Richardson Extrapolation

There are many approximation procedures in which one first picks a step size h and then generates an approximation $A(h)$ to some desired quantity A. Often the order of the error generated by the procedure is known. In other words

$$A = A(h) + Kh^k + O(h^{k+1}) \tag{1}$$

with k being some known constant, K being some other (probably unknown) constant and $O(h^{k+1})$ designating any function that is bounded by a constant times h^{k+1} for h sufficiently small. For example, A might be the value $y(t_f)$ at some final time t_f for the solution to an initial value problem $y' = f(t, y),\ y(t_0) = y_0$. Then $A(h)$ might be the approximation to $y(t_f)$ produced by Euler's method with step size h. In this case $k = 1$. If the improved Euler's method is used $k = 2$. If Runge-Kutta is used $k = 4$.

If we were to drop the, hopefully tiny, term $O(h^{k+1})$ from equation (1), we would have one linear equation in the two unknowns A, K. We can get a second such equation just by using a different step size. Then the two equations may be solved, yielding approximate values of A and K. This approximate value of A constitutes a new improved approximation, $B(h)$, for the exact A. We do this now. Taking 2^k times

$$A = A(h/2) + K(h/2)^k + O(h^{k+1}) \tag{2}$$

and subtracting equation (1) gives

$$\begin{aligned}\left(2^k - 1\right) A &= 2^k A(h/2) - A(h) + O(h^{k+1}) \\ A &= \frac{2^k A(h/2) - A(h)}{2^k - 1} + O(h^{k+1})\end{aligned}$$

Hence if we define

$$B(h) = \frac{2^k A(h/2) - A(h)}{2^k - 1} \tag{3}$$

then

$$A = B(h) + O(h^{k+1}) \tag{4}$$

and we have generated an approximation whose error is of order $k + 1$, one better than $A(h)$'s. One widely used numerical integration algorithm, called Romberg integration, applies this formula repeatedly to the trapezoidal rule.

Similarly, by subtracting equation (2) from equation (1), we can find K.

$$\begin{aligned}0 &= A(h) - A(h/2) + Kh^k \left(1 - \tfrac{1}{2^k}\right) + O(h^{k+1}) \\ K &= \frac{A(h/2) - A(h)}{h^k \left(1 - \frac{1}{2^k}\right)} + O(h^{k+1})\end{aligned}$$

Once we know K we can estimate the error in $A(h/2)$ by

$$\begin{aligned} E(h/2) &= A - A(h/2) \\ &= K(h/2)^k + O(h^{k+1}) \\ &= \frac{A(h/2) - A(h)}{2^k - 1} + O(h^{k+1}) \end{aligned}$$

If this error is unacceptably large, we can use

$$E(h) \cong Kh^k$$

to determine a step size h that will give an acceptable error. This is the basis for a number of algorithms that incorporate automatic step size control.

Note that $\frac{A(h/2)-A(h)}{2^k-1} = B(h) - A(h/2)$. One cannot get a still better guess for A by combining $B(h)$ and $E(h/2)$.

Example

$A = y(1) = 64.897803$ where $y(t)$ obeys $y(0) = 1,\ y' = 1 - t + 4y$.

$A(h)$ =approximate value for $y(1)$ given by improved Euler with step size h.

$B(h) = \frac{2^k A(h/2)-A(h)}{2^k-1}$ with $k = 2$.

h	$A(h)$	%	#	$B(h)$	%	#
.1	59.938	7.6	20	64.587	.48	60
.05	63.424	2.3	40	64.856	.065	120
.025	64.498	.62	80	64.8924	.0083	240
.0125	64.794	.04	160			

The "%" column gives the percentage error and the "#" column gives the number of evaluations of $f(t, y)$ used.