Runge–Kutta Methods

General Principle

Calculate (t_{n+1}, y_{n+1}) from (t_n, y_n) using a collection of intermediate points $(t_{n,i}, y_{n,i})$ with

$$t_{n,i} = t_n + c_i h_n \quad , \quad 1 \le i \le q$$

{parameters}

for suitable choices $c_i \in [0, 1]$. The slope of any solution at these points, viz.,

$$p_{n,i} = f(t_{n,i}, y_{n,i})$$

provide approximations to the true slope $z'(t_n) = f(t_n, z(t_n))$.

We use a linear combination of the slopes $p_{n,i}$ as our approximation to the slope at t_n.

→ We think of this as a q-stage (and 1-step!) process.

Now the details...

Let z be an exact solution to (E). Then,

$$z(t_{n,i}) = z(t_n) + \int_{t_n}^{t_{n,i}} f(t, z(t))\, dt$$

$$= z(t_n) + h_n \int_0^{c_i} f(t_n + u h_n, z(t_n + u h_n))\, du$$

(change of variables $t = t_n + u h_n$)

$\Rightarrow dt = h_n\, du$

Let's use a "Riemann sum" to estimate the integral above... If we choose an arbitrary mesh for $[0, c_i]$, we'd need to estimates for $z(t_n + u h_n)$ at those grid points, so to keep things simple, we use the mesh

$$c_1 < \dots < c_j < \dots < c_i \qquad 1 \leq j < i.$$

$$\Rightarrow \int_0^{c_i} g(u)\, du \approx \sum_{1 \leq j < i} a_{ij}\, g(c_j)$$

This gives us a recipe for $y_{n,i}$ given (t_n, y_n) and $t_{n,j}$ $(1 \leq j \leq i)$:

$$\boxed{y_{n,i} = y_n + h_n \sum_{1 \leq j < i} a_{ij}\, p_{n,j}}$$

Finally

$$z(t_{n+1}) = z(t_n) + \int_{t_n}^{t_{n+1}} f(t, z(t))\, dt$$

$$\underset{\text{(change of var)}}{=} z(t_n) + h_n \int_0^1 f(t_n + u h_n, z(t_n + u h_n))\, du$$

We estimate the integral using "Riemann sum":

$$\int_0^1 g(u)\, du \approx \sum_{j=1}^{q} b_j\, g(c_j)$$

This gives us the recipe for y_{n+1}:

$$\boxed{y_{n+1} = y_n + h_n \sum_{j=1}^{q} b_j\, p_{n,j}}$$

Thus, Runge-Kutta method involves a mesh $(c_i)_{1 \le i \le q}$ of size q of the interval $[0, 1]$, and $(q-1)+1$ many quadrature schemes.

$$\boxed{\text{ALGORITHM (Runge-Kutta Methods)}} \qquad (0 \leq n \leq N-1)$$

$$1 \leq i \leq q \quad \begin{cases} t_{n,i} = t_n + c_i h_n \\ y_{n,i} = y_n + h_n \displaystyle\sum_{1 \leq j < i} a_{ij}\, p_{n,j} \\ \\ p_{n,i} = f(t_{n,i}, y_{n,i}) \end{cases}$$

$$t_{n+1} = t_n + h_n$$

$$y_{n+1} = y_n + h_n \sum_{i=1}^{q} b_i\, p_{n,i}$$

This scheme is represented by the "Butcher Tableau":

$$\begin{array}{c|cccc} c_1 & 0 & 0 & \cdots & 0 \\ c_2 & a_{21} & 0 & \cdots & 0 \\ \vdots & & & \ddots & \vdots \\ c_q & a_{q1} & a_{q2} & \cdots & 0 \\ \hline & b_1 & b_2 & \cdots & b_q \end{array}$$

Rmk. these could be taken to be non-zero if we desire the stages to be implicit.

Thus, for Runge-Kutta method above

$$\begin{cases} \Phi(t,y,h) = \sum_{i=1}^{q} b_i\, f(t+c_i h,\, y_i) \\ y_i = y + h \sum_{1\le j<i} a_{ij}\, f(t+c_j h,\, y_j) \end{cases}$$

For consistency, we would need:

$$\Phi(t,y,0) = f(t,y)$$

$$\Rightarrow \quad \left(\sum_{i=1}^{q} b_j\right) f(t,y) = f(t,y)$$

$$\Rightarrow \quad \boxed{\sum_{i=1}^{q} b_i = 1}$$

We will henceforth assume this.

Rmk. Another simplifying assumption usually made is that

$$c_i = \sum_{1\le j<i} a_{ij} \qquad \left(\text{In particular } c_1 = 0\right)$$

but this is related not to consistency, but to higher order conditions. See later...

EXAMPLES.

(1)
$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array} \rightsquigarrow \quad \begin{aligned} y_{n,1} &= y_n \\ P_{n,1} &= f(t_n, y_n) \end{aligned}$$

This is just the Euler's method.

$$\boxed{y_{n+1} = y_n + h_n f(t_n, y_n)}$$

(2) Consider the following 1-parameter family of 2-stage R-K methods:

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \alpha & \alpha & 0 \\ \hline & 1-\frac{1}{2\alpha} & \frac{1}{2\alpha} \end{array} \qquad \alpha \in (0,1].$$

$$y_{n+1} = y_n + h_n \left\{ \left(1 - \frac{1}{2\alpha}\right) f(t_n, y_n) + \frac{1}{2\alpha} \cdot f(t_n + \alpha h_n, y_n + \alpha h_n f(t_n, y_n)) \right\}$$

Some special cases.

$\alpha = \frac{1}{2}$: MIDPOINT METHOD

$\alpha = 1$: Recovers the trapezium method of integration. (Called Heun's method)

$\alpha = 2/3$: this is called <u>Ralston's method</u>.

(3) The classical Runge-Kutta method is given by the following Butcher tableau:

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

$(q = 4)$

In this, the methods of integration used are:

$$\int_0^{1/2} g(u)\,du \approx \frac{1}{2} g(0)$$ (Left rectangle)

$$\int_0^{1/2} g(u)\,du \approx \frac{1}{2} g(1/2)$$ (right rectangle)

$$\int_0^1 g(u)\,du \approx g\left(\tfrac{1}{2}\right)$$ (mid point estimate)

finally, $$\int_0^1 g(t)\,dt \approx \frac{1}{6} g(0) + \frac{2}{6} g\left(\tfrac{1}{2}\right) + \frac{2}{6} g\left(\tfrac{1}{2}\right) + \frac{1}{6} g(1)$$

(Simpson approx!)

We will admit the following theorem without proof:

<u>THEOREM.</u> Suppose that $f(t,y)$ is Lipschitz in y w/ Lipschitz constant k. Let $\Phi(t,y,h)$ be Runge-Kutta method whose Butcher tableau is

$$
\begin{array}{c|ccccc}
c_1 & 0 & \cdots & \cdots & & 0 \\
c_2 & a_{21} & 0 & & & \vdots \\
\vdots & a_{31} & a_{32} & 0 & & \vdots \\
 & \vdots & \vdots & \ddots & 0 & \vdots \\
c_q & a_{q1} & a_{q2} & \cdots & a_{q\,q-1} & 0 \\
\hline
 & b_1 & \cdots & & b_{q-1} & b_q
\end{array}
$$

Then

(*) $\Phi(t,y,h)$ is Lipschitz in y w/ Lipschitz constant

$$\Lambda = k \cdot \left(\sum_{j=1}^{q} |b_j| \right) \left(\sum_{j=1}^{q-1} (\alpha k h_{max})^j \right)$$

where $\alpha = \max_i \left(\sum_{1 \le j < i} |a_{i,j}| \right)$ $\blacksquare$

COROLLARY. If f is Lipschitz in y, then Runge-Kutta methods are stable. ∎

§ Order of RK methods

- Order ≥ 1 $\Longleftrightarrow$ Consistent
  $\Longleftrightarrow$ $\sum_j b_j = 1$ — (*)

- Let's explore if a 2-stage ($q=2$) method can give us an order 2 method.

By an earlier exercise, we need:

$$\frac{\partial \Phi}{\partial h}(t, y, 0) = \frac{1}{2} f^{[1]}(t, y)$$

(in addition to (*)).

The rest of this is an unenlightening calculation, but we do it:

$$\Phi(t, y, h) = b_1 f(t + c_1 h, y_1) + b_2 f(t + c_2 h, y_2)$$

w/ $\quad y_1 = y$

$$y_2 = y + h\, a_{21}\, f(t+c_1 h, y)$$

$$\frac{\partial \Phi}{\partial h} = b_1\left[c_1 f_t(t+c_1h, y_1) + f_y(t+c_1h, y_1)\cancel{\frac{\partial y_1}{\partial h}}\right]$$

$y_1 = y$

$\frac{\partial y}{\partial h} = 0$

$$+\ b_2\left[c_2 f_t(t+c_2h, y_2) + f_y(t+c_2h, y_2)\frac{\partial y_2}{\partial h}\right]$$

$$a_{21}\left[f(t+c_1h, y) + h\,\frac{\partial f(t+c_1h, y)}{\partial h}\right]$$

$$\frac{\partial \Phi}{\partial h}(t,y,0) = (b_1c_1 + b_2c_2)\, f_t(t,y) + b_2 a_{21}\, f_y(t,y)\, f(t,y)$$

In order to neatly get $f^{[1]}(t,y)$

$$= (f_t + f_y f)$$

let's set $\begin{cases} a_{21} = c_2 \\ c_1 = 0 \end{cases}$. (compare w/ Rmk. just before the example.)

Then, we need: $b_2 c_2 = 1/2$. Compare to Example 2.

$\blacksquare$

§ Multi-step Methods.

In contrast to 1-step methods, Multi-step methods allow for extrapolation techniques.

Here's the key idea behind many multi-step methods:

Start as usual with FTC:

$$z(t_{n+1}) = z(t_n) + \int_{t_n}^{t_{n+1}} f(t, z(t))\, dt$$

We do not know $z(t)$ on $[t_n, t_{n+1}]$. But in a multi-step method, we know z on the chosen mesh in $[t_{n+1-s}, t_n]$ (and hence $f(t, z(t))$)

The idea is simply to fit a polynomial on $(t_k, f(t_k, y_k))$ for $n-(s-1) \leq k \leq n$ of degree $s-1$.

We illustrate this for 3-step ($s = 3$) method.

(This special method is called Adams-Bashforth Method.)

Let us review fitting quadratic polynomials to three points in $\mathbb{R}^2$:

$$(t_1, f_1),\quad (t_2, f_2),\quad (t_3, f_3).$$

Idea 1. Want $p(x) = \alpha_2 x^2 + \alpha_1 x + \alpha_0$ such that

$$p(t_i) = f_i$$

This gives the system:

$$\begin{pmatrix} t_1^2 & t_1 & 1 \\ t_2^2 & t_2 & 1 \\ t_3^2 & t_3 & 1 \end{pmatrix} \begin{pmatrix} \alpha_2 \\ \alpha_1 \\ \alpha_0 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix}$$

$\rightsquigarrow$ Solve by inverting the (transpose) of the Vandermonde matrix!

Idea 2. (Change the "Ansatz"!)

Want $p(x) = \beta_3 (x - t_1)(x - t_2) + \beta_2 (x - t_1)(x - t_3)$
$\qquad + \beta_1 (x - t_2)(x - t_3)$

such that $p(t_i) = f_i$

This immediately gives:

$$\forall i \quad \beta_i = \frac{f_i}{(t_i - t_j)(t_i - t_k)}. \qquad \{i, j, k\} = \{1, 2, 3\}$$

Applying this with $t_1 = t_{n-2}$, $f_1 = f(t_{n-2}, y_{n-2})$
$t_2 = t_{n-1}$, $f_2 = f(t_{n-1}, y_{n-1})$
$t_3 = t_n$, $f_3 = f(t_n, t_n)$.

We propose:

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} p(x)\, dx$$

Let me carry out the integrals under the assumption that $h_n = t_{n+1} - t_n$ is independent of n and equals h say.

$$\int_{t_n}^{t_{n+1}} (x - t_{n-2})(x - t_{n-1})\, dx = h \int_0^1 (t_n + hu - t_{n-2})(t_n + hu - t_{n-1})\, du$$

$u = \dfrac{x - t_n}{h}$

$\Rightarrow dx = h \cdot du$

$$= h^3 \int_0^1 \underbrace{(u+2)(u+1)}_{u^2+3u+2}\, du$$

$$= h^3\left(\frac{1}{3} + \frac{3}{2} + 2\right) = h^3\left(\frac{23}{12}\right)$$

The same change-of-variable trick can be employed to get the other integrals [EXERCISE!]

In summary, this gives:

$$y_{n+1} = y_n + \frac{h}{12}\left(5f_{n-2} - 16 f_{n-1} + 23 f_n\right)$$

Remarks

(1) Note that this method is exact if $f(t, z(t))$ is a polynomial of degree 2.

This suggests (though doesn't prove!) that this method has order 2.

(2) Note that $\frac{y_{n+1} - y_n}{h}$ is linear in f_{n+1-s} through f_n, in general. Thus this method offers good numerical stability.

END

Exercises.

1. Derive the 2-step Adams-Bashforth method.

2. Explicate the RK method whose Butcher Tableau is

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

3. Suppose that $\Phi(t, y, h) : [0,2] \times \mathbb{R} \times [0,1] \longrightarrow \mathbb{R}$ is Lipschitz in y w/ Lipschitz constant $\Lambda > 0$. Consider the 1-step method:

$$\begin{cases} y_0 = y(0) \\ y_{n+1} = y_n + h_n \Phi(t_n, y_n, h_n) \\ t_{n+1} = t_n + h_n \end{cases}$$

If this method has order 2, how large can h_{max} be while still ensuring that y_n agrees with true solution upto 1 decimal place?