Appendix A
Runge-Kutta Methods

The Runge-Kutta methods are an important family of iterative methods for the approximation of solutions of ODE's, that were develoved around 1900 by the german mathematicians C. Runge (1856–1927) and M.W. Kutta (1867–1944). We start with the considereation of the explicit methods. Let us consider an initail value problem (IVP)

$$\frac{d\,\mathbf{x}}{d\,t} = f(t, \mathbf{x}(t)), \tag{A.1}$$

$\mathbf{x}(t) = (x_1(t), x_2(t), \ldots x_n(t))^T$, $f \in [a,b] \times \mathbb{R}^n \to \mathbb{R}^n$, with an initial condition

$$\mathbf{x}(0) = \mathbf{x}_0\,. \tag{A.2}$$

We are interested in a numerical approximation of the continuously differentiable solution $\mathbf{x}(t)$ of the IVP (A.1)–(A.2) over the time interval $t \in [a,b]$. To this aim we subdivide the interval $[a,b]$ into M equal subintervals and select *the mesh points* t_j [11, 8]

$$t_j = a + j\,h\,, \qquad j = 0, 1, \ldots, M, \quad h = \frac{b-a}{M}\,. \tag{A.3}$$

The value h is called *a step size*.

The family of explicit Runge–Kutta (RK) methods of the m'th stage is given by [11, 9]

$$\mathbf{x}(t_{n+1}) := \mathbf{x}_{n+1} = \mathbf{x}_n + h\sum_{i=1}^{m} c_i k_i\,, \tag{A.4}$$

where

$$
\begin{aligned}
k_1 &= f(t_n, \mathbf{x}_n), \\
k_2 &= f(t_n + \alpha_2 h, \mathbf{x}_n + h\beta_{21} k_1(t_n, \mathbf{x}_n)), \\
k_3 &= f(t_n + \alpha_3 h, \mathbf{x}_n + h(\beta_{31} k_1(t_n, \mathbf{x}_n) + \beta_{32} k_2(t_n, \mathbf{x}_n))), \\
&\vdots \\
k_m &= f(t_n + \alpha_m h, \mathbf{x}_n + h \sum_{j=1}^{m-1} \beta_{mj} k_j).
\end{aligned}
$$

To specify a particular method, we need to provide the integer m (the number of stages), and the coefficients α_i (for $i = 2, 3, ..., m$), β_{ij} (for $1 \leq j < i \leq m$), and c_i (for $i = 1, 2, ..., m$). These data are usually arranged in a co-called *Butcher tableau* (after John C. Butcher) [11, 9]:

Table A.1 The Butcher tableau.

0					
α_2	β_{21}				
α_3	β_{31}	β_{32}			
$\vdots$	$\vdots$	$\vdots$	$\ddots$		
$\vdots$	$\vdots$	$\vdots$			
α_m	β_{m1}	β_{m2}	.....	β_{mm-1}	
	c_1	c_2	.....	c_{m-1}	c_m

Examples

1. Let $m = 1$. Then

$$
\begin{aligned}
k_1 &= f(t_n, \mathbf{x}_n), \\
\mathbf{x}_{n+1} &= \mathbf{x}_n + h c_1 f(t_n, \mathbf{x}_n).
\end{aligned}
$$

   On the other hand, the Taylor expansion yields

$$
\mathbf{x}_{n+1} = \mathbf{x}_n + h\dot{\mathbf{x}}\big|_{t_n} + \cdots = \mathbf{x}_n + h f(t_n, \mathbf{x}_n) + \mathscr{O}(h^2) \Rightarrow c_1 = 1.
$$

   Thus, the first-stage RK-method is equivalent to the explicit Euler's method. Note that the Euler's method is of the first order of accuracy. Thus we can speak about the RK method of the first order.
2. Now consider the case $m = 2$. In this case Eq. (A.4) is equivalent to the system

$$
\begin{aligned}
k_1 &= f(t_n, \mathbf{x}_n)\,, \\
k_2 &= f(t_n + \alpha_2 h, \mathbf{x}_n + h\beta_{21} k_1)\,, \\
\mathbf{x}_{n+1} &= \mathbf{x}_n + h(c_1 k_1 + c_2 k_2)\,.
\end{aligned}
\tag{A.5}
$$

Now let us write down the Taylor series expansion of $\mathbf{x}$ in the neighborhood of t_n up to the h^2 term, i.e.,

$$
\mathbf{x}_{n+1} = \mathbf{x}_n + h\left.\frac{d\mathbf{x}}{dt}\right|_{t_n} + \frac{h^2}{2}\left.\frac{d^2\mathbf{x}}{dt^2}\right|_{t_n} + \mathscr{O}(h^3)\,.
$$

However, we know that $\dot{\mathbf{x}} = f(t, \mathbf{x})$, so that

$$
\frac{d^2\mathbf{x}}{dt^2} := \frac{df(t,\mathbf{x})}{dt} = \frac{\partial f(t,\mathbf{x})}{\partial t} + f(t,\mathbf{x})\frac{\partial f(t,\mathbf{x})}{\partial \mathbf{x}}\,.
$$

Hence the Taylor series expansion can be rewritten as

$$
\mathbf{x}_{n+1} - \mathbf{x}_n = h\, f(t_n, \mathbf{x}_n) + \frac{h^2}{2}\left(\frac{\partial f}{\partial t} + f\frac{\partial f}{\partial \mathbf{x}}\right)\Bigg|_{(t_n,\mathbf{x}_n)} + \mathscr{O}(h^3)\,. \tag{A.6}
$$

On the other hand, the term k_2 in the proposed RK method can also expanded to $\mathscr{O}(h^3)$ as

$$
k_2 = f(t_n + \alpha_2 h, \mathbf{x}_n + h\beta_{21}k_1) = h\, f(t_n,\mathbf{x}_n) + h\,\alpha_2 \frac{\partial f}{\partial t}\Bigg|_{(t_n,\mathbf{x}_n)} + h\beta_{21}\, f\,\frac{\partial f}{\partial \mathbf{x}}\Bigg|_{(t_n,\mathbf{x}_n)} + \mathscr{O}(h^3)\,.
$$

Now, substituting this relation for k_2 into the last equation of (A.5), we achieve the following expression:

$$
\mathbf{x}_{n+1} - \mathbf{x}_n = h\,(c_1 + c_2)\, f(t_n, \mathbf{x}_n) + h^2 c_2\,\alpha_2 \frac{\partial f}{\partial t}\Bigg|_{(t_n,\mathbf{x}_n)} + h^2 c_2 \beta_{21} f \frac{\partial f}{\partial \mathbf{x}}\Bigg|_{(t_n,\mathbf{x}_n)} + \mathscr{O}(h^3)\,.
$$

Making comparision the last equation and Eq. (A.6) we can write down the system of algebraic equations for unknown coefficients

$$
\begin{aligned}
c_1 + c_2 &= 1\,, \\
c_2\,\alpha_2 &= \frac{1}{2}\,, \\
c_2\,\beta_{21} &= \frac{1}{2}\,.
\end{aligned}
$$

The system involves four unknowns in three equations. That is, one additional condition must be supplied to solve the system. We discuss two useful choices, namely

a) Let $\alpha_2 = 1$. Then $c_2 = 1/2$, $c_1 = 1/2$, $\beta_{21} = 1$. The corresponding Butcher tableau reads:

$$\begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & 1/2 & 1/2 \end{array}$$

Thus, in this case the two-stages RK method takes the form

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{2}\Big(f(t_n, \mathbf{x}_n) + f(t_n + h, \mathbf{x}_n + h f(t_n, \mathbf{x}_n)) \Big),$$

and is equivalent to the Heun's method, so we refer the last method to as RK-method of the second order.

b) Now let $\alpha_2 = 1/2$. In this case $c_2 = 1$, $c_1 = 0$, $\beta_{21} = 1/2$. The corresponding Butcher tableau reads:

$$\begin{array}{c|cc} 0 & & \\ 1/2 & 1/2 & \\ \hline & 0 & 1 \end{array}$$

In this case the second-order RK method (A.4) can be written as

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\, f\big(t_n + \frac{h}{2}, \mathbf{x}_n + \frac{h}{2} f(t_n, \mathbf{x}_n)\big)$$

and is called the *RK2 method*.

RK4 Methods

One member of the family of Runge–Kutta methods (A.4) is often referred to as *RK4 method* or *classical RK method* and represents one of the solutions corresponding to the case $m = 4$. In this case, by matching coefficients with those of the Taylor series one obtains the following system of equations [8]

$$
\begin{aligned}
c_1 + c_2 + c_3 + c_4 &= 1\,, \\
\beta_{21} &= \alpha_2\,, \\
\beta_{31} + \beta_{32} &= \alpha_3\,, \\
c_2\alpha_2 + c_3\alpha_3 + c_4\alpha_4 &= \frac{1}{2}\,, \\
c_2\alpha_2^2 + c_3\alpha_3^2 + c_4\alpha_4^2 &= \frac{1}{3}\,, \\
c_2\alpha_2^3 + c_3\alpha_3^3 + c_4\alpha_4^3 &= \frac{1}{4}\,, \\
c_3\alpha_2\beta_{32} + c_4(\alpha_2\beta_{42} + \alpha_3\beta_{43}) &= \frac{1}{6}\,, \\
c_3\alpha_2\alpha_3\beta_{32} + c_4\alpha_4(\alpha_2\beta_{42} + \alpha_3\beta_{43}) &= \frac{1}{8}\,, \\
c_3\alpha_2^2\beta_{32} + c_4(\alpha_2^2\beta_{42} + \alpha_3^2\beta_{43}) &= \frac{1}{12}\,, \\
c_4\alpha_2\beta_{32}\beta_{43} &= \frac{1}{24}\,.
\end{aligned}
$$

The system involves thirteen unknowns in eleven equations. That is, two additional condition must be supplied to solve the system. The most useful choices is [9]

$$
\alpha_2 = \frac{1}{2}\,, \qquad \beta_{31} = 0\,.
$$

The corresponding Butcher tableau is presented in Table A.2. The tableau A.2 yields

Table A.2 The Butcher tableau corresponding to the RK4 method.

0				
1/2	1/2			
1/2	0	1/2		
1	0	0	1	
	1/6	1/3	1/3	1/6

the equivalent corresponding equations defining the classical RK4 method:

$$
\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{6}\big(k_1 + 2k_2 + 2k_3 + k_4\big), \tag{A.7}
$$

where

$$
\begin{aligned}
k_1 &= f(t_n, \mathbf{x}_n), \\
k_2 &= f(t_n + \frac{h}{2}, \mathbf{x}_n + \frac{h}{2}k_1), \\
k_3 &= f(t_n + \frac{h}{2}, \mathbf{x}_n + \frac{h}{2}k_2), \\
k_4 &= f(t_n + h, \mathbf{x}_n + hk_3).
\end{aligned}
$$

This method is reasonably simple and robust and is a good general candidate for numerical solution of ODE's when combined with an intelligent adaptive step-size routine or an embedded methods (,e.g., so-called Runge-Kutta-Fehlberg methods (RKF45)).

Remark:

Notice that except for the classical method (A.7), one can also construct other RK4 methods. We mention only so-called $3/8$-*Runge-Kutta method.* The Brutcher tableau, corresponding to this method is presented in Table A.3.

Table A.3 The Butcher tableau corresponding to the $3/8$- Runge-Kutta method.

0				
1/3	1/3			
2/3	-1/3	1		
1	1	-1	1	
	1/8	3/8	3/8	1/8

Geometrical interpretation of the RK4 method

Let us consider a curve $\mathbf{x}(t)$, obtained by (A.7) over a single time step from t_n to t_{n+1}. The next value of approximation $\mathbf{x}_{n+1}$ is obtained ty integrating the slope function, i.e.,

$$
\mathbf{x}_{n+1} - \mathbf{x}_n = \int_{t_n}^{t_{n+1}} f(t, \mathbf{x})\, dt\,. \tag{A.8}
$$

Now, if the Simpson's rule is applied, the approximation to the integral of the last equation reads [10]

$$
\int_{t_n}^{t_{n+1}} f(t, \mathbf{x})\, dt \approx \frac{h}{6}\left(f(t_n, \mathbf{x}(t_n)) + 4\, f(t_n + \frac{h}{2}, \mathbf{x}(t_n + \frac{h}{2})) + f(t_{n+1}, \mathbf{x}(t_{n+1}))\right). \tag{A.9}
$$

On the other hand, the values k_1, k_2, k_3 and k_4 are approximations for slopes of the curve $\mathbf{x}$, i.e., k_1 is the slope of the left end of the interval, k_2 and k_3 describe two estimations of the slope in the middle of the time interval, whereas k_4 corresponds to the slope at the right. Hence, we can choose $f(t_n, \mathbf{x}(t_n)) = k_1$ and $f(t_{n+1}, \mathbf{x}(t_{n+1})) = k_4$, whereas for the value in the middle we choose the average of k_2 and k_3, i.e.,

$$f(t_n + \frac{h}{2}, \mathbf{x}(t_n + \frac{h}{2})) = \frac{k_2 + k_3}{2}\,.$$

Then Eq. (A.8) becomes

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{6}\left(k_1 + \frac{4(k_2 + k_3)}{2} + k_4\right),$$

which is equivalent to the RK4 schema (A.7).

Stage versus Order

The local truncation error ε for the method (A.7) can be estimated from the error term for the Simpson's rule (A.9) and equals [10, 8]

$$\varepsilon_{n+1} = -h^5 \frac{\mathbf{x}^{(4)}}{2880}\,.$$

Now we can estimate the final global error E, if we suppose that only the error above is presented. After M steps the accumulated error for the RK4 method reads

$$E(\mathbf{x}(b), h) = -\sum_{k=1}^{M} h^5 \frac{\mathbf{x}^{(4)}}{2880} \approx \frac{b-a}{2880}\mathbf{x}^{(4)} h = \mathscr{O}(h^4)\,.$$

That is, the RK4 method (A.7) is of the fourth order. Now, let us compare two appximations, obtained using the time steps h and $h/2$. For the step size h we have

$$E(\mathbf{x}(b), h) \approx K h^4\,,$$

with $K = \text{const}$. Hence, for the step $h/2$ we get

$$E(\mathbf{x}(b), \frac{h}{2}) = K\frac{h^4}{16} \approx \frac{1}{16}E(\mathbf{x}(b), h)\,.$$

That is, if the step size in (A.7) is reduced by the factor of two, the global error of the method will be reduced by the factor of $1/16$.

Remark:

In general there are two ways to improve the accuracy:

1. One can reduce the time step h, i.e., the amount of steps increases;
2. The method of the higher convergency order can be used.

However, increasing of the convergency order p is reasonable only up to some limit, given by so-called *Butcher barrier* [11], which says, that the amount of stages m grows faster, as the order p. In other words, *for $m \geq 5$ there are no explicit RK methods with the convergency order $p = m$ (the corresponding system is unsolvable)*. Hence, in order to reach convergency order five one needs six stages. Notice that further increasing of the stage $m = 7$ leads to the convergency order $p = 5$ as well.

A.0.1 Adaptive stepsize control and embedded methods

As mentioned above, one way to guarantee accuracy in the solution of (A.1)–(A.1) is to solve the problem twice using step sizes h and $h/2$. To illustrate this approach, let us consider the RK method of the order p and denote an exact solution at the point $t_{n+1} = t_n + h$ by $\widetilde{\mathbf{x}}_{n+1}$, whereas $\mathbf{x}_1$ and $\mathbf{x}_2$ represent the approximate solutions, corresponding to the step sizes h and $h/2$. Now let us perform one step with the step size h and after that two steps each of size $h/2$. In this case the true solution and two numerical approximations are related by

$$\begin{aligned}
\widetilde{\mathbf{x}}_{n+1} &= \mathbf{x}_1 + C h^{p+1} + \mathscr{O}(h^{p+2}), \\
\widetilde{\mathbf{x}}_{n+1} &= \mathbf{x}_2 + 2C \left(\frac{h}{2}\right)^{p+1} + \mathscr{O}(h^{p+2}).
\end{aligned}$$

That is,

$$|\mathbf{x}_1 - \mathbf{x}_2| = C h^{p+1} \left(1 - \frac{1}{2^p}\right) \Leftrightarrow C = \frac{|\mathbf{x}_1 - \mathbf{x}_2|}{(1 - 2^{-p}) h^{p+1}}.$$

Substituing the relation for C in the second estimate for the true solution we get

$$\widetilde{\mathbf{x}}_{n+1} = \mathbf{x}_2 + \varepsilon + \mathscr{O}(h^{p+2}),$$

where

$$\varepsilon = \frac{|\mathbf{x}_1 - \mathbf{x}_2|}{2^p - 1}$$

can be considered as a convenient *indicator* of the truncation error. That is, we have improved our estimate to the order $p + 1$. For example, for $p = 4$ we get

$$\widetilde{\mathbf{x}}_{n+1} = \mathbf{x}_2 + \frac{|\mathbf{x}_1 - \mathbf{x}_2|}{15} + \mathscr{O}(h^6).$$

This estimate is accurate to fifth order, one order higter than with the original step h. However, this method is not efficient. First of all, it requires a significant amount

of computation (we should solve the equation three times at each time step). The second point is, that we have no possibility to control the truncation error of the method (higher order means not always higher accuracy).
However we can use an estimate ε for the *step size control*, namely we can compare ε with some *desired accuracy* ε_0 (see Fig A.1).

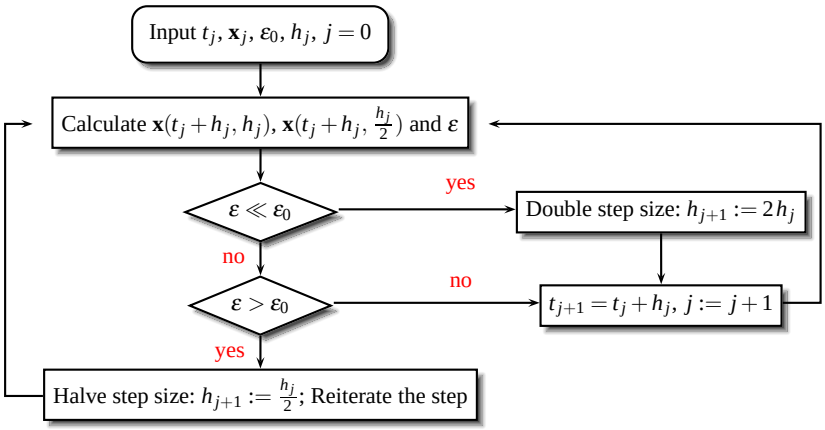

Fig. A.1 Flow diagramm of the step size control by use of the step doubling method.

Alternatively, using the estimate ε, we can try to formulate the following problem of the *adaptive step size control*, namely: Using the given values $\mathbf{x}_j$ and t_j, find the largest possible step size h_{new}, so that the truncation error after the step with this step size remains below some given desired accuracy ε_0, i.e,

$$Ch_{new}^{p+1} \le \varepsilon_0 \Leftrightarrow \left(\frac{h_{new}}{h}\right)^{p+1} \frac{|\mathbf{x}_1 - \mathbf{x}_2|}{1-2^{-p}} \le \varepsilon_0 .$$

That is,

$$h_{new} = h\left(\frac{\varepsilon_0}{\varepsilon}\right)^{1/p+1} .$$

Then if the two answers are in close agreement, the approximation is accepted. If $\varepsilon > \varepsilon_0$ the step size has to be decreased, whereas the relation $\varepsilon < \varepsilon_0$ means, that the step size has to be increased in the next step.

Notice that because our estimate of error is not exact, we should put some "safety" factor $\beta \simeq 1$ [11, 9]. Usually, $\beta = 0.8, 0.9$. The flow diagramm, corresponding to the the adaptive step size control is shown on Fig. A.2

Notice one additional technical point. The choise of the desired error ε_0 depends on the IVP we are interested in. In some applications it is convinient to set ε_0 propotional to h [9]. In this case the exponent $1/p+1$ in the estimate of the new time step is no longer correct (if h is reduced from a too-large value, the new predicted value h_{new} will fail to meet the desired accuracy, so instead of $1/p+1$ we should scale with $1/p$ (see [9] for details)). That is, the optimal new step size can be written as

$$h_{new} = \begin{cases} \beta h\left(\frac{\varepsilon_0}{\varepsilon}\right)^{1/p+1}, & \varepsilon \ge \varepsilon_0, \\ \beta h\left(\frac{\varepsilon_0}{\varepsilon}\right)^{1/p}, & \varepsilon < \varepsilon_0, \end{cases} \tag{A.10}$$

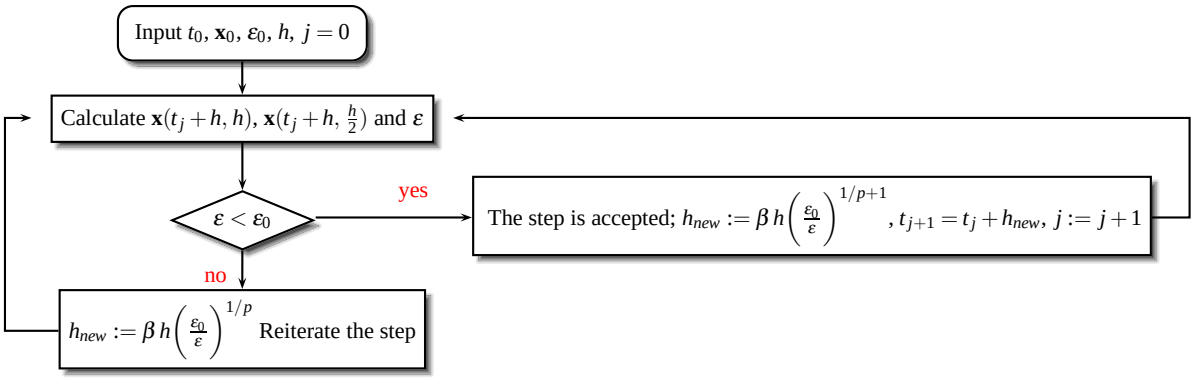

Fig. A.2 Flow diagramm of the adaptive step size control by use of the step doubling method.

where β is a "safety" factor.

Runge-Kutta-Fehlberg method

The alternative stepsize adjustment algorithm is based on the *embedded Runge-Kutta formulas*, originally invented by Fehlberg and is called *the Runge-Kutta-Fehlberg methods (RKF45)* [11, 10]. At each step, two different approximations for the solution are made and compared. Usually an fourth-order method with five stages is used together with an fifth-order method with six stages, that uses all of the points of the first one. The general form of a fifth-order Runge-Kutta with six stages is

$$\begin{aligned} k_1 &= f(t, \mathbf{x}), \\ k_2 &= f(t+\alpha_2 h, \mathbf{x}+h\beta_{21}k_1), \\ &\vdots \\ k_6 &= f(t+\alpha_6 h, \mathbf{x}+h\sum_{j=1}^{5}\beta_{6j}k_j)\,. \end{aligned}$$

The embedded fourth-order formula is

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\sum_{i=1}^{6} c_i k_i + \mathscr{O}(h^5)\,.$$

And a better value for the solution is determined using a Runge-Kutta method of fifth-order:

$$\mathbf{x}^*_{n+1} = \mathbf{x}_n + h\sum_{i=1}^{6} c^*_i k_i + \mathscr{O}(h^6)$$

The two particlular choises of unknown parametrs of the method are given in Tables A.4–A.5.

The error estimate is

$$\varepsilon = |\mathbf{x}_{n+1} - \mathbf{x}^*_{n+1}| = \sum_{i=1}^{6}(c_i - c^*_i)k_i\,.$$

Table A.4 Fehlberg parameters of the Runge-Kutta-Fehlberg 4(5) method.

1/4	1/4					
3/8	3/32	9/32				
12/13	1932/2197	-7200/2197	7296/2197			
1	439/216	-8	3680/513	-845/4104		
1/2	-8/27	2	-3544/2565	1859/4104	-11/40	
	25/216	0	1408/2565	2197/4104	-1/5	
	16/135	0	6656/12825	28561/56430	-9/50	2/55

Table A.5 Cash-Karp parameters of the Runge-Kutta-Fehlberg 4(5) method.

1/5	1/5					
3/10	3/40	9/40				
3/5	3/10	-9/10	6/5			
1	-11/54	5/2	-70/27	35/27		
7/8	1631/55296	175/512	575/13828	44275/110592	253/4096	
	37/378	0	250/621	125/594	512/1771	
	2825/27648	0	18575/48384	13525/55296	277/14336	1/4

As was mentioned above, if we take the current step h and produce an error ε, the corresponding "optimal" step h_{opt} is estimated as

$$h_{opt} = \beta h \left(\frac{\varepsilon_{tol}}{\varepsilon} \right)^{0.2},$$

where ε_{tol} is a desired accuracy and β is a "safety" factor, $\beta \simeq 1$. Then if the two answers are in close agreement, the approximation is accepted. If $\varepsilon > \varepsilon_{tol}$ the step size has to be decreased, whereas the relation $\varepsilon < \varepsilon_{tol}$ means, that the step size are to be increased in the next step.
Using Eq. (A.10), the optimal step can be often written as

$$h_{opt} = \begin{cases} \beta h \left(\frac{\varepsilon_{tol}}{\varepsilon} \right)^{0.2}, & \varepsilon \geq \varepsilon_{tol}, \\ \beta h \left(\frac{\varepsilon_{tol}}{\varepsilon} \right)^{0.25}, & \varepsilon < \varepsilon_{tol}, \end{cases}$$