RKF45 Method

- The famous Runge-Kutta-Fehlberg scheme assumes the following values:

$$
\begin{array}{c|cccccc}
0 & 0 & & & & & \\
\frac{1}{4} & \frac{1}{4} & & & & & \\
\frac{3}{8} & \frac{3}{32} & \frac{9}{32} & & & & \\
\frac{12}{13} & \frac{1932}{2197} & -\frac{7200}{2197} & \frac{7296}{2197} & & & \\
1 & \frac{439}{216} & -8 & \frac{3680}{513} & -\frac{845}{4104} & & \\
\frac{1}{2} & -\frac{8}{27} & 2 & -\frac{3544}{4104} & \frac{1859}{4104} & -\frac{11}{40} & \\
\hline
 & \frac{25}{216} & 0 & \frac{1408}{2565} & \frac{2197}{4104} & -\frac{1}{5} & \\
\hline
 & \frac{16}{135} & 0 & \frac{6656}{12825} & \frac{28561}{56430} & -\frac{9}{5} & \frac{2}{55}
\end{array}
$$

  ◇ RKF45 is indeed made of *two* Runge-Kutta methods.

    ▷ The first one is a 5-stage method that computes

$$\hat{y}_{n+1} = y_n + \frac{25}{216}k_1 + \frac{1408}{2565}k_3 + \frac{2197}{4104}k_4 - \frac{1}{5}k_5.$$

    ▷ The second one is a 6-stage method that computes

$$y_{n+1} = y_n + \frac{16}{135}k_1 + \frac{6656}{12825}k_3 + \frac{28561}{56430}k_4 - \frac{9}{5}k_5 + \frac{2}{55}k_6.$$

  ◇ Note that the same k values are used for both methods.

Error Estimator

- The reason we want to do *two* methods together is because they provide us an *almost free* error estimator:

$$E = \frac{1}{36}k_1 - \frac{128}{4275}k_3 - \frac{2197}{75240}k_4 + \frac{1}{50}k_5 + \frac{2}{55}k_6.$$

- Don't panic if you know where this error estimator comes from. It is the result of lots of research.

- The idea of using the error estimator goes as follows:

  ◇ It *predicts* what error at the current step is being made.

  ◇ If the error is too *big*, then give up the current value of y_{n+1}. Cut the step size, go back to the previously accepted y_n, and recompute a new y_{n+1} (at a different x_{n+1} because step size has been changed.)

  ◇ If the error is acceptable, then use the estimator to make a *conservative* estimatation of the next step size. For example, we can consider the possibility of enlarging the step size so that the *marching* can be faster, or we can foresee the coming of a difficult region and, hence, using smaller step size.

- The actual implementation of a variable step method is quite complicated. The MATLAB codes are variable step methods.