5.5 Error Control and the Runge-Kutta-Fehlberg Method.

Consider the IVP:

$$\begin{cases} y'(t) = f(t,y), & a \le t \le b \quad (1.1) \\ y(a) = \alpha & (1.2) \end{cases}$$

and the one-step method approximating its solution $y(t)$ given by

arbitrary function suff. smooth.

$$w_{i+1} = w_i + h\,\phi(t_i; w_i), \quad w_0 = \alpha. \quad (1.3)$$
$$i = 0, 1, .., N-1.$$

The goal is to use a numerical method, such that for any $\varepsilon > 0$ and $\varepsilon < \varepsilon_0$; for ε_0 fixed.

We can find a moderately large step size h Such that

$$\boxed{|y_i - w_i| < \varepsilon, \quad \text{for all } i = 0, 1, .., N.} \quad (1.4)$$

In other words, we want to use the minimum number of grid points (less computational effort) that Satisfies (1.4), or global error less than ε for all $i = 0, 1, .., N$.

In general equally spaced grid points do not satisfy this condition. So, we need to adjust the "h" somehow.

We will see later that if we can control the local truncation error (L.T.E.), then the global error can also be controlled.

Therefore, we will focous now on controlling the L.T.E.

How? In order to do that we will use two numerical methods, one of $O(h^n)$ and the other of $O(h^{n+1})$

$$O(h^n): \quad w_{i+1} = w_i + h\,\phi(t_i, w_i), \quad w_0 = \alpha. \quad (2.1)$$

$$O(h^{n+1}): \quad \tilde{w}_{i+1} = \tilde{w}_i + h\,\tilde{\phi}(t_i, \tilde{w}_i), \quad w_0 = \alpha. \quad (2.2)$$

Ask students: How do we define the L.T.E. for (2.1) and (2.2)?

From (2.1) and (2.2),

$$y(t_{i+1}) = y(t_i) + h\phi(t_i, y(t_i), h) + O(h^{n+1}) \quad (3.1)$$

with $\boxed{\tau_{i+1}(h) = O(h^n)}$

and
$$y(t_{i+1}) = y(t_i) + h\tilde{\phi}(t_i, y(t_i), h) + O(h^{n+2}) \quad (3.2)$$

with $\boxed{\tilde{\tau}_{i+1}(h) = O(h^{n+1})}$

Now,

$$L.T.E = \tau_{i+1}(h) = \frac{y_{i+1} - y_i}{h} - \phi(t_i, y_i, h)$$

$$= \frac{y_{i+1} - y_i - h\phi(t_i, y_i, h)}{h}$$

assuming $\boxed{w_i \simeq y_i \text{ at ith step}}$

$$\boxed{\tau_{i+1}(h) = \underbrace{\frac{y_{i+1} - (w_i + h\phi(t_i, w_i, h))}{h}} = \frac{y_{i+1} - w_{i+1}}{h}} \quad (3.3)$$

Similarly, $$\boxed{\tilde{\tau}_{i+1}(h) = \frac{1}{h}(y_{i+1} - \tilde{w}_{i+1})} \quad (3.4)$$

We can rewrite (3.3)

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

$$= \frac{1}{h}\left(y_{i+1} - \tilde{w}_{i+1}\right) + \frac{1}{h}\left(\tilde{w}_{i+1} - w_{i+1}\right)$$

$$= \tilde{\tau}_{i+1} + \frac{1}{h}\left(\tilde{w}_{i+1} - w_{i+1}\right) = O(h^n)$$

Since $\tilde{\tau}_{i+1}(h) = \mathcal{O}(h^{n+1})$

then, $\tau_{i+1}(h) \simeq \frac{1}{h}(\tilde{w}_{i+1} - w_{i+1}) = \mathcal{O}(h^n)$. (4.1)

or $\tau_{i+1}(h) = K h^n$ (for some K). (4.2)

The (4.1) equation implies that

$$\tau_{i+1}(h) < \varepsilon \iff \frac{1}{h}(\tilde{w}_{i+1} - w_{i+1}) < \varepsilon \quad (4.3)$$

If (4.3) is not satisfied, then we introduce a factor "q" multiplying h such that

$$\tau_{i+1}(qh) \simeq K(qh)^n = q^n K h^n \simeq q^n \tau_{i+1}(h)$$

$$\simeq \frac{q^n}{h}(\tilde{w}_{i+1} - w_{i+1}),$$

We choose q, s.t.

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

Show Textbook results for a particular Example.

So, the procedure is

At i^{th} step Check if

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

If yes, then keep h for $(i+1)^{th}$ step.

Otherwise,

Change h by qh and use it in step. $(i+1)^{th}$

The downside of this technique may be the Computational Cost. Because, we are using two different numerical scheme at every time step.

However, the way R-K-F is constructed, it only requires 6 evaluations employing two R-K One of order 5 and the other of order 4.
This Combination normally requires 10 evaluations.

Runge-Kutta-Fehlberg Method

One popular technique that uses Inequality (5.22) for error control is the **Runge-Fehlberg method**. (See [Fe].) This technique uses a Runge-Kutta method with loc cation error of order five,

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

to estimate the local error in a Runge-Kutta method of order four given by

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

where the coefficient equations are

$$k_1 = hf(t_i, w_i),$$

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

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

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

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

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

An advantage to this method is that only six evaluations of f are required per step. A Runge-Kutta methods of orders four and five used together (see Table 5.9 on pa require at least four evaluations of f for the fourth-order method and an addition the fifth-order method, for a total of at least 10 function evaluations. So the Rung Fehlberg method has at least a 40% decrease in the number of function evaluations use of a pair of arbitrary fourth- and fifth-order methods.

In the error-control theory, an initial value of h at the ith step is used to find values of w_{i+1} and $\tilde{w}_{i+1}$, which leads to the determination of q for that step, and calculations are repeated. This procedure requires twice the number of function ev per step as without the error control. In practice, the value of q to be used is chosen s

Since $R \le 10^{-5}$, we can accept the approximation 0.9204886 for $y(0.25)$, but we should adjust the step size for the next iteration to $h = 0.9461033291(0.25) \approx 0.2365258$. However, only the leading 5 digits of this result would be expected to be accurate because R has only about 5 digits of accuracy. Because we are effectively subtracting the nearly equal numbers w_i and $\tilde{w}_i$ when we compute R, there is a good likelihood of round-off error. This is an additional reason for being conservative when computing q.

The results from the algorithm are shown in Table 5.11. Increased accuracy has been used to ensure that the calculations are accurate to all listed places. The last two columns in Table 5.11 show the results of the fifth-order method. For small values of t, the error is less than the error in the fourth-order method, but the error exceeds that of the fourth-order method when t increases. ■

Table 5.11

		RKF-4				RKF-5	
t_i	$y_i = y(t_i)$	w_i	h_i	R_i	$\lvert y_i - w_i \rvert$	$\hat{w}_i$	$\lvert y_i - \hat{w}_i \rvert$
0	0.5	0.5			0.5		
0.2500000	0.9204873	0.9204886	0.2500000	6.2×10^{-6}	1.3×10^{-6}	0.9204870	2.424×10^{-7}
0.4865522	1.3964884	1.3964910	0.2365522	4.5×10^{-6}	2.6×10^{-6}	1.3964900	1.510×10^{-6}
0.7293332	1.9537446	1.9537488	0.2427810	4.3×10^{-6}	4.2×10^{-6}	1.9537477	3.136×10^{-6}
0.9793332	2.5864198	2.5864260	0.2500000	3.8×10^{-6}	6.2×10^{-6}	2.5864251	5.242×10^{-6}
1.2293332	3.2604520	3.2604605	0.2500000	2.4×10^{-6}	8.5×10^{-6}	3.2604599	7.895×10^{-6}
1.4793332	3.9520844	3.9520955	0.2500000	7×10^{-7}	1.11×10^{-5}	3.9520954	1.096×10^{-5}
1.7293332	4.6308127	4.6308268	0.2500000	1.5×10^{-6}	1.41×10^{-5}	4.6308272	1.446×10^{-5}
1.9793332	5.2574687	5.2574861	0.2500000	4.3×10^{-6}	1.73×10^{-5}	5.2574871	1.839×10^{-5}
2.0000000	5.3054720	5.3054896	0.0206668		1.77×10^{-5}	5.3054896	1.768×10^{-5}

EXERCISE SET 5.5

1. Use the Runge-Kutta-Fehlberg method with tolerance $TOL = 10^{-4}$, $hmax = 0.25$, and $hmin = 0.05$ to approximate the solutions to the following initial-value problems. Compare the results to the actual values.
   a. $y' = te^{3t} - 2y, \quad 0 \le t \le 1, \quad y(0) = 0$; actual solution $y(t) = \frac{1}{5}te^{3t} - \frac{1}{25}e^{3t} + \frac{1}{25}e^{-2t}$.
   b. $y' = 1 + (t - y)^2, \quad 2 \le t \le 3, \quad y(2) = 1$; actual solution $y(t) = t + 1/(1 - t)$.
   c. $y' = 1 + y/t, \quad 1 \le t \le 2, \quad y(1) = 2$; actual solution $y(t) = t \ln t + 2t$.
   d. $y' = \cos 2t + \sin 3t, \quad 0 \le t \le 1, \quad y(0) = 1$; actual solution $y(t) = \frac{1}{2}\sin 2t - \frac{1}{3}\cos 3t + \frac{4}{3}$.
2. Use the Runge-Kutta Fehlberg Algorithm with tolerance $TOL = 10^{-4}$ to approximate the solution to the following initial-value problems.
   a. $y' = (y/t)^2 + y/t, \quad 1 \le t \le 1.2, \quad y(1) = 1$, with $hmax = 0.05$ and $hmin = 0.02$.
   b. $y' = \sin t + e^{-t}, \quad 0 \le t \le 1, \quad y(0) = 0$, with $hmax = 0.25$ and $hmin = 0.02$.
   c. $y' = (y^2 + y)/t, \quad 1 \le t \le 3, \quad y(1) = -2$, with $hmax = 0.5$ and $hmin = 0.02$.
   d. $y' = t^2, \quad 0 \le t \le 2, \quad y(0) = 0$, with $hmax = 0.5$ and $hmin = 0.02$.
3. Use the Runge-Kutta-Fehlberg method with tolerance $TOL = 10^{-6}$, $hmax = 0.5$, and $hmin = 0.05$ to approximate the solutions to the following initial-value problems. Compare the results to the actual values.
   a. $y' = y/t - (y/t)^2, \quad 1 \le t \le 4, \quad y(1) = 1$; actual solution $y(t) = t/(1 + \ln t)$.
   b. $y' = 1 + y/t + (y/t)^2, \quad 1 \le t \le 3, \quad y(1) = 0$; actual solution $y(t) = t\tan(\ln t)$.
   c. $y' = -(y + 1)(y + 3), \quad 0 \le t \le 3, \quad y(0) = -2$; actual solution $y(t) = -3 + 2(1 + e^{-2t})^{-1}$.
   d. $y' = (t + 2t^3)y^3 - ty, \quad 0 \le t \le 2, \quad y(0) = \frac{1}{3}$; actual solution $y(t) = (3 + 2t^2 + 6e^{t^2})^{-1/2}$.