# Chapter 3

# Step Control

## 3.1 Preliminaries

In this chapter, we shall study in more details the error analysis involved in one-step methods. Our ultimate goal is to use this analysis to design control mechanism for changing step sizes.

**Definition 3.1.1** *A one-step method is a numerical shceme that can be written in the form*

$$y_{n+1} = y_n + h\psi_f(x_n, y_n, h) \tag{3.1}$$

*where $\psi_f(x_n, y_n, h)$ is determined by $f, x_n, y_n$ and $h$.*

**Definition 3.1.2** *The method (3.1) is convergent if for every fixed $x$, $y_n \longrightarrow y(x)$ whenever $y_0 \longrightarrow y(0)$, $h = \frac{x}{n}$ and $n \longrightarrow \infty$.*

**Definition 3.1.3** *The local truncation error at $x_{n+1}$ of the method (3.1) is defined to be $T_{n+1}$ where*

$$T_{n+1} := y(x_{n+1}) - y(x_n) - h\psi_f(x_n, y(x_n), h). \tag{3.2}$$

**Definition 3.1.4** *The method (3.1) is said to be of order $p$ if $p$ is the largest integer for which $T_{n+1} = O(h^{p+1})$ for every $n$.*

**Remark.** In general, the local truncation error for method (3.1) is of the form

$$T_{n+1} = \phi(x_n, y(x_n))h^{p+1} + O(h^{p+2}) \tag{3.3}$$

where $\phi(x_n, y(x_n))$ is called the principal error function.

**Example.** The local truncation error for a 2-stage Runge-Kutta method is

$$T_{n+1} = \frac{h^3}{6}\left[\left(\frac{3}{4\gamma}\right)\left(\frac{\partial^2 f}{\partial y^2}f^2 + 2\frac{\partial^2 f}{\partial x \partial y}f + \frac{\partial^2 f}{\partial x^2}\right) - \left(\frac{\partial f}{\partial y}\right)^2 f - \frac{\partial f}{\partial y}\frac{\partial f}{\partial x}\right]$$

**Definition 3.1.5** *The method (3.1) is said to be consistent if*

$$\psi_f(x, y, 0) = f(x, y).$$

**Remark.** If (3.1) is consistent, then

$$\begin{aligned} T &= y(x+h) - y(x) - h\psi_f(x, y(x), h) \\ &= hy'(x) - h\psi_f(x, y(x), 0) + O(h^2) = O(h^2). \end{aligned}$$

Thus a consistent method has order at least one.

## 3.2 Error Estimate for Runge-Kutta Methods

Suppose the underlined Runge-Kutta method is of order $p$ and that no previous errors have been introduced, then

$$y(x_{n+1}) - y_{n+1} = \phi(x_n, y(x_n))h^{p+1} + O(h^{p+2}). \tag{3.4}$$

Suppose we compute another approximation $y^*_{n+1}$ to $y(x_{n+1})$ by using the same method but with step size $2h$. We have

$$y(x_{n+1}) - y^*_{n+1} = \phi(x_{n-1}, y(x_{n-1}))(2h)^{p+1} + O(h^{p+2}) \tag{3.5}$$

$$nonumber \tag{3.6}$$

$$= \phi(x_n, y(x_n))(2h)^{p+1} + O(h^{p+2}). \tag{3.7}$$

It follows that

$$y_{n+1} - y^*_{n+1} = (2^{p+1} - 1)\phi(x_n, y(x_n))h^{p+1} + O(h^{p+2}).$$

The principal local truncation error, that is taken as an estimate for the local truncation error, may be expressed as

$$\phi(x_n, y(x_n))h^{p+1} = \frac{y_{n+1} - y^*_{n+1}}{2^{p+1} - 1}. \tag{3.8}$$

**Remark.** The above estimate is usually quite adequate for step control purpose, but it involves a considerable increase in the computational effort. Many other error estimators are available in the literature. For example,

1. The quantity

   $$E_{n+1} := \frac{1}{30}\left(10y_{n-2} + 9y_{n-1} - 18y_n - y_{n+1} + 3h\left[f_{n-2} + 6f_{n-1} + 3f_n\right]\right)$$

   can be used as an error estimator for the fourth order Runge-Kutta method. Note that the formula involves evaluations of $f$ that has already been made in previous steps. (Ref: R. E. Scraton, Estimation of the truncation error in RK and allied processes, Comput. J., 7(1964), pp246-248.)

2. The 5-stage Runge-Kutta method

$$\begin{array}{c|ccccc}
0 & 0 & & & & \\
\frac{2}{9} & \frac{2}{9} & 0 & & & \\
\frac{1}{3} & \frac{1}{12} & \frac{1}{4} & 0 & & \\
\frac{3}{4} & \frac{69}{128} & -\frac{243}{128} & \frac{270}{128} & 0 & \\
\frac{9}{10} & -\frac{3105}{10000} & \frac{18255}{10000} & -\frac{11016}{10000} & \frac{4896}{10000} & 0 \\
\hline
 & \frac{17}{162} & \frac{81}{170} & \frac{32}{135} & 0 & \frac{250}{1377}
\end{array}$$

is a fourth order method. It can be shown that the quantity defined by

$$T_{n+1} = \frac{hqr}{s}$$

where

$$\begin{array}{rcl}
q & = & -\frac{1}{18}k_1 + \frac{27}{170}k_3 - \frac{4}{15}k_4 + \frac{25}{153}k_5 \\
r & = & \frac{19}{24}k_1 - \frac{27}{8}k_2 + \frac{57}{20}k_3 - \frac{4}{15}k_4 \\
s & = & k_4 - k_1
\end{array}$$

is a reasonable estimate for the local truncation error.

It should be noted that both examples above do not require additional function evaluation in order to compute the error estimate. It appears that all error estimates of the local truncation errors for Runge-Kutta methods either average the error over a number of steps (memory) or require additional function evaluations. This is one of the drawback of nonlinear methods. (Ref: Shampine and Watts, The art of writing a Runge-Kutta code, Mathematical Software III, ed. John Rice, pp. 257-275.)

Another way to obtain error estimates is to derive Runge-Kutta methods in the form:

$$\begin{array}{c|c}
a & B \\
\hline
 & c^T \\
 & \hat{c}^T \\
\hline
 & E^T
\end{array}$$

That is, we seek two methods such that the one defined by $a, c, B$ has order $p$ and that defined by $a, \hat{c}, B$ has order $p+1$. The difference between the values for

$y_{n+1}$ generated by these two methods is then an estimate of the local truncation error. Define $E := \hat{c} - c$. Then the error estimate is given by

$$T_{n+1} := h \sum_{r=1}^{R} E_i k_i.$$

Such a method usually is labeled as $(p, p+1)$. Perhaps the most popular $(4,5)$ method is the so called RKF45 developed by Fehlberg:

$$\begin{array}{c|cccccc}
0 & 0 & & & & & \\
\frac{1}{4} & \frac{1}{4} & 0 & & & & \\
\frac{3}{8} & \frac{3}{32} & \frac{9}{32} & 0 & & & \\
\frac{12}{13} & \frac{1932}{2197} & -\frac{7200}{2197} & \frac{7296}{2197} & 0 & & \\
1 & \frac{439}{216} & -8 & \frac{3680}{513} & -\frac{845}{4104} & 0 & \\
\frac{1}{2} & -\frac{8}{27} & 2 & -\frac{3544}{2565} & \frac{1859}{4104} & -\frac{11}{40} & 0 \\
\hline
 & \frac{25}{216} & 0 & \frac{1408}{2565} & \frac{2197}{4104} & -\frac{1}{5} & 0 \\
 & \frac{16}{135} & 0 & \frac{6656}{12825} & \frac{28561}{56430} & -\frac{9}{50} & \frac{2}{55} \\
\hline
 & \frac{1}{360} & 0 & -\frac{128}{4275} & -\frac{2197}{75240} & \frac{1}{50} & \frac{2}{55}
\end{array}$$

Note that if the error estimate is not required, then this is a 5-stage (fourth order) method.

## 3.3 Global Error Analysis

In this section we prove that the global error is usually one order less than the local truncation error.

**Theorem 3.3.1** *Assume $\psi_f(x, y, h)$ in (3.1) is continuous in $x, y$ and $h$ for $0 \le x \le b, 0 \le h \le h_0$ and all $y$. Furthermore, suppose $\psi_f$ satisfies a Lipschitz condition in $x, y$ and $h$. Then the method (3.1) is convergent if and only if it is consistent.*

*Proof.* We first show consistency implies convergence. Denote $\psi_f(x, y, 0) = g(x, y)$. Consider the IVP

$$\begin{aligned} z' &= g(x, z) \\ z(0) &= y_0. \end{aligned} \tag{3.9}$$

We want to show that the numerical solution of (3.1) converges to $z(x)$. Observe that

$$\begin{aligned} y_{n+1} &= y_n + h\psi_f(x_n, y_n, h) \\ z^{(i)}_{n+1} &= z^{(i)}_n + hg^{(i)}(x_n + \theta^{(i)}h, z(x_n + \theta^{(i)}h)) \end{aligned}$$

where $^{(i)}$ means the $i$-the component of a vector. It follows that

$$e^{(i)}_{n+1} = e^{(i)}_n + h\Big[\psi^{(i)}_f(x_n, y_n, h) - \psi^{(i)}_f(x_n, z(x_n), h) + \psi^{(i)}_f(x_n, z(x_n), h) \\ -\psi^{(i)}_f(x_n, z(x_n), 0) + \psi^{(i)}_f(x_n, z(x_n), 0) - g^{(i)}(x_n + \theta^{(i)}h, z(x_n + \theta^{(i)}h))\Big]$$

We have the following estimates:

$$\begin{aligned} \|\psi^{(i)}_f(x_n, y_n, h) - \psi^{(i)}_f(x_n, z(x_n), h)\| &\leq L_y\|e_n\| \\ \|\psi^{(i)}_f(x_n, z(x_n), h) - \psi^{(i)}_f(x_n, z(x_n), 0)\| &\leq L_h h \end{aligned}$$

and

$$\begin{aligned} |\psi^{(i)}_f(x_n, z(x_n), 0) - & \quad g^{(i)}(x_n + \theta^{(i)}h, z(x_n + \theta^{(i)}h))| \\ &= |g^{(i)}(x_n, z(x_n)) - g^{(i)}(x_n + \theta^{(i)}h, z(x_n + \theta^{(i)}h))| \\ &\leq L_x\theta^{(i)}h + L_y\|z'(x_n + \xi\theta^{(i)}h)\|\theta^{(i)}h \leq Lh. \end{aligned}$$

Therefore,

$$\begin{aligned} \|e_{n+1}\| &\leq \|e_n\| + hL_y\|e_n\| + h^2(L_h + L) \\ &= (1 + hL_y)\|e_n\| + h^2(L_h + L). \end{aligned}$$

It follows (see the proof of Theorem 1.2.2) that

$$\|e_N\| \leq (L_h + L)h\frac{e^{L_y b - 1}}{L_y} + e^{L_y b}\|e_0\|$$

where $x = x_N$, $h = \frac{x}{N}$. Obviously, as $h$ goes to zero ($N$ goes to $\infty$), so does $e_N$.

Suppose now the method converges. Then $z(x) \equiv y(x)$. The uniqueness theorem implies that $g(x, y) = f(x, y)$. □

**Remark.** The hypotheses about $\psi_f$ in the above theorem are satisfied, in general, so long as the function $f$ satisfies a Lipschitz condition in $y$.

**Theorem 3.3.2** *Assume $\psi_f$ satisfies conditions in Theorem 3.3.1. Assume also that the local truncation error $T_{n+1}$ defined in (3.2) is bounded by*

$$\|T_{n+1}\| \leq Dh^{p+1}. \tag{3.10}$$

*Then*

$$\|y_n - y(x_n)\| \leq Dh^p\frac{e^{L_y b - 1}}{L_y} + e^{L_y b}\|y_0 - y(x_0\|. \tag{3.11}$$

*Proof.* By definition, we have

$$\begin{aligned} e_{n+1} &= y_n - y(x_n) + [h\psi_f(x_n, y_n, h) - (y(x_{n+1}) - y(x_n)] \\ &= e_n + h[\psi_f(x_n, y_n, h) - \psi_f(x_n, y(x_n), h)] + T_n. \qquad (3.12) \end{aligned}$$

It follows that

$$\|e_{n+1}\| \leq \|e_n\| + hL_y\|e_n\| + h^{p+1}D. \qquad (3.13)$$

The assertion follows. □

**Remark.** We see from above theorem that the global error is one less than the local truncation error.

**Theorem 3.3.3** *Suppose (3.3) holds, i.e., suppose*

$$T_{n+1} = \phi(x_n, y(x_n))h^{p+1} + O(h^{p+2}).$$

*Suppose also that the function $\psi_f$ has continuous second derivatives. Then*

$$e_n = \delta(x_n)h^p + O(h^{p+1}) \qquad (3.14)$$

*where $\delta(x)$ solves the IVP*

$$\begin{aligned} \delta' &= f_y(x, y(x))\delta + \phi(x, y(x)) \qquad (3.15) \\ \delta(0) &= \frac{e_0}{h^p}. \qquad (3.16) \end{aligned}$$

*Proof.* Define $\delta_n := \frac{e_n}{h^p}$. Upon substitution into (3.12), we obtain

$$\begin{aligned} \delta_{n+1} &= \delta_n + h^{1-p}\left[\psi_f(x_n, y(x_n) + h^p\delta_n, h) - \psi_f(x_n, y(x_n), h)\right] \\ &+ h\phi(x_n, y(x_n)) + O(h^2). \end{aligned}$$

Observe that

$$\begin{aligned} \psi_f(x_n, y(x_n) + h^p\delta_n, h) &= \psi_f(x_n, y(x_n), h) + \frac{\partial\psi_f(x_n, y(x_n), h)}{\partial y}h^p\delta_n + k_1h^{2p}, \\ \frac{\partial\psi_f(x_n, y(x_n), h)}{\partial y}h^p\delta_n &= \frac{\partial\psi_f(x_n, y(x_n), 0)}{\partial y}h^p\delta_n + k_2h^{p+1}. \end{aligned}$$

We may, therefore, write

$$\delta_{n+1} = \delta_n + h\left[\frac{\partial f(x_n, y(x_n))}{\partial y}\delta_n + \phi(x_n, y(x_n)) + k_1h^p + k_2h\right].$$

This difference scheme can be viewed as a Euler step applied to the differential equation (3.15). □

## 3.4 Step Control

We have learned from numerical experiences that in order to solve an IVP efficiently and accurately, one has to adjust the step size $h$ over the interval by taking into account the following three considerations:

1. The principal local truncation error remains at each step less than a prescribed tolerance. This is the error control.
2. The quantity $\bar{h}$ lies inside the region of absolute stability. This is the stability control.
3. When a nonlinear algebraic equation has to be solved (such as when an implicit method is applied), the iteration scheme (such as the Newton method) has to converge. This is the convergence control.

**Remark.** All the results discussed in this chapter remains true for variable step methods, provided

(a). We iterpret $h$ as the maximum step size.

(b). Assume the existence of a function $0 < \Delta \leq \theta(t) \leq 1$ such that the step size $h_n$ from $x_n$ to $x_{n+1}$ is given by

$$h_n := h\theta(x_n. \tag{3.17}$$

The objective of a variable-step variable-order method is to generate a numerical result that is acceptable as an approximation to the true solution with minimum effort. We now consider the different possibilities:

**Choice of Order.** Assume we have available a number of methods with orders $r = 1, 2, \ldots$. Suppose for each method we also have determined the best choice of step size $h_r$ so that the overhead is optimal while satisfying the error tolerance. Among the many candidates (of different orders) we want to select the optimal order that minimizes the total work. We can argue, similar to Theorem 3.3.3, that for a variable-step method of order $r$:

$$e_n = \delta_r(x_n)h_r^r + O(h_r^{r+1})$$

where $\delta_r(x)$ satisfies

$$\delta'(x) = \frac{\partial f(x,y)}{\partial y}\delta(x) + \theta^r(x)\phi(x,y). \tag{3.18}$$

Thus, an estimate of $h_r$ is given by

$$\|e_n\| \approx \|\delta_r(x_n)\|h_r^r \leq E. \tag{3.19}$$

Suppose that the amount of work per step is $k_r$ in the $r$-th order method. The total amount of work is therefore $Nk_r$ where $N$ is the total step taken. In general,

$$\frac{b-a}{h_r} \leq N \leq \frac{b-a}{\Delta h_r}$$

and hence $N = O(\frac{1}{h_r})$. The total work $W_r$ is approximately given by

$$W_r = \frac{L_r}{h_r}$$

where $L_r$ depends on the problem. Now consider

$$\|e_N\| \approx \|\delta_r\| \left(\frac{L_r}{W_r}\right)^r.$$

A qualitative graph of would be:

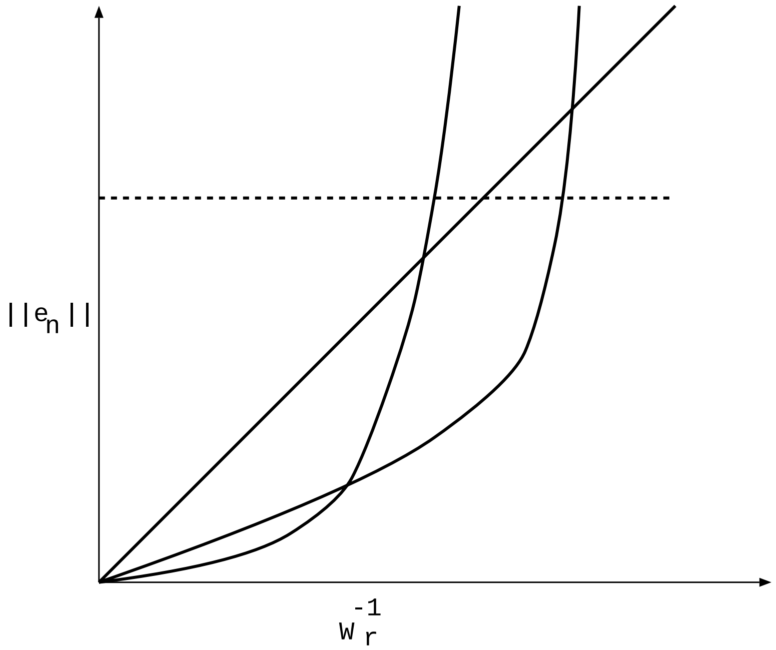

Figure 3.1: Qualitative graph of $\|e_N\|$ versus $W_r^{-1}$.

The points at which these curves cross depend on the values of $\|\delta_r\|$ and $L_r$. However, we can already draw quite a general conclusion as follows:

i. For sufficient small $e_n$, we should use higher order method (as this will result in larger $W_r^{-1}$.

ii. For low accuracy problems, we should use lower order methods.

iii. It is not always true that more accuracy can be obtained by going to a higher order method when a step size is fixed.

In general for one-step method the choice of order is not as essential as the choice of step size because all the function evaluations are based on information in the current interval $[x_n, x_n + h_n]$. No techniques are currently available on the selection of orders for Runge-Kutta methods.

**Choice of Step Size.** Although the optimal order and the step size can be asymptotically approximated, generally these criteria are too complicated to make practical applications. We usuall have to give up the optimum choices by working on a heuristic basis. This is particularly true for a one-step method. (Remember that a multi-step method can eventually be reduced to a one-step method.)

We now discuss how a step size is controlled. We shall assume that a system of differential equation is being solved. We shall take into account that each component of the solution may behave differently. Suppose at each step an error estimate $ERR_i$, $i = 1, \ldots, n$ has been obtained for the $i$-th component $Y_i$ of the solution. Let $RE$ and $AE$ denote, respectively, the relative error tolerance and the absolute error tolerance. Define $EPS := \max(RE, AE)$. To each component, we associate a prescribed weight function $W_i$. For each step we attempt to control the error so that

$$ERK := \left(\sum_{i=1}^{n}\left(\frac{ERR_i}{W_i}\right)^2\right)^{1/2} \leq EPS. \tag{3.20}$$

**Example.** By choosing the weight function appropriately, we can specify a variety of error criteria:

- $W_i := 1.0 \Longrightarrow$ Absolute error in the $i$-th component is $\leq EPS$.
- $W_i := |Y_i| \Longrightarrow$ Relative error in the $i$-th component is $\leq EPS$.
- $W_i := |Y_i' \Longrightarrow$ Relative error with respect to the first derivative.
- $W_i := |Y_i| * RE/EPS + AE/EPS \Longrightarrow |ERR_i| \leq |Y_i * RE + AE$.
- $W_i := |Y_i|$ or 1 at $t = 0$, and then $W_i := \max(W_i, |Y_i| \Longrightarrow$ Relative error for increasing components and absolute error for decreasing components.

A heuristic control mechanism is as follows:

1. Compute
$$\rho := \left(\frac{EPS}{2ERK}\right)^{1/p}. \tag{3.21}$$
2. If (3.20) is satisfied, the step is successful. Estimate the next step size $h_{n+1}$:
    (a) If $\rho \geq 2$, then $h_{n+1} := 2h_n$.
    (b) If $1 \leq \rho < 2$, then either $h_{n+1} := h_n$ or $h_{n+1} := \rho^{9/10} h_n$.
3. If (3.20) fails, the step is not acceptable. Reduce the step size by a factor of $\max\{0.5, \min\{0.9, \rho\}\}$.