Appendix A
Measuring Errors

To discuss the accuracy of a numerical solution, or the relative virtues of one numerical method, versus another, it is necessary to choose a manner of measuring that error. It may seem obvious what is meant by the error, but as we will see there are often many different ways to measure the error which can sometimes give quite different impressions as to the accuracy of an approximate solution.

A.1 Errors in a scalar value

First consider a problem in which the answer is a single value $\hat{z} \in \mathbb{R}$. Consider, for example, the scalar ordinary differential equation (ODE)

$$u'(t) = f(u(t)), \quad u(0) = \eta,$$

and suppose we are trying to compute the solution at some particular time T, so $\hat{z} = u(T)$. Denote the computed solution by z. Then the error in this computed solution is

$$E = z - \hat{z}.$$

A.1.1 Absolute error

A natural measure of this error would be the absolute value of E,

$$|E| = |z - \hat{z}|.$$

This is called the *absolute error* in the approximation.

As an example, suppose that $\hat{z} = 2.2$, while some numerical method produced a solution $z = 2.20345$. Then the absolute error is

$$|z - \hat{z}| = 0.00345 = 3.45 \times 10^{-3}.$$

This seems quite reasonable—we have a fairly accurate solution with three correct digits and the absolute error is fairly small, on the order of 10^{-3}. We might be very pleased

with an alternative method that produced an error of 10^{-6} and horrified with a method that produced an error of 10^6.

But note that our notion of what is a large error or a small error might be thrown off completely if we were to choose a different set of units for measuring $\hat{z}$. For example, suppose the $\hat{z}$ discussed above were measured in meters, so $\hat{z} = 2.2$ meters is the correct solution. But suppose that instead we expressed the solution (and the approximate solution) in nanometers rather than meters. Then the true solution is $\hat{z} = 2.2 \times 10^9$ and the approximate solution is $z = 2.20345 \times 10^9$, giving an absolute error of

$$|z - \hat{z}| = 3.45 \times 10^6.$$

We have an error that seems huge and yet the solution is just as accurate as before, with three correct digits.

Conversely, if we measured $\hat{z}$ in kilometers, then $\hat{z} = 2.2 \times 10^{-3}$ and $z = 2.20345 \times 10^{-3}$ so

$$|z - \hat{z}| = 3.45 \times 10^{-6}.$$

The error seems much smaller and yet there are still only three correct digits.

A.1.2 Relative error

The above difficulties arise from a poor choice of scaling of the problem. One way to avoid this is to consider the *relative error*, defined by

$$\left| \frac{z - \hat{z}}{\hat{z}} \right|.$$

The size of the error is scaled by the size of the value being computed. For the above examples, the relative error in z is equal to

$$\left| \frac{2.20345 - 2.2}{2.2} \right| = \left| \frac{2.20345 \times 10^9 - 2.2 \times 10^9}{2.2 \times 10^9} \right| = 1.57 \times 10^{-3}.$$

The value of the relative error is the same no matter what units we use to measure $\hat{z}$, a very desirable feature. Also note that in general a relative error that is on the order of 10^{-k} indicates that there are roughly k correct digits in the solution, matching our intuition.

For these reasons the relative error is often a better measure of accuracy than the absolute error. Of course if we know that our problem is "properly" scaled, so that the solution $\hat{z}$ has magnitude order 1, then it is fine to use the absolute error, which is roughly the same as the relative error in this case.

In fact it is generally better to ensure that the problem is properly scaled than to rely on the relative error. Poorly scaled problems can lead to other numerical difficulties, particularly if several different scales arise in the same problem so that some numbers are orders of magnitude larger than others for nonphysical reasons. Unless otherwise noted below, we will assume that the problem is scaled in such a way that the absolute error is meaningful.

A.2 "Big-oh" and "little-oh" notation

In discussing the rate of convergence of a numerical method we use the notation $O(h^p)$, the so-called big-oh notation. In case this is unfamiliar, here is a brief review of the proper use of this notation.

If $f(h)$ and $g(h)$ are two functions of h, then we say that

$$f(h) = O(g(h)) \quad \text{as } h \to 0$$

if there is some constant C such that

$$\left|\frac{f(h)}{g(h)}\right| < C \quad \text{for all } h \text{ sufficiently small}$$

or, equivalently, if we can bound

$$|f(h)| < C|g(h)| \quad \text{for all } h \text{ sufficiently small.}$$

Intuitively, this means that $f(h)$ decays to zero *at least as fast* as the function $g(h)$ does. Usually $g(h)$ is some monomial h^q, but this isn't necessary.

It is also sometimes convenient to use the "little-oh" notation

$$f(h) = o(g(h)) \quad \text{as } h \to 0.$$

This means that

$$\left|\frac{f(h)}{g(h)}\right| \to 0 \quad \text{as } h \to 0.$$

This is slightly stronger than the previous statement and means that $f(h)$ decays to zero *faster* than $g(h)$. If $f(h) = o(g(h))$, then $f(h) = O(g(h))$, although the converse may not be true. Saying that $f(h) = o(1)$ simply means that the $f(h) \to 0$ as $h \to 0$.

Examples:

$2h^3 = O(h^2)$ as $h \to 0$, since $\dfrac{2h^3}{h^2} = 2h < 1$ for all $h < \dfrac{1}{2}$.

$2h^3 = o(h^2)$ as $h \to 0$, since $2h \to 0$ as $h \to 0$.

$\sin(h) = O(h)$ as $h \to 0$, since $\sin h = h - \dfrac{h^3}{3} + \dfrac{h^5}{5} + \cdots < h$ for all $h > 0$.

$\sin(h) = h + o(h)$ as $h \to 0$, since $(\sin h - h)/h = O(h^2)$.

$\sqrt{h} = O(1)$ as $h \to 0$, and also $\sqrt{h} = o(1)$, but $\sqrt{h}$ is not $O(h)$.

$1 - \cos h = o(h)$ and $1 - \cos h = O(h^2)$ as $h \to 0$.

$h^2/\sqrt{h + h^3} = O(h^{1.5})$ and $h^2/\sqrt{h + h^3} = o(h)$ as $h \to 0$.

$e^{-1/h} = o(h^q)$ as $h \to 0$ for every value of q.

To see this, let $x = 1/h$ then $\dfrac{e^{-1/h}}{h^q} = e^{-x}x^q \to 0$ as $x \to \infty$.

Note that saying $f(h) = O(g(h))$ is a statement about how f behaves in the limit as $h \to 0$. This notation is sometimes abused by saying, for example, that if $h = 10^{-3}$, then the number 3×10^{-6} is $O(h^2)$. Although it is clear what is meant, this is really meaningless mathematically and may be misleading when analyzing the accuracy of a numerical method. If the error $E(h)$ on a grid with $h = 10^{-3}$ turns out to be 3×10^{-6}, we cannot conclude that the method is second order accurate. It could be, for example, that the error $E(h)$ has the behavior

$$E(h) = 0.003\, h, \tag{A.1}$$

in which case $E(10^{-3}) = 3 \times 10^{-6}$, but it is not true that $E(h) = O(h^2)$. In fact the method is only first order accurate, which would become apparent as we refined the grid.

Conversely, if

$$E(h) = 10^6\, h^2, \tag{A.2}$$

then $E(10^{-3}) = 1$, which is much larger than h^2, and yet it is still true that

$$E(h) = O(h^2) \quad \text{as } h \to 0.$$

Also note that there is more to the choice of a method than its asymptotic rate of convergence. While in general a second order method outperforms a first order method, if we are planning to compute on a grid with $h = 10^{-3}$, then we would prefer a first order method with error (A.1) over a second order method with error (A.2).

A.3 Errors in vectors

Now suppose $\hat{z} \in \mathbb{R}^s$, i.e., the true solution to some problem is a vector with s components. For example, $\hat{z}$ may be the solution to a system of s ODEs at some particular fixed time T. Then z is a vector of approximate values and the error $e = z - \hat{z}$ is also a vector in $\mathbb{R}^s$. In this case we can use some vector norm to measure the error.

There are many ways to define a vector norm. In general a vector norm is simply a mapping from vectors x in $\mathbb{R}^s$ to nonnegative real numbers, satisfying the following conditions (which generalize important properties of the absolute value for scalars):

1. $\|x\| \geq 0$ for any $x \in \mathbb{R}^s$, and $\|x\| = 0$ if and only if $x = \vec{0}$.
2. If a is any scalar, then $\|ax\| = |a|\, \|x\|$.
3. If $x,\ y \in \mathbb{R}^m$, then $\|x + y\| \leq \|x\| + \|y\|$ (triangle inequality).

One common choice is the max-norm (or infinity-norm) denoted by $\|\cdot\|_\infty$:

$$\|e\|_\infty = \max_{1 \leq i \leq s} |e_i|. \tag{A.3}$$

It is easy to verify that $\|\cdot\|_\infty$ satisfies the required properties. A bound on the max-norm of the error is nice because we know that every component of the error can be no greater than the max-norm. For some problems, however, there are other norms which are either more appropriate or easier to bound using our analytical tools.

Two other norms that are frequently used are the 1-norm and 2-norm,

$$\|e\|_1 = \sum_{i=1}^{s} |e_i| \qquad \text{and} \qquad \|e\|_2 = \sqrt{\sum_{i=1}^{s} |e_i|^2}. \tag{A.4}$$

These are special cases of the general family of q-norms, defined by

$$\|e\|_q = \left[\sum_{i=1}^{s} |e_i|^q \right]^{1/q}. \tag{A.5}$$

Note that the max-norm can be obtained as the limit as $q \to \infty$ of the q-norm. (Usually p is used instead of q in defining these norms, but in this book p is often used for the order of accuracy, which might be measured in some q-norm.)

A.3.1 Norm equivalence

With so many different norms to choose from, it is natural to ask whether results on convergence of numerical methods will depend on our choice of norm. Suppose $e(h)$ is the error obtained with some step size h, and that $\|e(h)\| = O(h^p)$ in some norm, so that the method is pth order accurate. Is it possible that the rate will be different in some other norm? The answer is "no," due to the following result on the "equivalence" of all norms on $\mathbb{R}^s$. (Note that this result is valid only as long as the dimension s of the vector is fixed as $h \to 0$. See Section A.5 for an important case where the length of the vector depends on h.)

Let $\|\cdot\|_\alpha$ and $\|\cdot\|_\beta$ represent two different vector norms on $\mathbb{R}^s$. Then there exist two constants C_1 and C_2 such that

$$C_1 \|x\|_\alpha \le \|x\|_\beta \le C_2 \|x\|_\alpha \tag{A.6}$$

for all vectors $x \in \mathbb{R}^m$. For example, it is fairly easy to verify that the following relations hold among the norms mentioned above:

$$\|x\|_\infty \le \|x\|_1 \le s\|x\|_\infty, \tag{A.7a}$$
$$\|x\|_\infty \le \|x\|_2 \le \sqrt{s}\|x\|_\infty, \tag{A.7b}$$
$$\|x\|_2 \le \|x\|_1 \le \sqrt{s}\|x\|_2. \tag{A.7c}$$

Now suppose that $\|e(h)\|_\alpha \le Ch^p$ as $h \to 0$ in some norm $\|\cdot\|_\alpha$. Then we have

$$\|e(h)\|_\beta \le C_2 \|e(h)\|_\alpha \le C_2 C h^p$$

and so $\|e(h)\|_\beta = O(h^p)$ as well. In particular, if $\|e(h)\| \to 0$ in some norm, then the same is true in any other norm and so the notion of "convergence" is independent of our choice of norm. This will *not* be true in Section A.4, where we consider approximating functions rather than vectors.

A.3.2 Matrix norms

For any vector norm $\|\cdot\|$ we can define a corresponding matrix norm. The norm of a matrix $A \in \mathbb{R}^{s\times s}$ is denoted by $\|A\|$ and has the property that $C = \|A\|$ is the *smallest* value of the constant C for which the bound

$$\|Ax\| \le C\|x\| \tag{A.8}$$

holds for *every* vector $x \in \mathbb{R}^s$. Hence $\|A\|$ is defined by

$$\|A\| = \max_{\substack{x\in\mathbb{R}^s \\ x\neq\vec{0}}} \frac{\|Ax\|}{\|x\|} = \max_{\substack{x\in\mathbb{R}^s \\ \|x\|=1}} \|Ax\|. \tag{A.9}$$

It would be rather difficult to calculate $\|A\|$ from the above definitions, but for the most commonly used norms there are simple formulas for computing $\|A\|$ directly from the matrix:

$$\|A\|_1 = \max_{1\le j\le s} \sum_{i=1}^{s} |a_{ij}| \qquad \text{(maximum column sum)}, \tag{A.10a}$$

$$\|A\|_\infty = \max_{1\le i\le s} \sum_{j=1}^{s} |a_{ij}| \qquad \text{(maximum row sum)}, \tag{A.10b}$$

$$\|A\|_2 = \sqrt{\rho(A^T A)}. \tag{A.10c}$$

In the definition of the 2-norm, $\rho(B)$ denotes the spectral radius of the matrix B (the maximum modulus of an eigenvalue). In particular, if A is a normal matrix, e.g., if $A = A^T$ is symmetric, then $\|A\|_2 = \rho(A)$.

We also mention the *condition number* of a matrix in a given norm, defined by

$$\kappa(A) = \|A\|\,\|A^{-1}\|, \tag{A.11}$$

provided the matrix is nonsingular. If the matrix is normal then the 2-norm condition number is the ratio of largest to smallest eigenvalue (in modulus). The condition number plays a role in the convergence rate of many iterative methods for solving a linear system with the matrix A (see Chapter 4). See, e.g., [35], [91] for more discussion.

A.4 Errors in functions

Now consider a problem in which the solution is a function $u(x)$ over some interval $a \le x \le b$ rather than a single value or vector. Some numerical methods, such as finite element or collocation methods, produce an approximate solution $U(x)$ which is also a function. Then the error is given by a function

$$e(x) = U(x) - u(x).$$

We can measure the magnitude of this error using standard function space norms, which are quite analogous to the vector norms described above. For example, the max-norm is given by

$$\|e\|_\infty = \max_{a\le x\le b} |e(x)|. \tag{A.12}$$

The 1-norm and 2-norm are given by integrals over $[a, b]$ rather than by sums over the vector elements:

$$\|e\|_1 = \int_a^b |e(x)|\, dx, \tag{A.13}$$

$$\|e\|_2 = \left(\int_a^b |e(x)|^2\, dx \right)^{1/2}. \tag{A.14}$$

These are again special cases of the general q-norm, defined by

$$\|e\|_q = \left(\int_a^b |e(x)|^q\, dx \right)^{1/q}. \tag{A.15}$$

A.5 Errors in grid functions

Finite difference methods do not produce a function $U(x)$ as an approximation to $u(x)$. Instead they produce a set of values U_i at grid points x_i. For example, on a uniform grid with N equally spaced in some interval (a, b) and grid spacing h, our approximation to $u(x)$ would consist of the N values $(U_1, U_2, \ldots, U_N)$. (Note that if $h = (b-a)/(m+1)$ as is often assumed in this book, then generally $N = m,\ m+1$, or $m+2$, depending on whether one or both boundary points are included in the set of unknowns. For our discussion here this is immaterial—what is important to note is that $N = O(1/h)$ as $h \to 0$.)

How can we measure the error in this approximation? We want to compare a set of discrete values with a function.

We must first decide what the values U_i are supposed to be approximating. Often the value U_i is meant to be interpreted as an approximation to the pointwise value of the function at x_i, so $U_i \approx u(x_i)$. In this case it is natural to define a vector of errors $e = (e_1, e_2, \ldots, e_N)$ by

$$e_i = U_i - u(x_i).$$

This is not always the proper interpretation of U_i, however. For example, some numerical methods are derived using the assumption that U_i approximates the average value of $u(x)$ over an interval of length h, e.g.,

$$U_i \approx \frac{1}{h} \int_{x_{i-1}}^{x_i} u(x)\, dx.$$

In this case it would be more appropriate to compare U_i to this cell average in defining the error. Clearly the errors will be different depending on what definition we adopt and may even exhibit different convergence rates, so it is important to make the proper choice for the method being studied.

Once we have defined the vector of errors $(e_1, \ldots, e_N)$, we can measure its magnitude using some norm. Since this is simply a vector with N components, it would be tempting to simply use one of the vector norms discussed above, e.g.,

$$\|e\|_1 = \sum_{i=1}^{N} |e_i|. \tag{A.16}$$

However, this choice would give a very misleading idea of the magnitude of the error. The quantity in (A.16) can be expected to be roughly N times as large as the error at any single grid point and here N is not the dimension of some physically relevant space, but rather the number of points on our grid. If we refine the grid and increase N, then the quantity (A.16) might well *increase* even if the error at each grid point decreases, which is clearly not the correct behavior.

Instead we should define the norm of the error by discretizing the integral in (A.13), which is motivated by considering the vector $(e_1, \ \ldots, \ e_N)$ as a discretization of some error function $e(x)$. This suggests defining

$$\|e\|_1 = h \sum_{i=1}^{N} |e_i| \tag{A.17}$$

with the factor of h corresponding to the dx in the integral. Note that since $h \approx (b-a)/N$, this scales the sum by $1/N$ as the number of grid points increases, so that $\|e\|_1$ is the average value of e over the interval (times the length of the interval), just as in (A.13). The norm (A.17) will be called a *grid function norm* and is distinct from the related vector norm. The set of values $(e_1, \ \ldots, \ e_N)$ will sometimes be called a *grid function* to remind us that it is a special kind of vector that represents the discretization of a function.

Similarly, the q-norm should be scaled by $h^{1/q}$, so that the q-norm for grid functions is

$$\|e\|_q = \left(h \sum_{i=1}^{N} |e_i|^q \right)^{1/q}. \tag{A.18}$$

Since $h^{1/q} \to 1$ as $q \to \infty$, the max-norm remains unchanged,

$$\|e\|_\infty = \max_{1 \le i \le N} |e_i|,$$

which makes sense from (A.12).

In two space dimensions we have analogous norms of functions and grid functions, e.g.,

$$\|e\|_q = \left(\iint |e(x, y)|^q \, dx \, dy \right)^{1/q} \qquad \text{for functions,}$$

$$\|e\|_q = \left(\Delta x \, \Delta y \sum_i \sum_j |e_{ij}|^q \right)^{1/q} \qquad \text{for grid functions}$$

with the obvious extension to more dimensions.

A.5.1 Norm equivalence

Note that we still have an equivalence of norms in the sense that, for any *fixed* N (and hence fixed h), there are constants C_1 and C_2 such that

$$C_1 \|x\|_\alpha \le \|x\|_\beta \le C_2 \|x\|_\alpha$$

for any vector $e \in \mathbb{R}^N$. For example, translating (A.7a) to the context of grid function norms gives the bounds

$$h\|e\|_\infty \le \|e\|_1 \le Nh\|e\|_\infty = (b-a)\,\|e\|_\infty, \tag{A.19a}$$

$$\sqrt{h}\,\|e\|_\infty \le \|e\|_2 \le \sqrt{Nh}\,\|e\|_\infty = \sqrt{b-a}\,\|e\|_\infty, \tag{A.19b}$$

$$\sqrt{h}\,\|e\|_2 \le \|e\|_1 \le \sqrt{Nh}\,\|e\|_2 = \sqrt{b-a}\,\|e\|_2. \tag{A.19c}$$

However, since these constants may depend on N and h, this equivalence does not carry over when we consider the behavior of the error as we refine the grid so that $h \to 0$ and $N \to \infty$.

We are particularly interested in the convergence rate of a method. If $e(h)$ is the vector of errors obtained on a grid with spacing h, we would like to show that

$$\|e(h)\| \le O(h^q)$$

for some q. In the last section we saw that the rate is independent of the choice of norm if $e(h)$ is a vector in the space $\mathbb{R}^m$ with fixed dimension m. But now $m = N$ and grows as $h \to 0$, and as a result the rate may be quite different in different norms. This is particularly noticeable if we approximate a discontinuous function, as the following example shows.

Example 3.1. Set

$$u(x) = \begin{cases} 0 & x \le \frac{1}{2}, \\ 1 & x > \frac{1}{2} \end{cases}$$

and define the grid function approximation as indicated in Figure A.1. Then the error $e_i(h)$ is zero at all grid points but the one at the discontinuity, where it has the value $1/2$. Then no matter how fine the grid is, there is always an error of magnitude $1/2$ at one grid point and hence

$$\|e(h)\|_\infty = \frac{1}{2} \quad \text{for all } h.$$

On the other hand, in the 1-norm (A.17) we have

$$\|e(h)\|_1 = h/2 = O(h) \quad \text{as } h \to 0.$$

We see that the 1-norm converges to zero as h goes to zero while the max-norm does not.

How should we interpret this? Should we say that $U(h)$ is a first order accurate approximation to $u(x)$ or should we say that it does not converge? It depends on what we

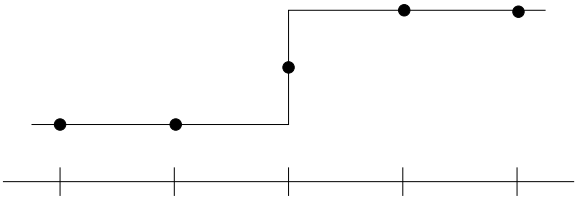

Figure A.1. *The function $u(x)$ and the discrete approximation.*

are looking for. If it is important that the maximum error over all grid points be uniformly small, then the max-norm is the appropriate norm to use and the fact that $\|e(h)\|_\infty$ does not approach zero tells us that we are not achieving our goal. On the other hand, this may not be required, and in fact this example illustrates that it is unrealistic to expect pointwise convergence in problems where the function is discontinuous. For many purposes the approximation shown in Figure A.1 would be perfectly acceptable.

This example also illustrates the effect of choosing a different definition of the "error." If we were to define the error by

$$e_i(h) = U_i - \frac{1}{h}\int_{x_i-h/2}^{x_i+h/2} u(x)\,dx,$$

then we would have $e_i(h) \equiv 0$ for all i and h and $\|e(h)\| = 0$ in every norm, including the max-norm. With this definition of the error our approximation is not only acceptable, it is the best possible approximation.

If the function we are approximating is sufficiently smooth, and if we expect the error to be roughly the same magnitude at all points, then it typically does not matter so much which norm is chosen. The convergence rate for a given method, as observed on sufficiently smooth functions, will often be the same in any q-norm. The norm chosen for analysis is then often determined by the nature of the problem and the availability of mathematical techniques for estimating different error norms. For example, for linear problems where Fourier analysis can be applied, the 2-norm is often a natural choice. For conservation laws where integrals of the solution are studied, the 1-norm is often simplest to use.

A.6 Estimating errors in numerical solutions

When developing a computer program to solve a differential equation, it is generally a good idea to test the code and ensure that it is producing correct results with the expected accuracy. How can we do this?

A first step is often to try the code on a problem for which the exact solution is known, in which case we can compute the error in the numerical solution exactly. Not only can we then check that the error is small on some grid, we can also refine the grid and check how the error is behaving asymptotically, to verify that the expected order of accuracy, and perhaps even error constant, is seen. Of course one must be aware of some of the issues raised earlier, e.g., that the expected order may appear only for h sufficiently small.

It is important to test a computer program by doing grid refinement studies even if the results look quite good on one particular grid. A subtle error in programming (or in deriving the difference equations or numerical boundary conditions) can lead to a program that gives reasonable results and may even converge to the correct solution, but at less than the optimal rate. Consider, for example, the first approach of Section 2.12.

Of course in practice we are usually trying to solve a problem for which we do not know the exact solution, or we wouldn't bother with a numerical method in the first place. However, there are often simplified versions of the problem for which exact solutions are known, and a good place to start is with these special cases. They may reveal errors in the code that will affect the solution of the real problem as well.

This is generally not sufficient, however, even when it is possible, since in going from the easy special case to the real problem new errors may be introduced. How do we estimate the error in a numerical solution if we do not have the exact solution with which to compare it?

The standard approach, when we can afford to use it, is to compute a numerical solution on a very fine grid and use this as a "reference solution" (or "fine-grid solution"). This can be used as a good approximation to the exact solution in estimating the error on other, much coarser, grids. When the fine grid is fine enough, we can obtain good estimates not only for the errors but also for the order of accuracy. See Section A.6.2.

Often we cannot afford to take very fine grids, especially in more than one space dimension. We may then be tempted to use a grid that is only slightly finer than the grid we are testing in order to generate a reference solution. When done properly this approach can also yield accurate estimates of the order of accuracy, but more care is required. See Section A.6.3.

A.6.1 Estimates from the true solution

First suppose we know the true solution. Let $E(h)$ denote the error in the calculation with grid spacing h, as computed using the true solution. In this section we suppose that $E(h)$ is a scalar, typically some norm of the error over the grid, i.e.,

$$E(h) = \|U(h) - \hat{U}(h)\|,$$

where $U(h)$ is the numerical solution vector (grid function) and $\hat{U}(h)$ is the true solution evaluated on the same grid.

If the method is pth order accurate, then we expect

$$E(h) = Ch^p + o(h^p) \quad \text{as } h \to 0,$$

and if h is sufficiently small, then

$$E(h) \approx Ch^p. \tag{A.20}$$

If we refine the grid by a factor of 2, say, then we expect

$$E(h/2) \approx C(h/2)^p.$$

Defining the *error ratio*

$$R(h) = E(h) \,/\, E(h/2), \tag{A.21}$$

we expect

$$R(h) \approx 2^p, \tag{A.22}$$

and hence

$$p \approx \log_2(R(h)). \tag{A.23}$$

Here refinement by a factor of 2 is used only as an example, since this choice is often made in practice. But more generally if h_1 and h_2 are any two grid spacings, then we can estimate p based on calculations on these two grids using

$$p \approx \frac{\log(E(h_1)/E(h_2))}{\log(h_1/h_2)}. \tag{A.24}$$

Hence we can estimate the order p based on any two calculations. (This will be valid only if h is small enough that (A.20) holds, of course.)

Note that we can also estimate the error constant C by

$$C \approx E(h)/h^p$$

once p is known.

A.6.2 Estimates from a fine-grid solution

Now suppose we don't know the exact solution but that we can afford to run the problem on a very fine grid, say, with grid spacing $\bar{h}$, and use this as a reference solution in computing the errors on some sequence of much coarser grids. To compare $U(h)$ on the coarser grid with $U(\bar{h})$ on the fine grid, we need to make sure that these two grids contain coincident grid points where we can directly compare the solutions. Typically we choose the grids in such a way that all grid points on the coarser grid are also fine-grid points. (This is often the hardest part of doing such grid refinement studies—getting the grids and indexing correct.)

Let $\bar{U}(h)$ be the restriction of the fine-grid solution to the h-grid, so that we can define the approximate error $\bar{E}(h) \equiv \|U(h) - \bar{U}(h)\|$, analogous to the true error $E(h) = \|U(h) - \hat{U}(h)\|$. What is the error in this approximate error? We have

$$U(h) - \bar{U}(h) = (U(h) - \hat{U}(h)) + (\hat{U}(h) - \bar{U}(h)).$$

If the method is supposed to be pth order accurate and $\bar{h}^p \ll h^p$, then the second term on the right-hand side (the true error on the $\bar{h}$-grid) should be negligible compared to the first term (the true error on the h-grid) and $\bar{E}(h)$ should give a very accurate estimate of the error.

Warning: Estimating the error and testing the order of accuracy by this approach only confirm that the code is converging to *some* function with the desired rate. It is very possible that the code is converging very nicely to the *wrong* function. Consider a second order accurate method applied to 2-point boundary value problem, for example, and suppose that we code everything properly except that we mistype the value of one of the boundary values. Then a grid-refinement study of this type would show that the method is converging with second order accuracy, as indeed it is. The fact that it is converging to the solution of the wrong problem would not be revealed by this test. One must use other tests as well, not least of which is checking that the computed solutions make sense physically, e.g., that the correct boundary conditions are in fact satisfied.

More generally, a good understanding of the problem being solved, a knowledge of how the solution should behave, good physical intuition, and common sense are all necessary components in successful scientific computing. Don't believe the numbers coming out simply because they are generated by a computer, even if the computer also tells you that they are second order accurate!

A.6.3 Estimates from coarser solutions

Now suppose that our computation is very expensive even on relatively coarse grids, and we cannot afford to run a calculation on a much finer grid to test the order of accuracy.

Suppose, for example, that we are willing to run the calculation only on grids with spacing h, $h/2$, and $h/4$ and we wish to estimate the order of accuracy from these three calculations, without using any finer grids. Since we can estimate the order from any two values of the error, we could define the errors in the two coarser grid calculations by using the $h/4$ calculation as our reference solution. Will we get a good estimate for the order?

In the notation used above, we now have $\bar{h} = h/4$, while $h = 4\bar{h}$ and $h/2 = 2\bar{h}$. Assuming the method is pth order accurate and that h is small enough that (A.20) is valid (a poor assumption, perhaps, if we are using very coarse grids!), we expect

$$\begin{aligned}\bar{E}(h) &= E(h) - E(\bar{h}) \\ &\approx Ch^p - C\bar{h}^p \\ &= (4^p - 1)C\bar{h}^p.\end{aligned}$$

Similarly,

$$\bar{E}(h/2) \approx (2^p - 1)C\bar{h}^p.$$

The ratio of approximate errors is thus

$$\bar{R}(h) = \bar{E}(h)/\bar{E}\left(\frac{h}{2}\right) \approx \frac{4^p - 1}{2^p - 1} = 2^p + 1.$$

For modest p this differs significantly from (A.22). For a first order accurate method with $p = 1$, we now have $\bar{R}(h) \approx 3$ and we should expect the apparent error to decrease by a factor of 3 when we go from h to $h/2$, not by the factor of 2 that we normally expect. For a second order method we expect a factor of 5 improvement rather than a factor of 4. This increase in $\bar{R}(h)$ results from the fact that we are comparing our numerical solutions to another approximate solution that has a similar error.

We can obtain a good estimate of p from such calculations (assuming (A.20) is valid), but to do so we must calculate p by

$$p \approx \log_2(\bar{R}(h) - 1)$$

rather than by (A.23). The approximation (A.23) would overestimate the order of accuracy.

Again we have used refinement by factors of 2 only as an example. If the calculation is very expensive we might want to refine the grid more slowly, using, for example, h, $3h/4$, and $h/2$. One can develop appropriate approximations to p based on any three grids. The tricky part may be to estimate the error at grid points on the coarser grids if these are not also grid points on the $\bar{h}$ grid. Interpolation can be used, but then one must be careful to ensure that sufficiently accurate interpolation formulas are used that the error in interpolation does not contaminate the estimate of the error in the numerical method being studied.

Another approach that is perhaps simpler is to compare the solutions

$$\tilde{E}(h) \equiv U(h) - U(h/2) \quad \text{and} \quad \tilde{E}(h/2) = U(h/2) - U(h/4).$$

In other words, we estimate the error on each grid by using the next finer grid as the reference solution, rather than using the same reference solution for both coarser grids. In this case we have

$$\tilde{E}(h) = E(h) - E\left(\frac{h}{2}\right) \approx C\left(1 - \frac{1}{2^p}\right)h^p$$

and

$$\tilde{E}(h/2) = E\left(\frac{h}{2}\right) - E\left(\frac{h}{4}\right) \approx C\left(1 - \frac{1}{2^p}\right)\frac{h^p}{2^p}$$

and so

$$\tilde{E}(h)/\tilde{E}\left(\frac{h}{2}\right) \approx 2^p.$$

In this case the approximate error decreases by the same factor we would expect if the true solution were used as the reference solution on each grid.