Lenny Wu

Rmk. Modified Euler's uses "average slope"

Ex. Let $f(t,y)$ be continuous on $[a,b]\times\mathbb{R}$ and Lipschitz continuous in y.
Show that Modified Euler's is stable and convergent.

Pf. Want to satisfy conditions in Thm. 5.20.
WTS: $\phi(t,w,h) = \frac{1}{2}f(t,w) + \frac{1}{2}f(t+h, w+f(t,w))$
$\forall\ (t,w,h)\in\Omega = [a,b]\times\mathbb{R}\times[0,h_0]$
is continuous and Lipschitz continuous in w (and consistent).

1. continuity: f continuous $\Rightarrow$ ϕ continuous
2. Lipschitz continuity: since f is Lipschitz continuous in y on $[a,b]\times\mathbb{R}$, we have:

$$
\begin{aligned}
|\phi(t,w,h) - \phi(t,\tilde{w},h)| &= \tfrac{1}{2}\left|f(t,w) - f(t,\tilde{w}) + f(t+h, w+f(t,w)) - f(t+h, \tilde{w}+f(t,\tilde{w}))\right| \\
&\le \tfrac{L}{2}|w-\tilde{w}| + \tfrac{L}{2}|w-\tilde{w} + hf(t,w) - hf(t,\tilde{w})| \\
&\le \tfrac{L}{2}|w-\tilde{w}| + \tfrac{L^2}{2}h|w-\tilde{w}| \\
&\le \left(\tfrac{L}{2}+\tfrac{L^2}{2}h\right)|w-\tilde{w}|
\end{aligned}
$$

So ϕ is Lipschitz continuous in w with Lipschitz constant $\tilde{L} = \frac{L + L^2 h_0}{2}$

Thus, by Thm. 5.20, Modified Euler's is stable.

To show convergence, note consistency $\Leftrightarrow$ convergence (Thm. 5.20).

Let $h=0$. We have:

$$\phi(t,w,0) = \tfrac{1}{2} f(t,w) + \tfrac{1}{2} f(t+h, w+0) = f(t,w)$$

$\Rightarrow$ method is consistent

$\Rightarrow$ method is convergent by Thm. 5.20

Runge-Kutta of Order 4 (RK-4)

$$w_0 = \alpha$$

$$k_1 = hf(t_i, w_i)$$

$$k_2 = hf(t_i + \tfrac{h}{2}, w_i + \tfrac{1}{2}k_1)$$

$$k_3 = hf(t_i + \tfrac{h}{2}, w_i + \tfrac{1}{2}k_2)$$

$$k_4 = hf(t_i + h, w_i + k_3)$$

$$w_{i+1} = w_i + \tfrac{1}{6}\underbrace{(k_1 + 2k_2 + 2k_3 + k_4)}_{\phi}, \quad i = 0, 1, \dots N-1$$

Rmk. 1. This is the most common RK scheme

2. Method has local truncation error $O(h^4)$ (assuming 5 continuous deriv.s)

3. RK4: 4 evaluations of f per step
Euler: 1 evaluation of f per step
$\Rightarrow$ if RK4 is to be superior, it should give more accurate answers than Euler's method with 1/4 the step size.

General R-K Methods

Def. Runge-Kutta methods are defined by

$$\begin{cases} w_0 = \alpha \\ w_{i+1} = w_i + \sum_{j=1}^{s} b_j k_j \end{cases}$$

where $k_j = hf(t_i + c_j h, w_i + \sum_{l=1}^{s} a_{jl} k_l)$

- h: step size
- s: # of substeps (stages) taken from t_i
- b_j: weight of the j^{th} substep to update w_i
- c_j: increment factor of j^{th} substep
- a_{jl}: weight that the l^{th} substep contributes to the j^{th} substep

Note: The method is $\begin{cases} \text{explicit} & \text{if } a_{jl} = 0 \ \forall\, l \geq j \\ \text{implicit} & \text{otherwise} \end{cases}$

Up until now, we've dealt with explicit methods only, i.e. k_j only depends on previous k's:

$$k_j \leftarrow (k_1, k_2, \dots k_{j-1})$$

Ex. RK4 is explicit!

Facts: 1. $\sum_{j=1}^{s} b_j = 1 \Rightarrow$ RK method consistent

2. $\sum_{j=1}^{s} a_{jl} = c_j \Rightarrow$ RK method stable

A compact way to describe RK methods is using a Butcher Tableau.

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & & \\ \vdots & \vdots & \vdots & \ddots & \\ c_3 & a_{s1} & a_{32} & & a_{ss} \\ \hline & b_1 & b_2 & & b_3 \end{array}$$

Rmk. 1. $c_1 = 0$ used in general, and $a_{jj} = 0$ holds for explicit methods.

2. Explicit RK method s substeps may not have order > s, and thereby the matrix A is lower-triangular, with all diagonal entries being 0.

Ex. Find the Butcher Tableau for the modified Euler's method.

Sol. Recall:

$$w_{i+1} = w_i + \underset{b_1}{\frac{1}{2}} \underbrace{hf(t_i, w_i)}_{k_1} + \underset{b_2}{\frac{1}{2}} \underbrace{hf(t_i+h, w_i+f(t_i, w_i))}_{k_2}$$

and
$$\begin{cases} w_0 = \alpha \\ w_{i+1} = w_i + \sum_{j=1}^{s} b_j k_j, \\ \qquad k_j = hf(t_i + c_j h, w_j + \sum_{j=1}^{s} a_{j\ell} k_j) \end{cases}$$

We get:

$s=2$

$b_1 = b_2 = \frac{1}{2}$

$c_1 = 0, \ c_2 = 1$

$a_{11} = 0, \ a_{12} = 0, \ a_{21} = 1, \ a_{22} = 0$

The Butcher Tableau is:

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

Ex. Find the Butcher Tableau for RK-4.

Sol.

$$w_0 = \alpha$$

$$k_1 = hf(t_i, w_i)$$

$$k_2 = hf(t_i + \tfrac{h}{2}, w_i + \tfrac{1}{2}k_1)$$

$$k_3 = hf(t_i + \tfrac{h}{2}, w_i + \tfrac{1}{2}k_2)$$

$$k_4 = hf(t_i + h, w_i + k_3)$$

$$w_{i+1} = w_i + \tfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

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

Rmk.

1. Main computation in RK methods is in function evaluations.
2. The relationship between function evaluations and local error of order p is:

# func evals	2	3	4	$5 \le n \le 7$	$8 \le n \le 9$	$n \ge 10$
order of accuracy	$O(h^2)$	$O(h^3)$	$O(h^4)$	$O(h^{n-1})$	$O(h^{n-2})$	$O(h^{n-3})$

3. RK-4 is among most popular methods since:

- simple to code
- reasonable accuracy (4^{th} order)
- no 5-stage 5^{th} order method

Beyond 4^{th} order, the RK methods are relatively more expensive to compute.

5.5 Error Control and Runge-Kutta-Fehlberg Method

Error Control:

IVP $\begin{cases} y' = f(t,y) & a \le t \le b \\ y(a) = \alpha \end{cases}$

Goal: Given $\varepsilon > 0$, pick minimal number of time points (or "optional" step sizes st error does not exceed ε at every step.

Idea: Consider n^{th} order and $(n+1)^{th}$ order method

① $y_{i+1} = y_i + h\phi(t_i, y_i, h) + O(h^{n+1})$

(error: $\tau_{i+1}(h) = \dfrac{y_{i+1}-y_i}{h} - \phi = O(h^n)$)

② $y_{i+1} = y_i + h\hat{\phi}(t_i, w_i, h) + O(h^{n+2})$

(error: $\hat{\tau}_{i+1}(h) = \dfrac{y_{i+1}-y_i}{h} - \hat{\phi} = O(h^{n+1})$)

The corresponding difference eqns. are:

$$\begin{cases} w_0 = \alpha \\ w_{i+1} = w_i + h\phi(t_i, w_i, h) \end{cases} \quad \text{and} \quad \begin{cases} \hat{w}_0 = \alpha \\ \hat{w}_{i+1} = \hat{w}_i + h\hat{\phi}(t_i, \hat{w}_i, h) \end{cases}$$

Assume $w_i = y_i = \hat{w}_i$ and choose a fixed h to generate approximations w_{i+1} and $\hat{w}_{i+1}$, to approximate y_{i+1}.

Then,
$$\begin{aligned} \tau_{i+1}(h) &= \frac{y_{i+1}+y_i}{h} - \phi(t_i, y_i, h) \\ &= \frac{y_{i+1}+y_i}{h} - \phi(t_i, w_i, h) \\ &= \frac{y_{i+1} - [w_i + h\phi(t_i, w_i, h)]}{h} \\ &= \frac{1}{h}(y_{i+1} - w_{i+1}) \end{aligned}$$

Similarly,
$$\begin{aligned} \hat{\tau}_{i+1}(h) &= \frac{1}{h}(y_{i+1} - \hat{w}_{i+1}) \\ \underbrace{\tau_{i+1}(h)}_{O(h^n)} &= \frac{1}{h}(y_{i+1} - w_{i+1}) \\ &= \frac{1}{h}[(y_{i+1} - \hat{w}_{i+1}) + (\hat{w}_{i+1} - w_{i+1})] \\ &= \underbrace{\hat{\tau}_{i+1}(h)}_{O(h^{n+1})} + \frac{1}{h}(\hat{w}_{i+1} - w_{i+1}) \end{aligned}$$

Recall: $\hat{\tau}_{i+1}(h) = O(h^{n+1})$ but $\tau_{i+1}(h) = O(h^n)$

$\Rightarrow$ Big portion of error comes from $\frac{1}{h}(\hat{w}_{i+1} - w_{i+1})$ (when h small)

$\Rightarrow \tau_{i+1}(h) \approx \frac{1}{h}(\hat{w}_{i+1} - w_{i+1})$ (for small enough h)

Now, want to adjust step size to keep τ_{i+1} within specific bound.

$\tau_{i+1}(h) = O(h^n)$

$\Rightarrow \tau_{i+1}(h) \approx k \cdot h^n$ for some $k > 0$

Let the step size of interest be $q \cdot h$.

$$\Rightarrow \tau_{i+1}(qh) \approx k(qh)^n = q^n(kh)^n$$
$$\approx q^n \tau_{i+1}(h)$$

Want: $\tau_{i+1}(qh) < \varepsilon$

$\rightarrow q^n \tau_{i+1}(h) \leq \varepsilon$

$\rightarrow q^n/h \, |\hat{w}_{i+1} - w_{i+1}| \leq \varepsilon$

$\rightarrow q \leq \left(\frac{\varepsilon \cdot h}{|\hat{w}_{i+1} - w_{i+1}|}\right)^{1/n}$ (*)

Rmk. Use 2 methods of different orders to create new method (w/ particularly chosen step size).

Runge-Kutta-Fehlberg Method (RKF)

A popular method that uses (*)

Idea: Use RK method of order 5:

$$\hat{w}_{i+1} = w_i + \frac{16}{135}k_1 + \frac{6656}{12855}k_2 + \frac{28561}{56430}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6$$

to estimate local error in the following RK method of order 4:

$$w_{i+1} = w_i + \frac{25}{216}k_1 + \frac{1408}{2565}k_3 + \frac{2197}{4104}k_4 - \frac{1}{5}k_5, \quad \text{where}$$

$$k_1 = hf(t_i, w_i)$$

$$k_2 = hf\left(t_i + \frac{h}{4},\ w_i + \frac{1}{4}k_1\right)$$

$$k_3 = hf\left(t_i + \frac{3}{8}h,\ w_i + \frac{3}{32}k_1 + \frac{9}{32}k_2\right)$$

$$k_4 = hf\left(t_i + \frac{12}{13}h,\ w_i + \frac{1932}{2197}k_1 - \frac{7200}{2197}k_2 + \frac{7296}{2197}k_3\right)$$

$$k_5 = hf\left(t_i + h,\ w_i + \frac{439}{216}k_1 - 8k_2 + \frac{3680}{513}k_3 - \frac{845}{4104}k_4\right)$$

$$k_6 = hf\left(t_i + \frac{h}{2},\ w_i + \frac{8}{27}k_1 - 2k_2 + \frac{3544}{2565}k_3 + \frac{1859}{4104}k_4 - \frac{11}{40}k_5\right)$$

Rmk. 1. This particular choice of RK4 and RK5 requires only 6 func evals!

(using other arbitrary RK methods: $\Rightarrow 4+5=9$ evals)

2. $q = \left(\frac{\varepsilon h}{|\hat{w}_{i+1} - w_{i+1}|}\right)^{1/n}$ could have a different value at each time step

3. at each time step:
   - if $q<1$, then $\underbrace{\varepsilon < \tau_{i+1}(h)}_{\text{what we want!}}$

   reject h, and recompute w_{i+1} with $q \cdot h$
   - if $q>1$, then accept current w_{i+1}, but change the next step size to $q \cdot h$ for the $(i+1)^{th}$ step

   (skip size was not large enough to begin with)

4. In practice, to avoid spending too much time taking small steps, a common choice for RKF-4 is
$$q = \left(\frac{\varepsilon h}{2|\hat{w}_{i+1} - w_{i+1}|}\right)^{1/4} = 0.84\left(\frac{\varepsilon h}{|\hat{w}_{i+1} - w_{i+1}|}\right)^{1/4}$$

5. Can similarly estimate errors using one method w/ 2 different step sizes.