Error Analyses for the Euler and Runge-Kutta Methods

Error in the Euler Method: The idea behind the Euler method is clearly discussed and illustrated in the textbook. For an ODE $\frac{dy}{dx} = f(x,y)$ with initial condition $y(x_0) = y_0$, successive points in the approximation for a step size of h are found using

$$x_n = x_{n-1} + h$$
$$y_n = y_{n-1} + hf\left(x_{n-1}, y_{n-1}\right).$$

The cumulative error in this method is

$$\left|y(x_n) - y_n\right| \leq Ch, \quad \text{for some constant } C.$$

So if we cut the step size, h, in half, this will cut the error bound, Ch, in half. Sometimes a specific error bound can be computed using

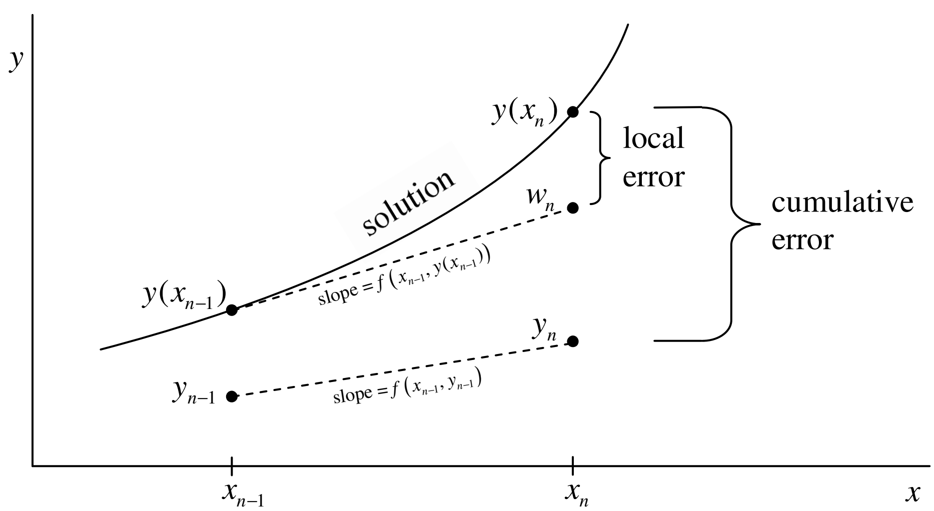
$$\text{Error} \leq \frac{Mh}{2} He^{LH}$$

where

$$| y''(x) | \leq M,$$
$$| f_y(x,y) | \leq L, \text{ and}$$
$$H = x_n - x_0$$

In order to see why this is the case, recall that the errors compound for two different reasons. First, we generate each estimate by moving along a tangent line rather than an actual solution curve. The degree to which this generates errors is determined by the rate at which the solutions deviate from the tangent, thus the need to bound $y''(x)$. Second, once we are off of the desired solution we are plugging y_n into the ODE to determine the next slope instead of $y(x_n)$. That is, we are moving along a tangent line with the wrong slope. The degree to which the slope at these two y-values can differ is determined by the size of $f_y(x,y)$. Both of these errors are repeated n times.

Consider the following diagram illustrating these two sources of error at the n^{th} step of the Euler method:



In this diagram w_n is the height we would obtain by applying the Euler method to the actual point on the solution $(x_n, y(x_n))$ and y_n is the height obtained by the Euler method from the previous approximation point. Thus

$$\begin{aligned} w_n &= y(x_{n-1}) + hf\left(x_{n-1}, y(x_{n-1})\right) \\ y_n &= y_{n-1} + hf\left(x_{n-1}, y_{n-1}\right) \end{aligned} \tag{1}$$

The cumulative error may then be broken into two parts

$$\left|y(x_n) - y_n\right| \le \left|y(x_n) - w_n\right| + \left|w_n - y_n\right| \tag{2}$$

Where $\left|y(x_n) - w_n\right|$ is the local error, i.e., how far off we would be had we started at the correct point on the previous step. By Taylor's theorem and the definition of w_n in (1),

$$\begin{aligned} y(x_n) &= y(x_{n-1}) + hy'(x_{n-1}) + \tfrac{1}{2}h^2 y''(c_n) \\ &= w_n + \tfrac{1}{2}h^2 y''(c_n) \end{aligned}$$

where c_n is between x_{n-1} and x_n. Thus the local error is

$$\left|y(x_n) - w_n\right| \le \tfrac{1}{2}h^2 \left|y''(c_n)\right| \le \tfrac{1}{2}Mh^2. \tag{3}$$

Now we want to see how this local error compounds. Using the definitions of w_n and y_n from (1), the remaining portion of the error is

$$\begin{aligned} \left|w_n - y_n\right| &= \left|y(x_{n-1}) + hf\left(x_{n-1}, y(x_{n-1})\right) - y_{n-1} - hf\left(x_{n-1}, y_{n-1}\right)\right| \\ &\le \left|y(x_{n-1}) - y_{n-1}\right| + h\left|f\left(x_{n-1}, y(x_{n-1})\right) - f\left(x_{n-1}, y_{n-1}\right)\right| \end{aligned} \tag{4}$$

Applying Taylor's theorem to $f\left(x_{n-1}, y\right)$ where x_{n-1} is held constant and *y* is allowed to vary between y_{n-1} and $y(x_{n-1})$, we obtain

$$f\left(x_{n-1}, y(x_{n-1})\right) = f\left(x_{n-1}, y_{n-1}\right) + \left(y(x_{n-1}) - y_{n-1}\right) f_y\left(x_{n-1}, d_n\right) \tag{5}$$

where d_n is between y_{n-1} and $y(x_{n-1})$. Thus combining (4) and (5),

$$\begin{aligned} \left|w_n - y_n\right| &\le \left|y(x_{n-1}) - y_{n-1}\right| + h\left|y(x_{n-1}) - y_{n-1}\right|\left|f_y\left(x_{n-1}, d_n\right)\right| \\ &\le \left|y(x_{n-1}) - y_{n-1}\right|\left(1 + hL\right). \end{aligned} \tag{6}$$

Putting the two parts of error back together from (2) using the bounds from (3) and (6), the total error is

$$\left|y(x_n) - y_n\right| \le \tfrac{1}{2}Mh^2 + \left|y(x_{n-1}) - y_{n-1}\right|\left(1 + hL\right).$$

Now we have a recursive relationship on the error at each step. Applying the same inequality to the previous step, for example, would yield

$$\left|y(x_n) - y_n\right| \le \tfrac{1}{2}Mh^2 + \tfrac{1}{2}Mh^2\left(1 + hL\right) + \left|y(x_{n-2}) - y_{n-2}\right|\left(1 + hL\right)^2$$

Repeating this until we reach the starting point where $y(x_0) - y_0 = 0$, we get

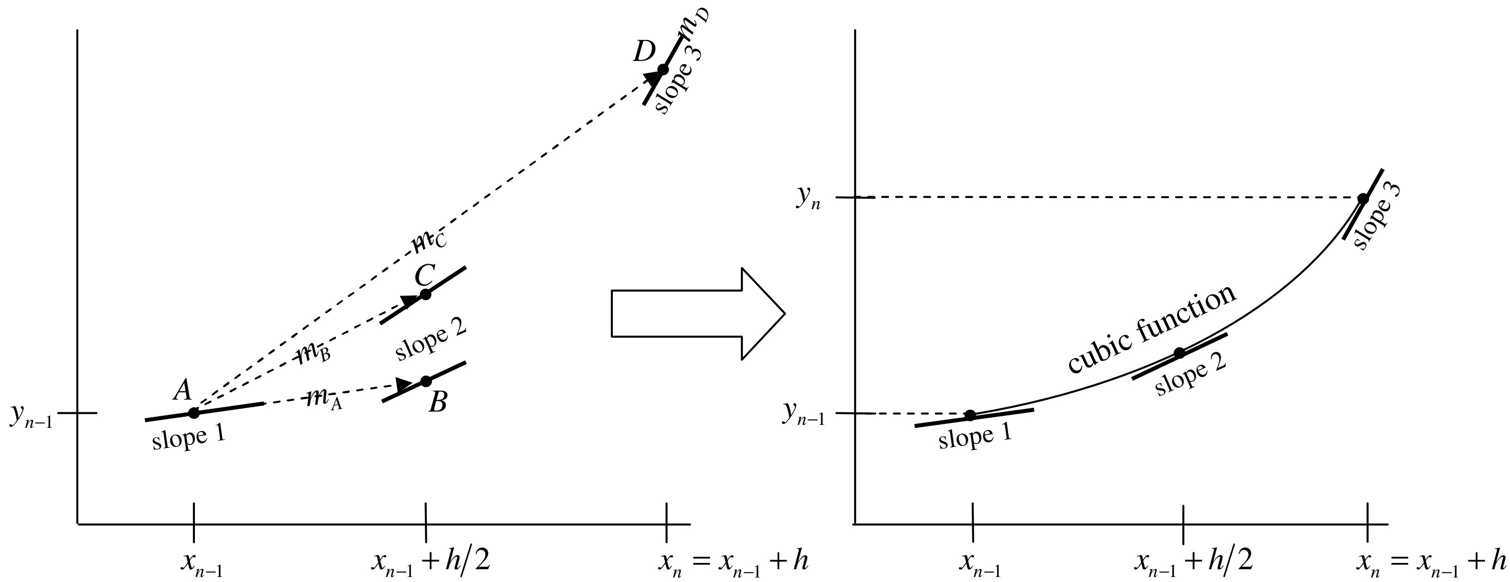
$$\left|y(x_n) - y_n\right| \le \tfrac{1}{2}Mh^2 \sum_{k=0}^{n-1} \left(1 + hL\right)^k.$$

Since $1 + hL \le e^{Lh}$,

$$\left|y(x_n) - y_n\right| \le \tfrac{1}{2}Mh^2 \sum_{k=0}^{n-1} e^{Lhk} \le \tfrac{1}{2}Mh^2 \sum_{k=0}^{n-1} e^{Lhn} = \tfrac{1}{2}Mh^2 n e^{Lhn} = \frac{Mh}{2} He^{LH}.$$

Improved Euler Method: The improved Euler method uses an average of two slopes to generate the tangent line approximation. This is a second-order method, meaning the error is bounded by Ch^2 for some constant C. This is covered well in the book. Give particular effort to understanding Figure 2.5.3 which illustrates the underlying construction of the improved Euler method. You do not need to memorize the algorithm.

Runge-Kutta Idea: At each step, the Runge-Kutta method of order four, uses four slopes, one calculated at the left-hand endpoint of the step, one calculated at the right-hand endpoint of the step, and two calculated at the midpoint of the step. The two at the midpoint are averaged producing a sequence of three slopes equally spaced from x_{n-1} to $x_{n-1}+h/2$ to $x_n = x_{n-1}+h$. Then the height for the next step, y_n, is determined by starting at the point $\left(x_{n-1}, y_{n-1}\right)$ and following a cubic function that matches the three slopes at these equally spaced points (see diagram).



Runge-Kutta Method: Given the ODE $\dfrac{dy}{dx} = f(x,y)$ with initial condition $y(x_0) = y_0$, the Runge-Kutta approximations y_n to the actual solution values $y(x_n)$ using step size of h are generated recursively by

$$x_n = x_{n-1} + h$$
$$k_1 = f(x_{n-1}, y_{n-1})$$
$$k_2 = f(x_{n-1} + \frac{1}{2}h, y_{n-1} + \frac{1}{2}hk_1)$$
$$k_3 = f(x_{n-1} + \frac{1}{2}h, y_{n-1} + \frac{1}{2}hk_2)$$
$$k_4 = f(x_n, y_{n-1} + hk_3)$$
$$y_n = y_{n-1} + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

You do not need to memorize this algorithm.

Error in the Runge-Kutta Method: The Runge-Kutta method shown here is a fourth-order method, so that the cumulative error is

$$\text{cumulative error} = |y(x_n) - y_n| \le Ch^4, \text{ for some constant } C.$$

Thus if we cut the step size, h, in half, this will cut the error bound, Ch^4, by a factor of ***one sixteenth***, $\frac{1}{16} = \left(\frac{1}{2}\right)^4$. This error bound results in a more rapid decrease in the magnitude of errors when the step size is reduced than occurs in the Euler method.

It is theoretically possible to formulate a specific error bound of the form

$$\text{cumulative error} \le RHe^{LH} \bullet h^4$$

where L and H are the same as for the bound on the Euler method (that is, $|f_y(x, y)| \le L$ and $H = x_n - x_0$), but unfortunately, R is a much more complicated constant depending on the values of f, all of the fourth-order partial derivatives of f, and the fifth order partial derivative of f with respect to x. In general, R is not possible or practical to compute.

Instead, a common technique is to repeatedly halve the step size, h, until successive approximations are within the desired accuracy. We then assume (hope) that the actual value is also within that degree of accuracy of the last approximation.

Adaptive Step Size: An even better technique is to do something to monitor the potential size of the error at each step and adjust the step size along the way. Such an approach, using variable step sizes, is called an adaptive method. A simple example of an adaptive method is to repeatedly halve the step size as in the previous bullet, but to do this for each step to determine a good for that step only. Specifically you compute a *single step* using step size h then again using two steps of size $h/2$. This results in two approximations y_n and $\tilde{y}_n$, respectively. Since the method is order 4,

$$y(x_n) = y_n + C_1 h^4$$

and

$$y(x_n) = \tilde{y}_n + C_2 \left(\frac{h}{2}\right)^4$$

Although it is not necessarily true that the constants C_1 and C_2 are the same, in most cases for small step sizes, they will be very similar. Then $C_2 = C_1 \approx C$, and subtracting the two equations, we obtain

$$0 \approx y_n - \tilde{y}_n + Ch^4 - C\left(\frac{h}{2}\right)^4$$

or

$$\tilde{y}_n - y_n \approx \frac{15}{16} Ch^4 .$$

Then an estimate for this local error is

$$\text{local error} = Ch^4 \approx \frac{16}{15}(\tilde{y}_n - y_n) .$$

One may use this estimate to adjust the step size h until an acceptable estimate for the local error is achieved. Just as in the Euler method, the local error compounds since subsequent slopes are computed at slightly incorrect points and the process is repeated n times. In fact, it is precisely the He^{LH} factor in the error bound for both the Euler method and the Runge-Kutta method that accounts for both aspects of this compounding. Thus, if ε is the total allowable error at the final step, then requiring

$$\tilde{y}_n - y_n \leq \frac{\varepsilon}{He^{LH}}$$

Will generally ensure that the local error not compound beyond ε.

The MATLAB routine *ode45* uses an adaptive Runge-Kutta method with a slightly more complex method for controlling the step size. We will be using this routine later in the course.

Roundoff Error: The total error is composed of the sort of error we have been discussing that is due to differences between the exact solution and Runge-Kutta method approximations. This error is called the *cumulative error* in the textbook and is sometimes also called the *truncation error*. There are also errors due to rounding in the computer that occur every time any computation is performed. Since the number of computations is proportional to the number of steps n, the *roundoff error* will be

$$\text{roundoff error} \approx \frac{B}{h}, \text{ for some constant } B.$$

Thus as h decreases, the roundoff error increases. This means we have to find a step size h which is small enough to make the truncation error small, but big enough to keep the roundoff error small. Since the *total error* is the sum of the truncation error and the roundoff error, the graphs of B/h and Ch^4 gives an image of what we are looking for: