Variable Step Size Numerical Solvers

Jason Brewer and George Little

May 24, 2002

Abstract

This article will explore how the basic Rung-Kutta method formulas are found and how they are adapted to use variable step size and error control.

1. Introduction

The purpose of developing numerical methods such as the popular Rung-Kutta methods is to approximate the solution to the well posed initial value problem

$$\frac{dy}{dt} = f(t,y), \quad a \leq t \leq b, \quad y(a) = \alpha \tag{1}$$

which has an exact solution $y(t)$.

2. Taylor's Method

When developing a numerical method, the first step is to put the exact solution into the form of a Taylor Power Series

$$y(t) = y(a) + y'(a)(t-a) + \frac{y''(a)}{2!}(t-a)^2 + \cdots + \frac{y^{(n)}(a)}{n!}(t-a)^n + \cdots \tag{2}$$

We next set $t = a + h$ and perform the following steps:

$$y(a+h) = y(a) + y'(a)(a+h-a) + \frac{y''(a)}{2!}(a+h-a)^2 + \cdots + \frac{y^{(n)}(a)}{n!}(a+h-a)^n + \cdots,$$
$$y(a+h) = y(a) + y'(a)h + \frac{y''(a)}{2!}h^2 + \cdots + \frac{y^{(n)}(a)}{n!}h^n + \cdots.$$

Since a is the current working t value, it can be expressed that $a = t_i$. Because $t_i + h$ is just the next t, it can be expressed that $t_i + h = t_{i+1}$. These transformations end up giving us the Taylor series in the form

$$y(t_{i+1}) = y(t_i) + y'(t_i)h + \frac{y''(t_i)}{2!}h^2 + \cdots + \frac{y^{(n)}(t_i)}{n!}h^n + \cdots. \tag{3}$$

We next change the infinite Taylor series into a finite series by finding a ξ_i where $t_i < \xi_i < t_{i+1}$ such that

$$y(t_i + h) = y(t_i) + y'(t_i)h + \frac{y''(t_i)}{2!}h^2 + \cdots + \frac{y^{(n)}(t_i)}{n!}h^n + \frac{y^{(n+1)}(\xi)}{(n+1)!}h^{n+1}, \tag{4}$$

giving an exact solution with a finite number of terms. Since $y(t)$ is an exact solution, we get the following:

$$\begin{aligned} y'(t) &= f(t,y), \\ y''(t) &= f'(t,y), \\ &\vdots \\ y^{(n)}(t) &= f^{(n-1)}(t,y). \end{aligned}$$

By replacing the y's with the f's, and by dropping the $n+1$ term we get the equation

$$w_{i+1} = w_i + hf(t_i, w_i) + \frac{h^2}{2!}f'(t_i, w_i) + \cdots + \frac{h^n}{n!}f^{(n-1)}(t_i, w_i), \tag{5}$$

where w is an approximate solution, h is the step size, and t is the time value. The w's replace all instances of the y's because when the $n+1$ term is dropped, the equation becomes an approximation.

By factoring out an h from equation (5), we get

$$w_{i+1} = w_i + h[f(t_i, w_i) + \frac{h}{2!} f'(t_i, w_i) + \cdots + \frac{h^{n-1}}{n!} f^{(n-1)}(t_i, w_i)]. \tag{6}$$

And by setting

$$\Phi(t_i, w_i) = T^{(n)}(t_i, w_i) = f(t_i, w_i) + \frac{h}{2!} f'(t_i, w_i) + \cdots + \frac{h^{n-1}}{n!} f^{(n-1)}(t_i, w_i) \tag{7}$$

we get the difference equation in the form of

$$w_{i+1} = w_i + h\Phi(t_i, w_i). \tag{8}$$

This difference equation is known as Taylor's Method of Order n. By increasing n, the order of the Taylor method being used increases, and the error being made by the method decreases. To see this error reduction, we will use Taylor methods of order 1,2, and 4 on the initial value problem

$$y' = -y + t + 1, 0 \leq t \leq 1, y(0) = 1 \tag{9}$$

2.1. First Order Taylor Method

The First Order Taylor Method

$$w_{i+1} = w_i + hf(w_i, t_i), \tag{10}$$

also known as Euler's Method, for equation (9) can be expressed as

$$\begin{aligned} w_0 &= 1, \\ w_{i+1} &= w_i + h(-w_i + t_i + 1). \end{aligned} \tag{11}$$

By calculating the First Order Taylor's Method with a fixed step size of $h = 0.1$, we get the results shown in Table 1.

2.2. Second Order Taylor Method

To express the second order Taylor Method

$$w_{i+1} = w_i + h[f(w_i, t_i) + \frac{h}{2} f'(w_i, t_i)] \tag{12}$$

t	Exact value	Euler's Method	Error
0.0	1.000000	1.000000	0
0.1	1.004837	1.000000	4.837×10^{-3}
0.2	1.018730	1.010000	8.731×10^{-3}
0.3	1.040818	1.029000	1.182×10^{-2}
0.4	1.070320	1.056100	1.422×10^{-2}
0.5	1.106530	1.090490	1.604×10^{-2}
0.6	1.148811	1.131441	1.737×10^{-2}
0.7	1.196585	1.178297	1.829×10^{-2}
0.8	1.249328	1.230467	1.887×10^{-2}
0.9	1.306569	1.287420	1.915×10^{-2}
1.0	1.367879	1.348678	1.920×10^{-2}

Table 1: First Order Taylor Method

for the initial value problem (9) the derivative of $f(t, y(t)) = -y+t+1$ is needed to complete the second order Taylor Method.

$$\begin{aligned} f'(t, y(t)) &= \frac{d}{dt}(-y+t+1) \\ &= -\frac{dy}{dt} + 1 \\ &= (-y+t+1)+1 \\ &= y-t-1+1 \\ &= y-t \end{aligned}$$

To complete the second order Taylor Method, $f'(t,y)$ must be substituted into expression (11) and then simplified.

$$
\begin{aligned}
w_{i+1} &= w_i + h[f(w_i,t_i) + \frac{h}{2}f'(w_i,t_i)] \\
w_{i+1} &= w_i + h[-w_i + t_i + 1 + \frac{h}{2}(w_i - t_i)] \\
w_{i+1} &= w_i + h[-w_i + t_i + 1 + \frac{h}{2}w_i - \frac{h}{2}t_i] \\
w_{i+1} &= w_i + h[-w_i + \frac{h}{2}w_i + t_i - \frac{h}{2}t_i + 1] \\
w_{i+1} &= w_i + h[-w_i(1 - \frac{h}{2}) + t_i(1 - \frac{h}{2}) + 1] \\
w_{i+1} &= w_i + h[(1 - \frac{h}{2})(t_i - w_i) + 1]
\end{aligned}
$$

Giving us the Second Order Taylor's Method

$$
\begin{aligned}
w_0 &= 1, \\
w_{i+1} &= w_i + h[(1 - \frac{h}{2})(t_i - w_i) + 1].
\end{aligned} \tag{13}
$$

By calculating the Second Order Taylor's Method with a fixed step size of $h = 0.1$, we get the results shown in Table 2.

2.3. Fourth Order Taylor Method

To express the fourth order Taylor Method

$$
w_{i+1} = w_i + h[f(w_i,t_i) + \frac{h}{2}f'(w_i,t_i) + \frac{h^2}{3!}f''(w_i,t_i) + \frac{h^3}{4!}f'''(w_i,t_i)] \tag{14}
$$

t	Exact value	Taylor's Method Order Two	Error
0.0	1.000000	1.000000	0
0.1	1.004837	1.005000	1.626×10^{-4}
0.2	1.018730	1.019025	2.942×10^{-4}
0.3	1.040818	1.041218	3.998×10^{-4}
0.4	1.070320	1.070802	4.820×10^{-4}
0.5	1.106530	1.107076	5.453×10^{-4}
0.6	1.148811	1.149404	5.924×10^{-4}
0.7	1.196585	1.197211	6.257×10^{-4}
0.8	1.249328	1.249976	6.470×10^{-4}
0.9	1.306569	1.307227	6.583×10^{-4}
1.0	1.367879	1.368541	6.616×10^{-4}

Table 2: Second Order Taylor Method

for the initial value problem (9) the first, second, and third derivatives of $f(t, y(t)) = -y + t + 1$ are needed to complete the fourth order Taylor Method.

$$\begin{aligned} f'(t, y(t)) &= \frac{d}{dt}(-y + t + 1) \\ &= -\frac{dy}{dt} + 1 \\ &= -(-y + t + 1) + 1 \\ &= y - t - 1 + 1 \\ &= y - t \end{aligned}$$

$$\begin{aligned} f''(t, y(t)) &= \frac{d}{dt}(y - t) \\ &= -\frac{dy}{dt} - 1 \\ &= -(-y + t + 1) - 1 \\ &= -y + t \end{aligned}$$

$$
\begin{aligned}
f'''(t, y(t)) &= \frac{d}{dt}(-y + t) \\
&= -\frac{dy}{dt} + 1 \\
&= -(-y + t + 1) + 1 \\
&= y - t - 1 + 1 \\
&= y - t
\end{aligned}
$$

To complete the fourth order Taylor Method, $f'(t, y)$, $f''(t, y)$, and $f'''(t, y)$ must be subbed into expression (14) and then simplified.

$$
\begin{aligned}
w_{i+1} &= w_i + h[f(w_i, t_i) + \frac{h}{2} f'(w_i, t_i) + \frac{h^2}{3!} f''(w_i, t_i) + \frac{h^3}{4!} f'''(w_i, t_i)] \\
w_{i+1} &= w_i + h[-w_i + t_i + 1 + \frac{h}{2}(w_i - t_i) + \frac{h^2}{6}(-w_i + t_i) + \frac{h^3}{24}(w_i - t_i)] \\
w_{i+1} &= w_i + h[(t_i - w_i) - \frac{h}{2}(t_i - w_i) + \frac{h^2}{6}(t_i - w_i) - \frac{h^3}{24}(t_i - w_i) + 1] \\
w_{i+1} &= w_i + h[(1 - \frac{h}{2} + \frac{h^2}{6} - \frac{h^3}{24})(t_i - w_i) + 1]
\end{aligned}
$$

Giving us the Fourth Order Taylor's Method

$$
\begin{aligned}
w_0 &= 1, \\
w_{i+1} &= w_i + h[(1 - \frac{h}{2} + \frac{h^2}{6} - \frac{h^3}{24})(t_i - w_i) + 1].
\end{aligned}
\tag{15}
$$

By calculating the Fourth Order Taylor's Method with a fixed step size of $h = 0.1$, we get the results shown in Table 3.

2.4. Error of Taylor Methods

By analyzing tables 1,2 and 3 we can see that the different orders of Taylor's method are creating very different values for the error they make. The first order Taylor Method consistently makes an error of approximately 10^{-2}, which we will call first order error. The second order Taylor Method consistently makes an error of approximately 10^{-4}, showing that the second order Taylor Method has second order error. It can also be seen that the fourth order Taylor Method has a consistent error of about 10^{-7} giving it fourth order error.

t	Exact value	Taylor's Method Order Four	Error
0.0	1.000000	1.000000	0
0.1	1.004837	1.004837	8.200×10^{-8}
0.2	1.018730	1.018730	1.483×10^{-7}
0.3	1.040818	1.040818	2.013×10^{-7}
0.4	1.070320	1.070320	2.429×10^{-7}
0.5	1.106530	1.106530	2.747×10^{-7}
0.6	1.148811	1.148811	2.983×10^{-7}
0.7	1.196585	1.196585	3.149×10^{-7}
0.8	1.249328	1.249329	3.256×10^{-7}
0.9	1.306569	1.306569	3.125×10^{-7}
1.0	1.367879	1.367879	3.332×10^{-7}

Table 3: Fourth Order Taylor Method

3. Rung-Kutta Methods

While the error control offered by the higher order Taylor Methods is good, there is a price for this increased error control. As the order increases, so do the number of higher order derivatives that must be taken. Even for relatively simple equations, the fourth and fifth derivatives can become very difficult. It is for this reason that a way to eliminate the higher order derivatives while retaining the higher order error control is sought out. The Rung-Kutta methods do just that, they eliminate the need to take higher order derivatives while retaining the higher order error control.

3.1. Local Truncation Error (LTE)

The Local Truncation Error (LTE) is the amount by which the Difference Equation (8) is not satisfied when the approximated values are replaced by the exact values. Take the Difference Equation using approximated values:

$$w_{i+1} = w_i + h\Phi(t_i, w_i).$$

By moving the w_{i+1} term to the right side of the equation and giving

$$0 = w_i + h\Phi(t_i, w_i) - w_{i+1},$$

Introduction
Taylor's Method
Rung-Kutta Methods
Variable Step Size
Putting it all Together
Home Page
Title Page
Page 8 of 15
Go Back
Full Screen
Close
Quit

and by replacing the approximated values with the exact values,we get the equation for the LTE:

$$\tau_{n+1} = y(t_n) + h\Phi(t_n, y(t_n), h) - y(t_{n+1}). \tag{16}$$

By equation (3) and by setting

$$\Delta(t_n, y(t_n), h) = y'(t_i) + \frac{h}{2!}y''(t_i) + \cdots + \frac{h^{n-1}}{n!}y^{(n)}(t_i) + \cdots,$$

y_{n+1} can be represented by the difference equation

$$y(t_{n+1}) = y(t_n) + h\Delta(t_n, y(t_n), h). \tag{17}$$

By equations (16) and (17) the LTE can be written as

$$\tau_{n+1} = y(t_n) + h\Phi(t_n, y(t_n), h) - (y(t_n) + h\Delta(t_n, y(t_n), h)), \tag{18}$$

yielding

$$\tau_{n+1} = h[\Phi(t_n, y(t_n), h) - \Delta(t_n, y(t_n), h)]. \tag{19}$$

3.2. The General Explicit RK Formula

The general explicit RK Formula is represented as

$$w_{n+1} = w_n + h_n \sum_{i=1}^{s} b_i f_i, \tag{20}$$

where

$$f_1 = f(t_n, w_n),$$
$$f_i = f(t_n, c_i h_n, w_n + h_n \sum_{j=1}^{i-1} a_{ij} f_j),$$
$$i = 2, 3, \ldots, s.$$

It is important to note that as the number of stages s increase, the order of the RK method increases with it. For example, a two stage RK formula is second order while a three stage RK formula is third order.

3.3. Two stage RK Formula

Using the General Explicit RK Formula with $s = 2$ we get a two stage RK formula.

$$w_{n+1} = w_n + h_n \sum_{i=1}^{2} b_i f_i,$$

where

$$\begin{aligned} f_1 &= f(t_n, w_n), \\ f_2 &= f(t_n + c_2 h_n, w_n + h_n a_{21} f_1), \\ w_{n+1} &= w_n + h_n(b_1 f_1 + b_2 f_2). \end{aligned}$$

Now that a two stage Rung-Kutta Formula has been defined, the coefficients b_1, b_2, a_{21}, and c_2 must be defined to give us a specific Rung-Kutta formula. This is done by first seeing that by the difference equation (8)

$$\begin{aligned} \Phi &= b_1 f_1 + b_2 f_2, \\ \Phi &= b_1 f_1 + b_2 f(t_n + c_2 h_n, w_n + h_n a_{21} f_1). \end{aligned}$$

Next by expanding Φ into a Taylor Series of two variables, we get

$$\begin{aligned} \Phi &= b_1 f + b_2[f + (c_2 h f_t + a_{21} h f f_y) + \frac{1}{2}(c_2^2 h^2 f_{xx} + 2a_{21} h c_2 h f f_{xy} + a_{21}^2 h^2 f^2 f_{yy}) + O(h^3)], \\ \Phi &= (b_1 + b_2) f + b_2[h(c_2 f_x + a_{21} f f_y) + \frac{h^2}{2}(c_2^2 f_{xx} + 2a_{21} c_2 f f_{xy} + a_{21}^2 f^2 f_{yy}) + O(h^3)]. \end{aligned}$$

In order to simplify this equation we must set $a_{21} = c_2$, giving us:

$$\begin{aligned} \Phi &= (b_1 + b_2) f + b_2[h(c_2 f_x + c_2 f f_y) + \frac{h^2}{2}(c_2^2 f_{xx} + 2c_2 c_2 f f_{xy} + c_2^2 f^2 f_{yy}) + O(h^3)], \\ \Phi &= (b_1 + b_2) f + b_2[h(c_2 f_x + c_2 f f_y) + \frac{h^2}{2}(c_2^2 f_{xx} + 2c_2^2 f f_{xy} + c_2^2 f^2 f_{yy}) + O(h^3)], \\ \Phi &= (b_1 + b_2) f + b_2 c_2 (f_x + f f_y) + \frac{1}{2} b_2 c_2^2 h^2 (f_{xx} + 2 f f_{xy} + f^2 f_{yy}) + O(h^3), \\ \Phi &= (b_1 + b_2) y'(t) + b_2 c_2 y''(t) + \frac{1}{2} b_2 c_2^2 h^2 y'''(t) + O(h^3). \end{aligned}$$

By taking the LTE from equation (19), we get

$$\begin{aligned}\tau_{n+1} &= h(\Phi - \Delta) \\ &= h[(b_1 + b_2) - 1]y'(t) + h^2[b_2c_2 - \frac{1}{2}]y''(t) + h^3[\frac{1}{2}b_2c_2^2 - \frac{1}{6}]y''(t) + O(h^4).\end{aligned}$$

To find the order of the LTE we must now set as many of the lower order terms as possible to zero. Because this is a two stage Rung-Kutta method, only the first two terms can be set to zero. Doing this we get the following equations for the coefficients

$$b_1 + b_2 = 1, \tag{21}$$

$$b_2c_2 = \frac{1}{2}. \tag{22}$$

Since there are three parameters b_1, b_2, and c_3 to satisfy only two conditions, there are an infinite number of possible solutions. By choosing different sets of coefficients, different second order Rung-Kutta methods can be found. Because there are three parameters to satisfy to conditions, one of the parameters must be chosen in an independent manner. One possible choice is to make $c_2 = 1$, giving

$$\begin{aligned}b_2c_2 &= \frac{1}{2}, \\ b_2(1) &= \frac{1}{2}, \\ b_2 &= \frac{1}{2},\end{aligned}$$

and

$$\begin{aligned}b_1 + b_2 &= 1, \\ b_1 + \frac{1}{2} &= 1, \\ b_1 &= \frac{1}{2}.\end{aligned}$$

These parameters now give a Rung-Kutta Method of Order 2

$$w_{n+1} = w_n + \frac{h}{2}(f_1 + f_2) \tag{23}$$

where

$$f_1 = f(t_n, w_n)$$
$$f_2 = f(t_n + h, y_n + hf_1)$$

3.4. Three Stage RK Formula

The derivation of a Three Stage RK Formula follows the same ideas presented in the previous section. Once a three stage formula has been found, the coefficients must now be found by expanding Φ into a Taylor Series with the main difference being that there is an extra term. Equation (19) is now used to find the LTE and the order of the RK method. And by setting as many of the lower order terms as possible to zero, the coefficients of the RK method can be found.

4. Variable Step Size

The idea behind using a variable step size solver is to take the minimal number of points while keeping the error made by the solver within a specified tolerance. The first step is to determine the error made at t_n by using a step size h_n and adjust the step size accordingly. Secondly, the solver routine must predict an initial step size for the iteration of t_{n+1}.

4.1. Error Control

Calculating the error made by a RK formula of s stages requires the computation of the same y_{n+1} and step size with an RK formula of $s+1$ stages. Taking the difference of a RK2 and RK3 methods will result in an approximation of the error made by the RK2 method. Where y_{n+1} a two stage RK formula, $\hat{y}_{n+1}$ is a three stage RK formula, and δ_{n+1} is the error made by the y_{n+1} formula, giving us the equation

$$\delta_{n+1} = y_{n+1} - \hat{y}_{n+1}. \tag{24}$$

If the error δ_{n+1} is greater than the allowable tolerance a new step size needs to be calculated and the routine repeated. The new h value can be calculated using a step size predictor shown in the next section.

4.2. Step Size Prediction

The step size predictor routine itself uses the difference between two difference equations using Rung-Kutta Methods of order s and $s+1$.

$$\begin{aligned} w_0 &= \alpha \\ w_{i+1} &= w_i + h_i \Phi(t_i, w_i, h_i) \end{aligned} \tag{25}$$

has local truncation error τ_{i+1} of order $O(h^n)$, while

$$\begin{aligned} \hat{w}_0 &= \alpha \\ \hat{w}_{i+1} &= \hat{w}_i + h_i \hat{\Phi}(t_i, \hat{w}_i, h_i) \end{aligned} \tag{26}$$

has local truncation error τ_{i+1} of order $O(h^{n+1})$. By using the LTE formula in equation (16) and replacing the exact solution with the difference equation of order $n+1$, an approximation of the LTE is produced in the form of

$$\tau_{n+1} = w(t_n) + h\Phi(t_n, w(t_n), h) - [\hat{w}(t_n) + h\hat{\Phi}(t_n, \hat{w}(t_n), h)] \tag{27}$$

which can be rewritten as

$$\tau_{n+1} \approx |w_{i+1} - \hat{w}_{i+1}|. \tag{28}$$

Since τ_{i+1} is of order $O(h^n)$, a constant k exists with

$$\tau_{i+1} \approx kh^n. \tag{29}$$

The relationship between equations (28) and (29) imply that

$$kh^n \approx |w_{i+1} - \hat{w}_{i+1}|. \tag{30}$$

The object now is to use this estimate to choose an appropriate step size. To accomplish this, consider the truncation error with h replaced by qh, where q is positive but bounded above and away from zero. This gives us

$$\tau_{i+1} \approx k(qh)^n = q^n(kh^n) \approx q^n|w_{i+1} - \hat{w}_{i+1}|. \tag{31}$$

To bound $\tau i + 1$ by the error tolerance ε, we choose q so that

$$q^n|w_{i+1} - \hat{w}_{i+1}| \approx |\tau_{i+1}| \le \varepsilon, \tag{32}$$

which easily becomes

$$q^n|w_{i+1} - \hat{w}_{i+1}| \leq \varepsilon. \tag{33}$$

Solving the inequality for q we get

$$q \leq (\frac{\varepsilon}{|w_{i+1} - \hat{w}_{i+1}|})^{\frac{1}{n}}, \tag{34}$$

where q value found here is a scalar used to find the new step size by multiplying the old h by q, giving us the new h value.

5. Putting it all Together

To create a Rung-Kutta method variable step size solver using the methods shown here, you must follow these general steps:

1. You must first define an error tolerance, two Rung-Kutta methods of order n and $n+1$,an initial h_n, and the range of t that the solution is to be evaluated over.
2. Evaluate the initial value problem using both RK methods using t_n, w_n, h_n.
3. Use the results of the two Rung-Kutta Methods to find the error made by using equation (25). If the results of (25) meet the error criteria, then the solver increments t and uses the q value found in equation (34) to find the next step size. If the new t is out of the bounds of the initial value problem, the solver stops, otherwise it repeats step 2.

   If the results of (25) do not meet the error criteria, then the solver uses the q value found in equation (34) to calculate a new step size and repeates step 2 without incrementing t.

The resulting t and w values that the solver produce at each iteration when the error made meets the error tolerance are typically plotted on an (x, y) graph, giving a graphical representation of the solution to the initial value problem.

References

[1] Arnold, Dave *Technical Help and Advice*

[2] Brown, Greg *Program Inspiration*

[3] Burden, Richard L., J. Douglas Faires ***Numerical Analysis: Fourth Edition***, PWS-KENT Publishing Company, Boston: 1993

[4] Dormand, John R., ***Numerical Methods for Differential Equations: A Computational Approach*** CRC Press, Boca raton: 1996

[5] Romero, Chris *C++ Technical Help and Advice*