5.4. RUNGE-KUTTA METHODS

As usual we are trying to find a numerical solution to the initial value problem

$$y' = f(t,y), \;\; y(t_0) = y_0 \tag{1}$$

Runge-Kutta methods are a class of nonlinear methods of the form

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i k_i \tag{2}$$

$$k_i = f\left(t_n + c_i h, \;\; y_n + h\sum_{j=1}^{s} a_{ij} k_j \right), \;\; i = 1,2,...,s \tag{3}$$

subject to the constraints

$$c_i = \sum_{j=1}^{s} a_{ij} \tag{4}$$

$$\sum_{i=1}^{s} b_i = 1 \tag{5}$$

where $0 \le a_{ij}, b_j, c_j \le 1$.

An alternative and equivalent way of writing (2) and (3) is

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i f(t_n + c_i h, \;\; Y_i) \tag{6}$$

$$Y_i = y_n + h\sum_{i=1}^{s} a_{ij} f(t_n + c_j h, \;\; Y_j), \;\; i = 1,2,...,s \tag{7}$$

In general, the c_j will be less than one and therefore the function $f(t,y)$ needs to be evaluated at a grid spacing smaller than h, i.e., at $t_n + c_j h$ for $j = 1,2,...,s$.

The coefficients for a Runge-Kutta method are generally written in the form of the following ***Butcher Array***,

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & & & a_{2s} \\ \vdots & \vdots & & & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array} \tag{8}$$

From equation (4), the c_i are the row sums of the (a_{ij}) matrix.

Runge Kutta methods are implicit unless $a_{ij} = 0,\ j \ge i$. We will only be looking at <u>*explicit Runge-Kutta methods, for which the matrix A is strictly lower triangular*</u>.

The Butcher Array for an explicit method is written as follows:

$$\begin{array}{c|cccc} c_1 & 0 & 0 & \cdots & 0 \\ c_2 & a_{21} & 0 & & \vdots \\ \vdots & \vdots & \ddots & & \vdots \\ c_s & a_{s1} & \cdots & a_{s,s-1} & 0 \\ \hline & b_1 & b_2 & \cdots & b_s \end{array} \tag{9}$$

In general, the zeroes in the upper-triangular part of (a_{ij}) are omitted; furthermore, by equation (4), $c_1 = 0$ for an explicit matrix.

$$\begin{array}{c|cccc} 0 & & & & \\ c_2 & a_{21} & & & \\ \vdots & \vdots & \ddots & & \\ c_s & a_{s1} & \cdots & a_{s,s-1} & \\ \hline & b_1 & b_2 & \cdots & b_s \end{array} \tag{10}$$

The number *s* is referred to as the number of ***stages*** in the method. This is because there must be *s* function evaluations between each pair of grid-points (see equation 7).

<u>**Explicit Two-Stage Runge-Kutta Methods**</u>

The Butcher array for an explicit two-stage method can then be written as

$$\begin{array}{c|cc} c_1 & & \\ c_2 & a_{21} & \\ \hline & b_1 & b_2 \end{array} \tag{11}$$

By equations (4) and (5), this becomes

$$\begin{array}{c|cc} 0 & & \\ a & a & \\ \hline & b & 1-b \end{array} \tag{12}$$

Thus there are only two independent parameters.

To determine these parameters, we explicitly write out the method. From equation (2)

$$y_{n+1} = y_n + h(b_1 k_1 + b_2 k_2) = y_n + h(bk_1 + (1-b)k_2) \tag{13}$$

while from equation (3),

$$\begin{aligned} k_i &= f\left(t_n + c_i h,\ y_n + h(a_{i1}k_1 + a_{i2}k_2)\right),\ i = 1,2 \\ &= f(t_n + c_i h,\ y_n + ha_{i1}k_2) \end{aligned} \tag{14}$$

because $a_{i2} = 0$ in (12). Hence

$$\begin{aligned} k_1 &= f(t_n,\ y_n) = f_n \\ k_2 &= f(t_n + ah,\ y_n + hak_1) = f(t_n + ah,\ y_n + haf_n) \end{aligned} \tag{15}$$

Substituting (15) into (13) gives

$$y_{n+1} = y_n + h(bf_n + (1-b)f(t_n + ah,\ y_n + haf_n)) \tag{16}$$

We don't have any constraints on *a* and be except that they be in the interval [0, 1], but we can make some observations about special values:

If $a = 0$, we get Euler's method, regardless of the value of *b*:

$$\begin{aligned} y_{n+1} &= y_n + h(bf_n + (1-b)f(t_n,\ y_n)) = y_n + h(bf_n + (1-b)f_n) \\ &= y_n + hf_n \end{aligned} \tag{17}$$

If $a = 1$, we get a rule that looks very much like the theta-method:

$$y_{n+1} = y_n + h(bf_n + (1-b)f(t_n + h,\ y_n + hf_n)) \tag{18}$$

The only difference between (18) and the theta-method is the second argument of f, which for the theta method is y_{n+1} instead of $y_n + hf_n$. In this case, the second argument is an Euler's method approximation to y_{n+1}, not y_{n+1} itself.

If $a = 1$ and $b = 1/2$ we get the Improved Euler Method:

$$y_{n+1} = y_n + \frac{1}{2}h(f_n + f(t_n + h,\ y_n + hf_n)) \tag{19}$$

We can ask the following question: what is (are) the best s-stage method(s)? Ideally it would be the one with the highest order local truncation error.

<u>Local Truncation Error for Runge-Kutta Methods</u>

From Taylor's theorem,

$$y(t_{n+1}) = y(t_n) + hy'(t_n) + \frac{1}{2}h^2 y''(t_n) + \frac{1}{6}h^3 y'''(t_n) + O(h^4) \tag{20}$$

From equation (1), $y(t_n) = f_n$, and (omitting the "n" subscript), using the chain rule the second derivative is

$$y'' = \frac{dy'}{dt} = \frac{df}{dt} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial y}\frac{dy}{dt} = f_t + f_y y' = f_t + f_y f \tag{21}$$

The third derivative requires use first of the product rule and then the chain rule. First, we apply the product rule:

$$y''' = \frac{dy''}{dt} = \frac{d}{dt}(f_t + f_y f) = \frac{df_t}{dt} + f_y\frac{df}{dt} + f\frac{df_y}{dt} \tag{22}$$

The derivative df/dt in the middle term is exactly what we calculated in (21). Substituting that gives

$$y''' = \frac{df_t}{dt} + f_y(f_t + f_y f) + f\frac{df_y}{dt} \tag{23}$$

The other derivatives in (23) are calculated with the chain rule:

$$\frac{df_t}{dt} = f_{tt} + f_{ty}y' = f_{tt} + f_{ty}f \tag{24}$$

$$\frac{df_y}{dt} = f_{ty} + f_{yy}y' = f_{ty} + f_{yy}f \tag{25}$$

Substituting (24) and (25) into (23) gives

$$\begin{aligned} y''' &= f_{tt} + f_{ty}f + f_y(f_t + f_yf) + f(f_{ty} + f_{yy}f) \\ &= f_{tt} + 2f_{ty}f + f^2 f_{yy} + f_y(f_t + f_yf) \end{aligned} \tag{26}$$

To simplify the notation make the following substitutions:

$$F = f_t + f_yf, \quad G = f_{tt} + 2f_{ty} + f^2 f_{yy} \tag{27}$$

Equation (26) becomes

$$y''' = G + f_yF \tag{28}$$

and equation (22) becomes

$$y'' = F \tag{29}$$

Then the Taylor expansion in equation (20) is

$$y(t_{n+1}) = y_n + hf_n + \frac{1}{2}h^2F_n + \frac{1}{6}h^3(G_n + f_{y,n}F_n) + O(h^4) \tag{30}$$

assuming that $y(t_n) = y_n$, and elsewhere the subscript "*n*" indicates, as usual, evaluation at (t_n, y_n)

Lets drop the constraint (5) that the *b*'s sum to 1. For a two-stage method, from equation (16), and Taylor's theorem

$$\begin{aligned} y_{n+1} &= y_n + h(b_1f_n + b_2f(t_n + ah,\ y_n + haf_n)) \\ &= y_n + hb_1f_n + hb_2\Big[f_n + ahf_{t,n} + ahf_nf_{y,n} + \\ &\quad \frac{1}{2}[(ah)^2f_{tt,n} + 2(ah)^2f_nf_{ty,n} + (ahf_n)^2f_{yy,n}] + O(h^3)\Big] \end{aligned} \tag{31}$$

where we have used the 2-variable Taylor expansion in equation (31)

$$\begin{aligned} f(x+h,\ y+k) &= f(x,\ y) + hf_x(x,y) + kf_y(x,y) \\ &\quad + \frac{1}{2}[h^2f_{xx}(x,y) + 2hkf_{xy}(x,y) + k^2f_{yy}(x,y)] + O(h^3) \end{aligned} \tag{32}$$

Using the notation of equation (27) in (31) gives

$$\begin{aligned} y_{n+1} &= y_n + hb_1f_n + hb_2[f_n + ahF_n + \frac{1}{2}(ah)^2G_n + O(h^3)] \\ &= y_n + hb_1f_n + hb_2f_n + h^2ab_2F_n + \frac{1}{2}a^2b_2h^3G_n + O(h^4) \end{aligned} \tag{33}$$

The local truncation error for a two-stage RK method is then

$$\begin{aligned}LTE &= \frac{y_{n+1} - y(t_{n+1})}{h}\\ &= \frac{1}{h}\Bigg[y_n + hf_n + \frac{1}{2}h^2F_n + \frac{1}{6}h^3(G_n + f_{y,n}F_n) + O(h^4)\\ &\quad -y_n - hb_1f_n - hb_2f_n - h^2ab_2F_n - \frac{1}{2}a^2b_2h^3G_n + O(h^4)\Bigg]\\ &= f_n[1-b_1-b_2] + \frac{1}{2}hF_n[1-2ab_2] + O(h^2)\end{aligned} \tag{34}$$

The necessary condition for the method to be $O(h =)$ is precisely the constraint (4), that

$$b_1 + b_2 = 1 \tag{35}$$

Let

$$b_1 = b,\ b_2 = 1-b. \tag{36}$$

Furthermore, the condition for the method to be $O(h^2)$is that

$$0 = 1 - 2ab_2 = 1 - 2a(1-b) \tag{27}$$

or

$$a(1-b) = \frac{1}{2} \tag{30}$$

Thus there are an infinite number of possible 2-stage RK-methods that are second-order, given by

$$y_{n+1} = y_n + \frac{h}{2a}[(2a-1)f_n + f(t_n + ah,\ y_n + haf_n)] \tag{31}$$

Since

$$1-b = \frac{1}{2a},\quad b = 1 - \frac{1}{2a} = \frac{2a-1}{2a} \tag{32}$$

Observe that from equation (30), $a \neq 0$ so the definition (31), (32) makes sense. Any value of $0 < a \leq 1$ will work, although values close to zero can lead to computational problems because of divide-overflows near zero. Some of the two-stage second-order methods have special names:

$a = 1/2$: Polygon method:

$$y_{n+1} = y_n + hf(t_n + h/2,\ y_n + (h/2)f_n) \tag{31}$$

$a = 1$: Improved Euler' method:

$$y_{n+1} = y_n + (h/2)(f_n + f(t_n + h,\ y_n + hf_n)) \tag{32}$$

Local Truncation Error for Three-Stage Methods.

Again, we will drop the restriction on that the *b*'s sum to 1 and see if we can get constraints to obtain higher order methods. The general 3-stage method is

$$y_{n+1} = y_n + h(b_1 k_1 + b_2 k_2 + b_3 k_3) \tag{33}$$

$$k_i = f(t_n + c_i h,\ y_n + h(a_{i1}k_1 + a_{i2}k_2 + a_{i3}k_3)) \tag{34}$$

subject to the constraint

$$c_i = a_{i1} + a_{i2} + a_{i3} \tag{35}$$

The general lower-diagonal 3-stage Butcher array looks like this

$$\begin{array}{c|ccc} c_1 & & & \\ c_2 & a_{21} & & \\ c_3 & a_{31} & a_{32} & \\ \hline & b_1 & b_2 & b_3 \end{array} \tag{36}$$

This and the observation (from (35)) that $c_1 = 0$ simplifies (34) to

$$k_1 = f(t_n,\ y_n) = f_n \tag{37}$$

$$k_2 = f(t_n + c_2 h,\ y_n + h a_{21} f_n) \tag{38}$$

$$k_3 = f(t_n + c_3 h,\ y_n + h(a_{31} f_n + a_{32} k_2)) \tag{39}$$

From (35), in addition to $c_1 = 0$ we also have $a_{21} = c_2,\ \ a_{31} + a_{32} = c_3$, so that the Butcher array in (36) becomes

$$\begin{array}{c|ccc} 0 & & & \\ a_{21} & a_{21} & & \\ a_{31} + a_{32} & a_{31} & a_{32} & \\ \hline & b_1 & b_2 & b_3 \end{array} \tag{40}$$

Renaming the coefficients $\alpha = a_{21},\ \beta = a_{31},\ \gamma = a_{32}$ we can rewrite (40) as

$$\begin{array}{c|ccc} 0 & & & \\ \alpha & \alpha & & \\ \beta + \gamma & \beta & \gamma & \\ \hline & b_1 & b_2 & b_3 \end{array} \tag{41}$$

With this notation, and omitting the subscript "*n*" in the following steps where the meaning is clear (to shorten the notation), equations (38) and (39) become

$$k_2 = f(t + \alpha h,\ y + \alpha h f) \tag{42}$$

$$k_3 = f(t + h(\beta + \gamma),\ \ y + h(\beta f + \gamma k_2)) \tag{43}$$

Expanding k_2 in a Taylor series about (t, y) to $O(h^3)$ leads to

$$\begin{aligned}
k_2 &= f(t+\alpha h,\ y+\alpha hf) \\
&= f + \alpha hf_t + \alpha hff_y \\
&\quad + \frac{1}{2}(\alpha^2h^2f_{tt} + 2\alpha^2h^2ff_{ty} + \alpha^2h^2f^2f_{yy}) + O(h^3) \\
&= f + \alpha h(f_t + ff_y) + \frac{1}{2}\alpha^2h^2(f_{tt} + 2ff_{ty} + f^2f_{yy}) + O(h^3) \\
&= f + \alpha hF + \frac{1}{2}\alpha^2h^2G + O(h^3)
\end{aligned} \tag{43}$$

where equation (27) has been used in the last step.

Similarly, when we expand k_3 (from equation 43) about the point (t, y) we get

$$\begin{aligned}
k_3 &= f(t+h(\beta+\gamma),\quad y+h(\beta f+\gamma k_2)) \\
&= f + h(\beta+\gamma)f_t + h(\beta f+\gamma k_2)f_y \\
&\quad + \frac{1}{2}\left(h^2(\beta+\gamma)^2f_{tt} + 2h^2(\beta+\gamma)(\beta f+\gamma k_2)f_{ty} + h^2(\beta f+\gamma k_2)^2f_{yy}\right) \\
&\quad + O(h^3)
\end{aligned} \tag{44}$$

The expression $\beta f + \gamma k_2$ which appears multiple times in (44) can be written as

$$\begin{aligned}
\beta f + \gamma k_2 &= \beta f + \gamma(f + \alpha hF + \frac{1}{2}\alpha^2h^2G + O(h^3)) \\
&= (\beta+\gamma)f + \alpha\gamma hF + \frac{1}{2}\gamma\alpha^2h^2G + O(h^3)
\end{aligned} \tag{45}$$

Furthermore, squaring (45) and keeping terms only to less than $O(h)$ (the reason will become clear in the next step):

$$(\beta f + \gamma k_2)^2 = (\beta+\gamma)^2f^2 + O(h) \tag{46}$$

Using (45) and (46) to simplify (44) gives

$$\begin{aligned}
k_3 &= f + h(\beta+\gamma)f_t + hf_y[(\beta+\gamma)f + \alpha\gamma hF + \frac{1}{2}\gamma\alpha^2h^2G + O(h^3)] \\
&\quad + \frac{1}{2}[h^2f_{tt}(\beta+\gamma)^2 + 2h^2(\beta+\gamma)f_{ty}((\beta+\gamma)f + O(h)) \\
&\quad + h^2f_{yy}((\beta+\gamma)^2f^2 + O(h))] + O(h^3)
\end{aligned} \tag{47}$$

Collecting terms by each order of *h*,

$$\begin{aligned}
k_3 &= f + h(\beta+\gamma)(f_t + ff_y) + h^2\{\alpha\gamma Ff_y \\
&\quad + \frac{1}{2}[f_{tt}(\beta+\gamma)^2 + 2f_{ty}f(\beta+\gamma)^2 + f_{yy}(\beta+\gamma)^2f^2\} + O(h^3) \\
&= f + h(\beta+\gamma)F + h^2[\alpha\gamma Ff_y + \frac{1}{2}(\beta+\gamma)^2G] + O(h^3)
\end{aligned} \tag{48}$$

Therefore substituting from equations (37), (43) and (48) for k_1, k_2, k_3 into equation (33),

$$\begin{aligned} y_{n+1} &= y_n + hb_1 f + hb_2[f + \alpha hF + \frac{1}{2}\alpha^2 h^2 G + O(h^3)] \\ &\quad + hb_3[f + h(\beta+\gamma)F + h^2[\alpha\gamma F f_y + \frac{1}{2}(\beta+\gamma)^2 G] + O(h^3)] \\ &= y_n + h(b_1+b_2+b_3)f + h^2[b_2\alpha F + b_3(\beta+\gamma)F] \\ &\quad + h^3[\frac{1}{2}b_2\alpha^2 G + \alpha b_3\gamma F f_y + \frac{1}{2}b_3(\beta+\gamma)^2 G] + O(h^4) \end{aligned} \tag{49}$$

From equation (30)

$$y(t_{n+1}) = y_n + hf + \frac{1}{2}h^2 F + \frac{1}{6}h^3(G + f_y F) + O(h^4) \tag{50}$$

Thus

$$\begin{aligned} h \times \text{LTE} &= y_{n+1} - y(t_{n+1}) \\ &= y_n + h(b_1+b_2+b_3)f + h^2[b_2\alpha F + b_3(\beta+\gamma)F] \\ &\quad + h^3[\frac{1}{2}b_2\alpha^2 G + \alpha b_3\gamma F f_y + \frac{1}{2}b_3(\beta+\gamma)^2 G] \\ &\quad - y_n - hf - \frac{1}{2}h^2 F - \frac{1}{6}h^3(G + f_y F) + O(h^4) \\ &= hf(b_1+b_2+b_3-1) + h^2 F[b_2\alpha F + b_3(\beta+\gamma) - \frac{1}{2}] \\ &\quad + h^3[(\alpha b_3\gamma - \frac{1}{6})F f_y + \frac{1}{2}(b_2\alpha^2 + b_3(\beta+\gamma)^2 - \frac{1}{3})G)] + O(h^4) \end{aligned} \tag{51}$$

and therefore the local truncation error is

$$\begin{aligned} \text{LTE} &= f(b_1+b_2+b_3-1) + hF[b_2\alpha F + b_3(\beta+\gamma) - \frac{1}{2}] \\ &\quad + h^2[(\alpha b_3\gamma - \frac{1}{6})F f_y + \frac{1}{2}(b_2\alpha^2 + b_3(\beta+\gamma)^2 - \frac{1}{3})G)] + O(h^3) \end{aligned} \tag{52}$$

To ensure the method is at least $O(h)$ we must force the term that does not depend on h to be zero, and hence require that

$$b_1 + b_2 + b_3 = 1 \tag{53}$$

Similarly, to ensure that the method is at least $O(h^2)$ we require, in addition to (53), that the coefficient of h is also zero, so that

$$b_2\alpha F + b_3(\beta+\gamma) = \frac{1}{2} \tag{54}$$

Finally, to ensure that the method is at least $O(h^3)$ we require, in addition to (54) and (54), that the coefficient of h^2 be zero in equation (52):

$$(\alpha b_3\gamma - \frac{1}{6})Ff_y + \frac{1}{2}(b_2\alpha^2 + b_3(\beta+\gamma)^2 - \frac{1}{3})G) = 0 \tag{55}$$

Since equation (55) must hold for all functions *f*, it must hold for all functions F and G. Thus the coefficients of both F and G in (55) must be zero, giving us the conditions

$$\alpha b_3\gamma = \frac{1}{6} \tag{56}$$

$$b_2\alpha^2 + b_3(\beta+\gamma)^2 = \frac{1}{3} \tag{57}$$

The following theorem summarizes our results.

Theorem. *A 3-stage explicit Runge-Kutta method is given by*

$$y_{n+1} = y_n + h(b_1k_1 + b_2k_2 + b_3k_3)$$
$$k_1 = f(t_n,\ y_n) = f_n$$
$$k_2 = f(t+\alpha h,\ y+\alpha hf)$$
$$k_3 = f(t+h(\beta+\gamma),\quad y+h(\beta f+\gamma k_2))$$

and the method is $O(h^3)$ *for all functions f if and only if the following constraints are satisfied:*

(I) $b_1 + b_2 + b_3 = 1$

(II) $b_2\alpha F + b_3(\beta+\gamma) = \frac{1}{2}$

(III) $\alpha b_3\gamma = \frac{1}{6}$

(IV) $b_2\alpha^2 + b_3(\beta+\gamma)^2 = \frac{1}{3}$

Furthermore, there are no three-methods that are fourth order.

We did not prove the last statement; to do so would required expanding all of Taylor series one step further.

Since there are four constraints to be satisfied, but there are six arbitrary constants ($b_1, b_2, b_3, \alpha, \beta, \gamma$) there are an infinite number of possible third order three stage methods.

<u>Heun's Method</u> is given by the Butcher Array

$$\begin{array}{c|ccc} 0 & & & \\ 1/3 & 1/3 & & \\ 2/3 & 0 & 2/3 & \\ \hline & 1/4 & 0 & 3/4 \end{array}$$

It is straightforward to verify from $\alpha = 1/3,\ \beta = 0,\ \gamma = 2/3,\ b_1 = 1/4,\ b_2 = 0,\ b_3 = 3/4$, that Heun's method is third order. The iteration formulas can be written as:

$$k_1 = f(t_n,\ y_n) = f_n \tag{58}$$

$$k_2 = f(t_n + \frac{1}{3}h,\ y_n + \frac{1}{3}hf_n) \tag{59}$$

$$k_3 = f(t_n + \frac{2}{3}h,\ \ y + \frac{2}{3}hk_2) \tag{60}$$

$$y_{n+1} = y_n + h(\frac{1}{4}k_1 + \frac{3}{4}k_3) \tag{61}$$

While it is possible to combine these into a single formula, this is rarely done since no advantage is conferred by doing so.

<u>Kutta's Third-Order Method</u> is given by the Butcher Array

$$\begin{array}{c|ccc} 0 & & & \\ 1/2 & 1/2 & & \\ 1 & -1 & 2 & \\ \hline & 1/6 & 2/3 & 1/6 \end{array}$$

This third order method has the following iteration formulas:

$$k_1 = f_n$$

$$k_2 = f\left(t_n + \frac{h}{2}\ \ y_n + \frac{hk_1}{2}\right)$$

$$k_3 = f(t_n + h,\ y - h(k_1 - 2k_2))$$

$$y_{n+1} = y_n + h\left(\frac{1}{6}k_1 + \frac{2}{3}k_2 + \frac{1}{6}k_3\right)$$

Fourth Order Methods

Fourth order methods can be derived by expanding the Taylor series to $O(h^3)$ and repeating the process of setting the coefficients of the constant, h, h^2 and h^3 terms equal to zero. Also before there turn out to be an infinite number of possible methods. However, there is one method is particularly common; when the expression "Runge-Kutta Method" is used without qualification it almost always means this method, which we will refer to as the **Classical Runge Kutta Method**.

The Butcher array is given by

$$\begin{array}{c|cccc}
0 & & & & \\
1/2 & 1/2 & & & \\
1/2 & 0 & 1/2 & & \\
1 & 0 & 0 & 1 & \\
\hline
 & 1/6 & 1/3 & 1/3 & 1/6
\end{array}$$

The iteration formulas are:

$$k_1 = f_n$$

$$k_2 = f\left(t_n + \frac{1}{2}h,\ y_n + \frac{1}{2}hk_1\right)$$

$$k_3 = f\left(t_n + \frac{1}{2}h,\ y_n + \frac{1}{2}hk_2\right)$$

$$k_4 = f\left(t_n + h,\ y_n + k_3h\right)$$

$$y_{n+1} = y_n + h\left(\frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4\right)$$

For a general purpose solve, we would write the algorithm like this:

```
$Input(a, y_a, h, f(x,y))$
$x_0 = a; \quad y_0 = y_a;$
$For\ \ i = 1, 2, 3, \ldots$
    $k_1 = f(x_{i-1}, y_{i-1})$
    $k_2 = f(x_{i-1} + h/2,\ y_{i-1} + hk_1/2)$
    $k_3 = f(x_{i-1} + h/2,\ y_{i-1} + hk_2/2)$
    $k_4 = f(x_{i-1} + h,\ y_{i-1} + hk_3)$
    $y_i = y_{i-1} + h\left(\frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4\right)$
    $x_i = h + x_{i-1}$
$Return(y_1, y_2, y_3, \ldots)$
```

For any given differential equation, we could instead first find an iteration formula specific to that equation, as in the following example:

Example. Compute the iteration formula for the Classical Runga Kutta method for $y' = y$.

Since $f(x, y) = y$,

$$k_1 = f(x_n, y_n) = y_n$$

$$k_2 = f\left(x_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}hk_1\right) = y_n + \frac{1}{2}hy_n = y_n\left(1 + \frac{1}{2}h\right)$$

$$k_3 = f\left(x_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}hk_2\right) = y_n + \frac{1}{2}hy_n\left(1 + \frac{1}{2}h\right) = y_n\left(1 + \frac{1}{2}h + \frac{1}{4}h^2\right)$$

$$k_4 = f(x_n + h, y_n + hk_3) = y_n + hy_n\left(1 + \frac{1}{2}h + \frac{1}{4}h^2\right) = y_n\left(1 + h + \frac{1}{2}h^2 + \frac{1}{4}h^3\right)$$

and therefore

$$\begin{aligned} y_{n+1} &= y_n + h\left(\tfrac{1}{6}k_1 + \tfrac{1}{3}k_2 + \tfrac{1}{3}k_3 + \tfrac{1}{6}k_4\right) \\ &= y_n + h\left(\tfrac{1}{6}y_n + \tfrac{1}{3}y_n\left(1 + \frac{1}{2}h\right) + \tfrac{1}{3}y_n\left(1 + \frac{1}{2}h + \frac{1}{4}h^2\right) + \tfrac{1}{6}y_n\left(1 + h + \frac{1}{2}h^2 + \frac{1}{4}h^3\right)\right) \\ &= \left(1 + \tfrac{1}{6}h + \tfrac{1}{3}h\left(1 + \frac{1}{2}h\right) + \tfrac{1}{3}h\left(1 + \frac{1}{2}h + \frac{1}{4}h^2\right) + \tfrac{1}{6}h\left(1 + h + \frac{1}{2}h^2 + \frac{1}{4}h^3\right)\right)y_n \\ &= \left(1 + h + \frac{1}{2}h^2 + \frac{1}{6}h^3 + \frac{1}{24}h^4\right)y_n \end{aligned}$$

<u>Higher Order Methods</u>

Runga-Kutta methods to many higher orders have been tabulated in numerous places. One is apt to expect from our study of one, two and three stage methods that the number of stages required is equal to the order of the method; in general this is not true, and in fact in general for higher order methods one requires more than n stages for an nth order method.