Step size for numerical differential equations

John D. Cook*

February 22, 2008

Numerical methods for solving ordinary differential equations depend on a step size h. How small should h be? If it weren't for finite precision arithmetic, the answer would be "as small as possible" since the truncation error goes to zero as h goes to zero, at least for "nice" problems. Step size would be limited only by the number of steps we have time to take.

However, as the step size decreases and the number of steps increases, arithmetic error also increases. Thus in practice, h should not be too small. In realistic problems, no one knows an exact solution to compare the computed solution to, and error estimates are often impractical.

One approach is to solve the equation using smaller and smaller steps, comparing the solutions to see if they are converging. The following method is based on this idea, but it provides an objective criterion for knowing when to stop.

Suppose we are solving $u' = f(t, u)$ on $[0, T]$ and we are using a method of order p, *i.e.,* the error in our approximation for $u(T)$ is bounded by some constant times h^p. For example, $p = 1$ for Euler's method and $p = 4$ for the fourth order Runge-Kutta method. Solve the equation using 2^N steps, so $h = 2^{-N}T$ and let u_N denote this solution.

Assume that not only is the error *bounded* by a multiple of h^p but that for small enough h the error is approximately *equal* to a constant times h^p. That is, assume

$$u(T) - u_N(T) \approx ch^p.$$

Let

$$\begin{aligned} D_N &= u_N(T) - u_{N-1}(T) \\ &\approx (u(T) - c(2^{-N}T)^p) - (u(T) - c(2^{-(N-1)}T)^p) \\ &= 2^{-pN}(2^p - 1)cT^p \end{aligned}$$

Thus

$$\frac{D_{N-1}}{D_N} \approx \frac{2^{-pN+p}(2^p-1)cT^p}{2^{-pN}(2^p-1)cT^p} = 2^p$$

and so

$$R_N \equiv \frac{\ln |D_{N-1}/D_N|}{\ln 2} \approx p.$$

*http://www.JohnDCook.com

The deviation between R_N and p gives us a measure of how the method is converging. Typically, R_N gets closer to p initially, stays near p while the method is becoming more accurate, and then deviates from p as the error starts increasing due to accumulated arithmetic error. In this case, we pick h to be the smallest h for which R_N is close to p, or maybe the next smaller h. If R_N never gets close to p, this tells us that arithmetic error began dominating the solution before the method started to converge. If this happens, we are warned that our method will not work well on that equation.

Reference: *Differential Equations: A Dynamical Systems Approach, Part I* by John Hubbard and Beverly West. Springer-Verlag 1991.