Mathematics 255—January 23, 1996

Choosing step size in numerical methods for solving ordinary differential equations

Estimating accuracy

When you set out to approximate the solution of a differential equation

$$y' = f(x, y)$$

by a numerical method, you are given initial conditions $y(x_0) = y_0$ and a range of values $[x_0, x_f]$ (f for 'final') over which you want to approximate the solution. The particular numerical method you are going to use is usually specified in advance, and your major choice is then to choose the **step size** — the increment by which x changes in each step of the method.

The basic idea is that if you choose a small step size your approximation will be closer to the true solution, but at the cost of a larger number of steps. What is the exact relationship between step size and accuracy? Some explicit data can give you a feel for how things go. Here is the outcome of several runs of Euler's method for solving the equation

$$y' = 2xy - 1, \quad y(0) = 1$$

over the interval $[0, 1]$.

Number of steps	Step size	Estimated value of $y(1)$
4	0.250000	0.426758
8	0.125000	0.540508
16	0.062500	0.608672
32	0.031250	0.646763
64	0.015625	0.667026
128	0.007812	0.677495
256	0.003906	0.682819
512	0.001953	0.685503
1024	0.000977	0.686851

This table suggests that if we keep making the step size smaller, then the approximations to $y(1)$ will converge to something, but does not give a good idea, for example, of how many steps we would need to take if we wanted an answer correct to 4 decimals. To get this we need some idea of how rapidly the estimates are converging. The way to do this most easily is to add a column to record the difference between an estimate and the previous one.

Number of steps	Step size	Estimated value of $y(1)$	Difference from previous estimate
4	0.250000	0.426758	0.426758
8	0.125000	0.540508	0.113751
16	0.062500	0.608672	0.068164
32	0.031250	0.646763	0.038091
64	0.015625	0.667026	0.020263
128	0.007812	0.677495	0.010469
256	0.003906	0.682819	0.005323
512	0.001953	0.685503	0.002685
1024	0.000977	0.686851	0.001348

The pattern in the last column is simple. Eventually, the difference gets cut in half at each line. This means that at least when h is small, the difference is roughly proportional to the step size. Thus we can **extrapolate** entries in the last column to be

$$0.000674, \quad 0.000337, \quad 0.000169, \quad 0.000084, \quad \ldots.$$

But the total error should be the sum of all these differences, which can be written as

$$0.001348\,(1/2 + 1/4 + 1/8 + 1/16 + \ldots) = 0.001348!$$

It is not in fact necessary to look at a sequence of runs. We can summarize the discussion more simply: *if we make two runs of Euler's method, the error in the second of the two estimates for $y(x_f)$ will be* (roughly) *the size of the difference between the two estimates.*

The reason the error can be estimated in this way is because *the error in Euler's method is proportional to the step size.* Suppose we have made two runs of the method, one with step size h and another with step size $h/2$. Then because of the proportionality, and the meaning of error, we know that

$$\begin{aligned}
\textit{true answer} &\simeq \textit{estimate}(h) \quad + Ch \\
\textit{true answer} &\simeq \textit{estimate}(h/2) + C(h/2)
\end{aligned}$$

for some constant C. Here *estimate*(h) means the estimate corresponding to step size h. If we treat the approximate equalities as equalities, we get two equations in the two unknowns *true answer* and C. We can solve them to get

$$\begin{aligned}
\textit{true answer} &\simeq 2\ \textit{estimate}(h/2) - \textit{estimate}(h) \\
&= \textit{estimate}(h/2) + (\textit{estimate}(h/2) - \textit{estimate}(h)) \\
\textit{error in estimate}(h/2) &\simeq \textit{estimate}(h/2) - \textit{estimate}(h)
\end{aligned}$$

In other words, *one run of Euler's method will tell us virtually nothing about how accurate the run is, but two runs will allow us to get an idea of the error and an improved estimate for the value of $y(x_f)$ as well.*

The same is true for any of the other methods, since we know that in each case the error is roughly proportional to some power of h — for Improved Euler's it is h^2 and for $RK4$ it is h^4. In general, if the error is proportional to h^k and we make two runs of step size h and $h/2$ we get

$$\begin{aligned}
\textit{true answer} &\simeq \textit{estimate}(h) + Ch^k \\
\textit{true answer} &\simeq \textit{estimate}(h/2) + C(h/2)^k \\
2^k\ \textit{true answer} &\simeq 2^k\ \textit{estimate}(h/2) + Ch^k
\end{aligned}$$

and subtract getting

$$\begin{aligned}
(2^k - 1)\ \textit{true answer} &\simeq 2^k\ \textit{estimate}(h/2) - \textit{estimate}(h) \\
\textit{true answer} &\simeq \frac{2^k\ \textit{estimate}(h/2) - \textit{estimate}(h)}{2^k - 1} \\
&= \textit{estimate}(h/2) + \frac{\textit{estimate}(h/2) - \textit{estimate}(h)}{2^k - 1}
\end{aligned}$$

and in addition

$$\textit{error in estimate}(h/2) \simeq \frac{\textit{estimate}(h/2) - \textit{estimate}(h)}{2^k - 1}$$

The earlier formula is the special case of this with $k = 1$.

Step size choice

When we use one of the numerical methods to approximate the solution of the differential equation, we usually have ahead of time some idea of how accurate we want the approximation to be — to within 6 or 7 decimals of accuracy, say. The question is, how can we choose the step size in order to obtain this accuracy? The discussion above suggests the following procedure: (1) Make two runs of the method over the given range, with step size h and then with step size $h/2$. We get two estimates for $y(x_f)$, $estimate(h)$ and $estimate(h/2)$. If the method has error proportional to h^k (i.e. if the error is of order k) then we know that the error in $estimate(h)$ is approximately

$$\textit{error in estimate}(h) \simeq \frac{\textit{estimate}(h) - \textit{estimate}(h/2)}{1-2^{-k}}$$

If we multiply the step size by α we multiply the error by α^k, so if we want an error of about ϵ we must solve an equation to get the new step size h_ϵ.

$$\left(\frac{h_\epsilon}{h}\right)^k = \frac{\epsilon}{\textit{error in estimate}(h)}$$
$$h_\epsilon = h\left(\frac{\epsilon}{\textit{error in estimate}(h)}\right)^{1/k}$$

The procedure looks tedious, but plausible: we choose some initial step size h which we judge (arbitrarily) to be reasonable. We do two runs at step sizes h and $h/2$ to see how accuracy depends on step size (to get the constant of proportionality). We then calculate the step size h needed to get the accuracy we want, and do a third run at the new size.

Doing separate runs is work. It would be better if we could adjust the step size during the course of one run so as to make the single run as accurate as we want. This is possible, if *we use the above method to estimate accuracy in every step.*

In other words, in every step we do one step at some size h and two successive steps of size $h/2$. We compare the answers to estimate the error in the second approximation (which is the best one we have). If it is unacceptably high, we adjust the step size down to get something acceptable. One thing that makes this a reasonable procedure is that once we settle on an acceptable step size, it will not in fact have to be readjusted very often.

There is a hidden bonus to this plan. There are two ways in which the calculations might go wrong. The first is that the step size might be too large, and the approximations too inaccurate. The second is that the step size might be too small, and we are doing too many calculations. *If we keep a running estimate of local errors, then we can adjust h up or down as we go along, to take care of both of these problems.*

To get the details straight, I have to introduce a new concept. We usually want not just the final estimate for $y(x_f)$ to be accurate, but also for all the $y(x)$ in between. In every step a certain error is made, and as the calculation progresses the error accumulates. Thus the final error should be thought of as the accumulation of the errors made in each step. What we want to control, therefore, is the *rate* of error accumulation. If it is 0.0001 across each unit interval, the error across a range 0 - 10 will be roughly 10 times 0.0001 = 0.001. If I make an error of δ in a single step then I can expect to make an error of $n\delta$ in n steps. So I define the **error rate** to be the rate at which errors are made. If we make an error of e in one step of size h then the error rate is e/h. It is as an *acceptable error rate* that I will specify the accuracy wanted.

Here is the procedure in more detail: We have variables x, y, h, x_f, and *acceptable error rate*. We start with (1) x, y at their initial values, as well as (2) some specification of the acceptable error rate and (3) a first guess for a suitable step size h in the very first step. We will loop repeatedly, increasing x and changing y in steps of size which vary according to how well things are going. In each loop we will call a subroutine to advance x and y and set h to be the step size to be tried at the next advance. We stop when x reaches x. I should mention that there are a few technical problems here to be dealt with: (a) how to choose an initial value for h, when we start with so little information? (b) since the step sizes are not uniform, we have to be careful to hit x_f on the button instead of passing it by. At any rate, the advance is made in a subroutine which changes the values of x, y, and h. Inside the subroutine the proposed size for h is tried out, doing one step of size h and two of size $h/2$ necessary in order to estimate the error and the error rate. It then loops, adjusting the step size as I have suggested above, until the error rate is acceptable, and a new value of h for the next call of the subroutine is proposed. The value of x is advanced by the step size used (which is not necessarily the same as the one proposed for the next advance), and y is set to the presumed *true answer* given by formulas above. (Why not use the best guess we can make?)

Here is the outline of a subroutine to do the job.

```
advance(x, y, h)

begin
  repeat
    h_used = h
    y1 = value of y obtained by carrying out one step of the method
         with step size h
    y2 = value obtained by carrying out two successive steps of size h/2
    error = (y2 - y1)/appropriate factor
    error_rate = absolute value of (error/h)
    ratio = error_rate/acceptable_rate
    if ratio > 0 then h = (safety factor)*h/(appropriate root of ratio)
    else h = 2*h
  until ratio <= 1
  x = x + h_used
  y = y2 + error
end
```

There is one technical item in this that may cause trouble, and that is the choice of `appropriate factor`. The one I suggest is that for the estimate `y2`, since the values we are calculating for y are at least as accurate as `y2`.

Another item is the `safety factor`. In practice, the formula above for the proposed step size is a little sharp to produce stable behaviour, so we use a value slightly smaller:

$$\textit{proposed } h = (\textit{safety factor})\ h \left(\frac{\textit{acceptable error rate}}{\textit{current error rate}}\right)^{1/k}$$

where the safety factor should be somewhat smaller than 1.0. I use 0.9.