating higher order derivatives, because **the optimal values of h may be larger than expected**.

Moreover, there are serious concerns about the error - the possible accuracy is much less than that in the evaluation of f itself. Sometimes, one just accepts the limitations on error, or finds some way to avoid using derivatives in the calculations.

Some tools can be used to improve error by iterating on the formula - one powerful technique is introduce in the next section.

When dealing with finite sets of data, there are other issues that arise; see HW.

2 Richardson extrapolation

Here we introduce a powerful, rather general tool for both improving the accuracy of a formula and estimating error. The underlying idea is of fundamental importance to numerical analysis. In addition, we gain the essential practical skill of estimating error when the exact solution is not available!

2.1 Asymptotic error

Previously, we sought error in the form of 'one term' formulas like (for Taylor series)

$$f(x) = T_n(x) + \frac{f^{(n+1)}(\eta_x)}{(n+1)!}(x - x_0)^{n+1}.$$

The full Taylor series also yields an **asymptotic error series**

$$f(x) = T_n(x) + \sum_{i=n+1}^{\infty} \frac{f^{(i+1)}(x_0)}{(i+1)!}(x - x_0)^i.$$

so the error, given a choice of x_0, has the form

$$E(x) = c_{n+1}(x - x_0)^{n+1} + c_{n+2}(x - x_0)^{n+2} + \cdots$$

where each coefficient c is known, at least in theory (unlike the η_x from the one-term form). The advantage is that the nice dependence on x is retained.

In general: An **asymptotic series** in a limit $x \to L$, written as

$$f(x) \sim \sum_{i=0}^{\infty} g_i(x)$$

is a series such that each term is asymptotically smaller than the previous and f is asymptotic to any finite truncation as $x \to L$. That is,

$$f \sim \sum_{i=0}^{N} g_i(x) \text{ for each } N, \qquad g_{i+1} = o(g_i)$$

For example, every Taylor series is an asymptotic series (since $g_i = (x - x_0)^i$ is a sequence of functions of decreasing size in the o-sense). However, an asymptotic series is much more general because the infinite series **does not need to converge** (and the functions in the sum could be anything).

2.2 Richardson extrapolation: improving estimates

The idea of **Richardson extrapolation** is to use an asymptotic error series to get two approximations with related error, then 'cancel out' the leading term to do better.

The $O(h^2)$ centered difference formula serves as an illustrative example. Define

$$L = f'(x), \qquad A(h) = \frac{f(x+h) - f(x-h)}{2h}. \tag{3}$$

Suppose we pick a value of h and compute $A(h)$, but the approximation is not good enough. At this point we would like to know how to

- Cheaply improve the approximation, using the code for $A(h)$ and not much more
- Check the error to know for sure when it is good enough

Consider the full (infinite) Taylor series

$$f(x \pm h) = f(x) \pm hf'(x) + \frac{h^2}{2}f''(x) \pm \frac{h^3}{3!}f'''(x) + \cdots .$$

Every odd-power term cancels when computing $f(x+h) - f(x-h)$, and so

$$A(h) = L + \sum_{i=1}^{\infty} c_{2i}h^{2i} = L + c_2h^2 + c_4h^4 + \cdots \tag{4}$$

for some constants $c_2, c_4, \cdots$ whose values are not important. Now observe that if we have approximations using h and $h/2$, the error series are related (by plugging $h/2$ into (4)):

$$A(h) = L + c_2h^2 + c_4h^4 + \cdots , \qquad A(h/2) = L + \frac{c_2}{4}h^2 + \frac{c_4}{16}h^4 + \cdots$$

The h^2 term in the error can then be canceled out by combining the two. Or, we can use $A(h)$ and $A(h/2)$ to 'solve' for the h^2 term and get an estimate of the error (see next section).

Canceling the term: Take $4\times$ the second equation minus the first:

$$4A(h/2) - A(h) = 3L - \frac{3}{4}c_4h^4 - \frac{15}{16}c_6h^6 + \cdots$$

where $\cdots$ is a series in powers of h^2 starting with h^8. It follows that

$$\frac{4}{3}A(h/2) - \frac{1}{3}A(h) = L + \sum_{i=2}^{\infty} b_{2i}h^{2i} = b_4h^4 + b_6h^6 + \cdots$$

for constants $b_4, b_6, \cdots$. We have obtained a new formula $A_1(h) = \frac{4}{3}A(h/2) - \frac{1}{3}A(h)$ that is a **fourth-order** (error $O(h^4)$) formula for L. Moreover, this formula can be computed from knowing only $A(h)$ (if, say, the code is given and can't be changed).

We can also plug in the definition of A from (3) to get

$$\begin{aligned} A_1(h) &= \frac{4}{3}\left(\frac{f(x+h/2)-f(x-h/2)}{h}\right) - \frac{1}{3}\left(\frac{f(x+h)-f(x-h)}{2h}\right) \\ &= \frac{-f(x+h)+8f(x+h/2)-8f(x-h/2)+f(x-h)}{6h} \end{aligned}$$

which is the $O(h^4)$ centered difference formula for f' using points $x \pm h$ and $x \pm h/2$.

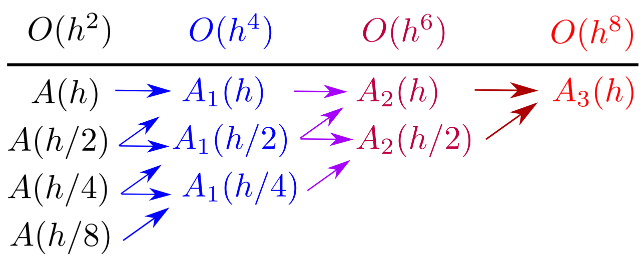
Iterating the process: But we can keep going! The approximation $A_1(h)$ is also an asymptotic error series

$$A_1(h) = L + b_4 h^4 + b_6 h^6 + \cdots .$$

Following the same process with $A_1(h)$ and $A_1(h/2)$ we obtain the formula

$$\begin{aligned} A_2(h) &= \frac{16}{15} A_1(h/2) - \frac{1}{15} A_1(h) \\ &= \frac{1}{45} A(h) - \frac{20}{45} A(h/2) + \frac{64}{45} A(h/4). \end{aligned}$$

The result is an $O(h^6)$ formula using A_1 at h and $h/2$ - so the formula uses the original approximation $A(h)$ at $h, h/2$ and $h/4$. Thus, the formula $A_2(h)$ uses the six points $\pm h, \pm h/2$ and $\pm h/4$ (this is the six-point centered difference formula).



In general, iterating the process yields a table of approximations

$$A_k(h2^{-j})$$

where the k-th approximation depends on A_{k-1} at each value of $h2^{-j}$ as shown above. The table can be computed efficiently, since each formula is just a simple linear combination of the previous ones. The result is a sequence of formulas

$$A_k(h) = \frac{1}{4^k - 1}(4^k A_{k-1}(h/2) - A_{k-1}(h)) = L + O(h^{2k}). \tag{5}$$

The procedure here is called **Richardson extrapolation**. It is a convenient (and remarkable effective!) way to improve accuracy without much (computational) effort.

Why extrapolation? Note that the desired value is the limit as $h \to 0$ of the 'function' $A(h)$. One can think of Richardson extrapolation as taking function data of $A(h)$ vs. h at sample points and extrapolating to $h = 0$.

To show the utility of the method, consider the centered difference formula Ah) and

$$f(x) = e^{2x}, \quad x_0 = 0, \quad h = 0.1.$$

Applying the centered difference formula, we obtain the following table for $A_k(h2^{-j})$:

j	A	A_1	A_2	A_3
0	2.0133600254	1.9999933254	2.0000000004	1.9999999999999998
1	2.0033350004	1.9999995832	2.000000000006205	
2	2.0008334375	1.9999999740	–	
3	2.0002083398	–		

The errors are:

j	A	A_1	A_2	A_3
0	1.336003×10^{-2}	-6.674608×10^{-6}	3.971161×10^{-10}	$-2.220446 \times 10^{-16}$
1	3.335000×10^{-3}	-4.167907×10^{-7}	6.204814×10^{-12}	
2	8.334375×10^{-4}	-2.604360×10^{-8}	–	
3	2.083398×10^{-4}	–		

The approximations $A_k(h)$ are in the first row; note that the last one is extremely accurate (to machine precision!). This result only required calling the formula to generate the first column (A); the rest was computed trivially with (5).

Note (avoiding rounding error): Incidentally, this process also illustrates an example of **iterative refinement**, where iterating on a not-so-accurate result can improve it, even in problems where rounding error might otherwise be trouble. The error in A_3 is at machine precision - much better than what rounding error allows for $A(h)$ at any h.

2.3 In general

Notice that while the centered difference was used as an example, almost none of its properties were used. What, then is required, for Richardson extrapolation? The requirements are:

- An approximation $A(h)$ that can be computed for several values of h
- Prior knowledge that an error series exists of the form

  $$A(h) = L + \sum_{k=1}^{\infty} c_k h^{n_k} \tag{6}$$

  for an increasing sequence of exponents n_k (e.g. $n_k = 2k$).
- The exponents n_k in the error series. We do **not** need to know the coefficients c_k!

In this sense Richardson extrapolation is an almost-free improvement once the hard work of the theory is done. The approximation formula $A(h)$ could even be some 'black box' algorithm (e.g. from a software library) that cannot change.

2.4 Essential variant: error estimation

We can use the same trick to estimate the error in an approximation instead. This is particularly useful when deciding whether to continue with the extrapolation table - we can stop once the estimated error is small enough. Consider

$$A(h) = L + E(h)$$

where $E(h)$ is the (unknown) error in the approximation. Suppose $A(h)$ has the series

$$A(h) = L + c_2h^2 + c_4h^4 + \cdots .$$

Then we have that

$$A(h/2) = L + c_2h^2/4 + c_4h^4/16 + \cdots$$

Rather than try to cancel the error out, we can try to 'solve' for the error using $A(h)$ and $A(h/2)$ (computable quantities) and cancel out L (unknown). Subtract the two equations:1

$$A(h) - A(h/2) = \frac{3c_2}{4}h^2 + \frac{15c_4}{16}h^4 + \cdots = \frac{3}{4}(c_2h^2) + O(h^4).$$

But observe that the error is

$$E(h) = c_2h^2 + O(h^4) \approx c_2h^2,$$

so the expression on the right can be written in terms of the error (up to $O(h^4)$):

$$E(h) = \frac{4}{3}(A(h) - A(h/2)) + O(h^4).$$

Alternately, the error in $A(h/2)$ could be found as $\approx (A(h) - A(h/2))/3$.

Key point: Combining two different approximations can yield an estimate of the error if we have some knowledge of the form of the error. This technique is a good way of doing practical error estimation for any method with an asymptotic error series.

Caution: The error estimate depends on the form of the series. For each problem, it's important to check to make sure the appropriate formula is used. One does need to derive the correct asymptotic series to know how to estimate the error!

Example: Return to the $f(x) = e^{2x}$, $x_0 = 0$ centered difference example. The error table for approximations up to A_3 was

j	A	A_1	A_2	A_3
0	1.336003×10^{-2}	-6.674608×10^{-6}	3.971161×10^{-10}	$-2.220446 \times 10^{-16}$
1	3.335000×10^{-3}	-4.167907×10^{-7}	6.204814×10^{-12}	
2	8.334375×10^{-4}	-2.604360×10^{-8}	$-$	
3	2.083398×10^{-4}	$-$		

We can also use the formulas

$$E_k(h) \approx \frac{4^{k+1}}{4^{k+1} - 1}(A_k(h) - A_k(h/2))$$

for the error $E_k(h)$ in $A_k(h)$ (first row), giving

$$E(h) \approx 1.4 \times 10^{-2}, \quad E_1(h) \approx -6.7 \times 10^{-6}, \quad E_2(h) \approx 4.0 \times 10^{-10},$$

which are quite accurate approximations to the real error!

Suppose a solution accurate to $\epsilon = 10^{-9}$ is required. Then we might start with $h = 0.1$ and:

- Compute $A(h)$, $A(h/2)$ and estimate the error to find it is $1.5 \times 10^{-2} > \epsilon$.
- Compute the next column $(A_1(h), A_1(h/2))$; estimat ethe error to find it is $7 \times 10^{-6} > \epsilon$
- Compute $A_2(h)$ and $A_2(h/2)$; since the estimate is $4 \times 10^{-10} < \epsilon$, we stop.

Note that more entries in previous columns also need to be computed here. the process allows the code to do only the necessary computations to get a desired accuracy, and as a bonus to have an estimate that justifies that accuracy (not a rigorous bound, but reasonable for a practical estimate).