Error Estimates for Euler-like Methods

We have seen that n^{th} order Taylor methods have error estimates of $O(h^n)$ for steps of size h. The problem with this information is that it only tells us that the error we make in a single step of the method is proportional to h^n. However, without knowledge of the constant of proportionality we can not compute an absolute error estimate.

The following method demonstrates how we can develop a more concrete error estimate.

Consider an n^{th} order Euler-like estimate

$$y_{i+1} \approx y_i + h\,\varphi(t_i, h, y_i)$$

We can express the error in this estimate by introducing a truncation error function $\tau_{i+1}(h)$ and writing

$$y(t_{i+1}) = y_i + h\,\varphi(t_i, h, y_i) + h\,\tau_{i+1}(h) \tag{1}$$

If we know that the truncation error is $O(h^n)$, all we can say at this point is that

$$\tau_{i+1}(h) = K\,h^n$$

Without specific knowledge of the constant of proportionality K we can not compute the absolute size of the error. Hence, we have no way of knowing whether or not the step size h that we have chosen is small enough to produce the desired error.

Here is a clever method to estimate the size of the actual error. Suppose we have at our disposal a second Euler-like estimate

$$\tilde{y}_{i+1} \approx y_i + h\,\tilde{\varphi}(t_i, h, y_i)$$

of order h^{n+1} with truncation error term $\tilde{\tau}_{i+1}(h)$. This allows us to write

$$y(t_{i+1}) = y_i + h\,\tilde{\varphi}(t_i, h, y_i) + h\,\tilde{\tau}_{i+1}(h) \tag{2}$$

Introduce $w_i = y_i$ and terms

$$w_{i+1} = w_i + h\,\varphi(t_i, h, y_i)$$

$$\tilde{w}_{i+1} = w_i + h\,\tilde{\varphi}(t_i, h, y_i)$$

In terms of these new variables equations (1) and (2) become

$$y(t_{i+1}) = w_{i+1} + h\,\tau_{i+1}(h)$$

$$y(t_{i+1}) = \tilde{w}_{i+1} + h\,\tilde{\tau}_{i+1}(h)$$

Setting these two equations equal to each other and solving for $\tau_{i+1}(h)$ gives

$$\tau_{i+1}(h) = \frac{\tilde{w}_{i+1} - w_{i+1}}{h} + \tilde{\tau}_{i+1}(h)$$

Noting that the higher order error estimate $\tilde{\tau}_{i+1}(h)$ is a factor of h smaller than $\tau_{i+1}(h)$ we see that most of the error must come from the first term on the right. This leads to a concrete estimate for the error term $\tau_{i+1}(h)$:

$$|\tau_{i+1}(h)| \approx \frac{|\tilde{w}_{i+1} - w_{i+1}|}{h}$$

Adjusting the step size to hit a required error tolerance

Now that we have a more concrete error estimate to work with we can use it to adjust our step size h to have the error fall below a desired tolerance ε. Our main tool for adjusting the error up or down is to manipulate the step size. Specifically, we can ask what happens to the error term when we replace the original step size h with a new step size $q\, h$. Since we know that the method we are using is an $O(h^n)$ method we can see that

$$|\tau_{i+1}(q\,h)| \approx \left|K\,(q\,h)^n\right| = q^n\,|K\,h^n| \approx q^n\,|\tau_{i+1}(h)| \approx q^n\,\frac{|\tilde{w}_{i+1} - w_{i+1}|}{h}$$

Thus to force the error term below the desired tolerance ε we need

$$q^n\,\frac{|\tilde{w}_{i+1} - w_{i+1}|}{h} < \varepsilon$$

or

$$q < \left(\frac{h\,\varepsilon}{|\tilde{w}_{i+1} - w_{i+1}|}\right)^{1/n}$$

The Runge-Kutta-Fehlberg method

To apply the strategy above we need to two multi-step methods, a first method of order n and a second method of order n+1. One approach would be to use the Runge-Kutta method of order 4 in combination with a method of order 5. This would work, but would typically require a total of 9 function evaluations per step. The Runge-Kutta-Fehlberg method uses an order 4 method in combination with an order 5 method, where the two methods are cleverly chosen to reuse function evaluations between the two methods. Specifically, we use a four stage method

$$w_{i+1} = w_i + \frac{25}{216}\,k_1 + \frac{1408}{2565}\,k_3 + \frac{2197}{4104}\,k_4 - \frac{1}{5}\,k_5$$

and a five stage method

$$\tilde{w}_{i+1} = w_i + \frac{16}{135}\,k_1 + \frac{6656}{12825}\,k_3 + \frac{28561}{56430}\,k_4 - \frac{9}{50}\,k_5 + \frac{2}{55}\,k_6$$

where

$$k_1 = h\, f(t_i\,,\, w_i)$$

$$k_2 = h\, f(t_i + \frac{1}{4}\, h\,,\, w_i + \frac{1}{4}\, k_1)$$

$$k_3 = h\, f(t_i + \frac{3}{8} h\,,\, w_i + \frac{3}{32}\, k_1 + \frac{9}{32}\, k_2)$$

$$k_4 = h\, f(t_i + \frac{12}{13}\, h\,,\, w_i + \frac{1932}{2197}\, k_1 - \frac{7200}{2197}\, k_2 + \frac{7296}{2197}\, k_3)$$

$$k_5 = h\, f(t_i + h\,,\, w_i + \frac{439}{216}\, k_1 - 8\, k_2 + \frac{3680}{513} k_3 - \frac{845}{4104}\, k_4)$$

$$k_6 = h\, f(t_i + \frac{1}{2}\, h\,,\, w_i - \frac{8}{27}\, k_1 + 2\, k_2 - \frac{3544}{2565}\, k_3 + \frac{1859}{4104}\, k_4 - \frac{11}{40}\, k_5)$$

Note that these method share a lot of stages, which reduces the total work needed.

To apply our error correction method, we first compute the error estimate

$$r = \frac{\left|\tilde{w}_{i+1} - w_{i+1}\right|}{h} = \frac{1}{h} \left| \frac{1}{360}\, k_1 - \frac{128}{4275}\, k_3 - \frac{2197}{75240}\, k_4 + \frac{1}{50}\, k_5 + \frac{2}{55}\, k_6 \right|$$

If the error term falls below our required tolerance ε we return the four stage estimate

$$w_{i+1} = w_i + \frac{25}{216}\, k_1 + \frac{1408}{2565}\, k_3 + \frac{2197}{4104}\, k_4 - \frac{1}{5}\, k_5$$

If the error r is too large, we replace the step size h with a smaller step size $q\, h$ where

$$q = (\varepsilon / r)^{1/4}$$

or a more conservative factor of

$$q = \left(\frac{\varepsilon}{2\, r}\right)^{1/4} = 0.84\, (\varepsilon / r)^{1/4}$$

and then redo the calculation. (Even if the first calculation is within the required tolerance, we can do better on the next iteration by computing q anyway and using a step of size $q\, h$ for the next iteration.)