11-27-17, Mon

Section 8.3: Runge-Kutta methods

Taylor series methods can give highly accurate results but they are a nuisance because they require us to take derivatives of $f(x,t)$. Each time problem (ODE) changes we have to recalculate the derivatives.

It is preferable to use a scheme that does not require the upfront work.

IVP:

$$\begin{cases} x' = f(x,t) \\ x(t_0) = x_0 \end{cases}$$

Starting from the Taylor series for

$$x(t+h) = x(t) + hx'(t) + \frac{h^2}{2!}x''(t) + \frac{h^3}{3!}x'''(t) + \dots$$

So,

$$x'(t) = f(x,t)$$

$$x''(t) = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial x}\frac{dx}{dt} = f_t + ff_x.$$

chain rule

Substitute these expressions into TS gives

$$x(t+h) = x + hf + \tfrac{1}{2}h^2(f_t + ff_x) + O(h^3) =$$

(1) $$= x + \tfrac{1}{2}hf + \tfrac{1}{2}h(f + hf_t + hff_x) + O(h^3).$$

But the multidimensional TS for $f(t+h, \underbrace{x+hf}_{\substack{\text{Euler}\\\text{approximation}\\\text{to soln. at new time}}})$ yields

$$f(t+h, x+hf) = f(t,x) + hf_t + hff_x + \mathcal{O}(h^2)$$

So we can rewrite ① as

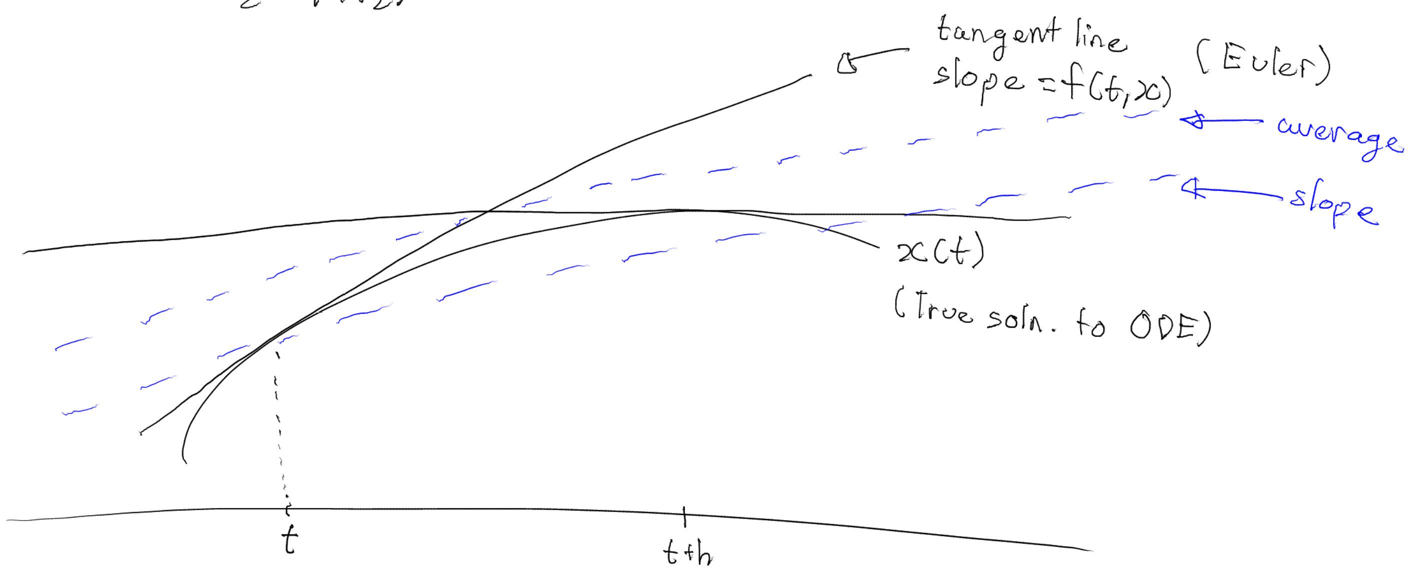
$$x(t+h) = x + \tfrac{1}{2}hf + \tfrac{1}{2}hf(t+h, x+hf) + \mathcal{O}(h^3).$$

We can rewrite this as

$$\left.\begin{array}{l} x(t+h) = x(t) + \tfrac{1}{2}(f_1 + f_2) \\ \text{where } f_1 = hf(t,x) \text{ and } f_2 = hf(t+h, x+f_1). \end{array}\right\}$$

This is a 2nd order Runge-Kutta method.

The term $\tfrac{1}{2}(f_1+f_2)$:



slope $= f(t+h, \underbrace{x(t) + hf(t,x)}_{x+\Delta x})$

$\downarrow$

$\frac{\Delta x}{\Delta t}$

This is a generalization of the trapezoid rule.

Idea at this point is there are choices for 2nd order Runge-Kutta schemes.
General scheme tries to reduce error as much as possible.

$$x(t+h) = x + \omega_1 \underset{\substack{\downarrow \\ \text{old slope}}}{hf} + \omega_2 h \underbrace{f(t+\alpha h, x+\beta hf)}_{\substack{\text{slope at unknown soln.} \\ \text{new point}}} + \mathcal{O}(h^3).$$

Using a TS for $f(t+\alpha h, x+\beta hf)$, we have

(2) $$x(t+h) = x(t) + \omega_1 hf + \omega_2 h\left[f + \alpha h f_t + \beta h f f_x\right] + \mathcal{O}(h^3).$$

Subtracting equations (1) and (2) gives

(2) − (1) $\Leftrightarrow$ (true) − (approximate)

$$\text{Truncation error} = (1-\omega_1-\omega_2)hf + \left(\tfrac{1}{2} - \omega_2\alpha\right)h^2 f_t + \left(\tfrac{1}{2} - \omega_2\beta\right)h^2 f f_x + \mathcal{O}(h^3).$$

So we would like

$$\begin{cases} 1-\omega_1-\omega_2 = 0 \\ \frac{1}{2} - \omega_2\alpha = 0 \\ \frac{1}{2} - \omega_2\beta = 0 \end{cases} \Rightarrow \begin{cases} \omega_1 + \omega_2 = 1 \\ \omega_2\alpha = \frac{1}{2} \\ \omega_2\beta = \frac{1}{2} \end{cases} \Rightarrow$$

$\Rightarrow$ truncation error is $\mathcal{O}(h^3)$.

Note that the system has many solutions.

(1) choice is $\omega_1 = \omega_2 = \frac{1}{2}$ and $\alpha = \beta = 1$.
This is Huen's method.

(2) choice is $\omega_1 = 0$, $\omega_2 = 2$, and $\alpha = \beta = \frac{1}{2}$.
This is Modified Euler.

So choice 2 gives

$$\begin{cases} x(t+h) = x(t) + f_2 \\ f_1 = hf(t, x) \\ f_2 = hf(t + \frac{1}{2}h, x + \frac{1}{2}f_1) \end{cases}$$

Higher order RK methods (e.g. fourth-order) can be derived similarly. The idea is to reproduce terms in TS up to and including h^4 terms. Error is $\mathcal{O}(h^5)$.

<u>Example:</u> $x(t+h) = x(t) + \frac{1}{6}(f_1 + 2f_2 + 2f_3 + f_4)$
where

$$f_1 = hf(t, x),$$

$$f_2 = hf(t + \tfrac{1}{2}h, x + \tfrac{1}{2}f_1),$$

$$f_3 = hf(t + \tfrac{1}{2}h, x + \tfrac{1}{2}f_2), \text{ and}$$

$$f_4 = hf(t + h, x + f_3).$$

This is a generalization of Simpson's rule.
(To see this, consider $f(t, x) = f(t)$, i.e. f depends only on x).

Example: Consider solving

$$\begin{cases} y'(x) = \frac{1}{1+x^2} - 2y^2 \\ y(0) \end{cases}$$

True solution: $y(x) = \frac{x}{1+x^2}$.

Using 4th order Runge-Kutta, we have $(h = 0.25, 2h = 0.5)$

x	$y_h(x)$	$y^{true} - y_h$	$y^{true} - y_{2h}$	ratio
2.0	0.39995699	4.35e-5	1.0e-3	24
4.0	0.23529159	2.5e-6	7.0e-5	28
6.0	0.16216179	3.7e-7	1.2e-5	32
8.0	0.12307683	9.2e-8	3.4e-6	36
10.0	0.9900987	3.1e-8	1.3e-6	41

In fact, method has a local truncation error of $O(h^5)$ and global error of $O(h^4)$. So theoretically, ratio is 16. We do better, but if you decrease h, ratio is closer to 16.

Theorem: If the Runge-Kutta method has a truncation error that is $O(h^{m+1})$ then global rate of convergence is $O(h^m)$.

Note: These schemes are highly accurate but the price you pay is a lot of function evaluations.

Section 8.4: Multistep methods

Recall one of the derivations of Euler's method.
Integrate

$$y'(t) = f(t, y(t))$$

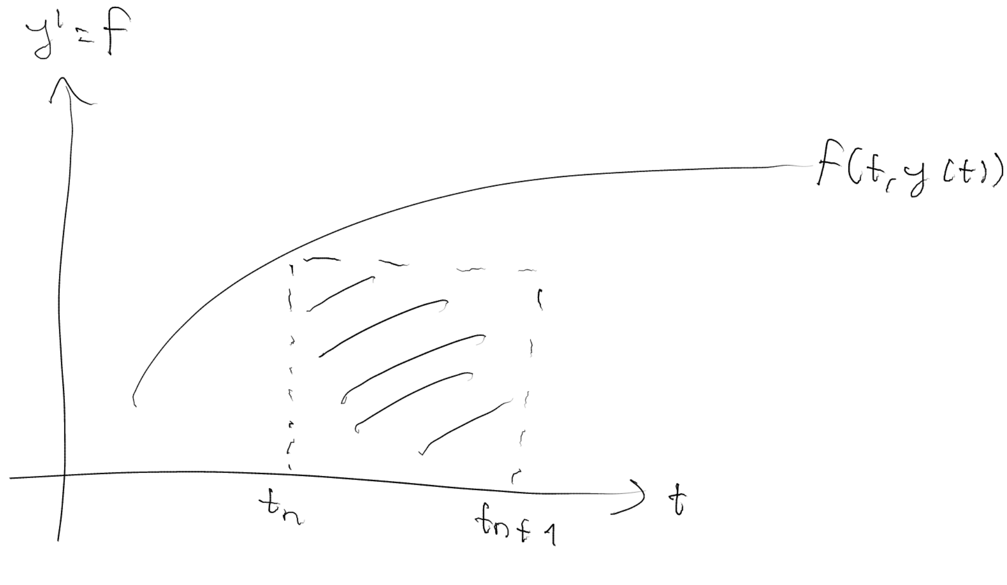
from t_n to t_{n+1}.

$$\int_{t_n}^{t_{n+1}} y'(t)\,dt = \int_{t_n}^{t_{n+1}} f(t, y(t))\,dt .$$

FTC gives

$$LHS = y(t_{n+1}) - y(t_n).$$

RHS can be approximated by different numerical integration schemes. Euler used left-hand rectangle rule.



$$y(t_{n+1}) - y(t_n) = \underbrace{(t_{n+1} - t_n)}_{\text{width}} \underbrace{f(t_n, y(t_n))}_{\text{heigh of rectangle}}$$

$$y(t_{n+1}) = y(t_n) + h f(t_n, y(t_n)).$$

This idea can be generalized to give other methods.

<u>Example</u>: (Midpoint method)

$$\int_{t_n}^{t_{n+1}} y'(t)\,dt = \int_{t_n}^{t_{n+1}} f(t, y(t))\,dt$$

If we apply the midpoint method directly to RHS, we need to evaluate f at a non-node point ($t_{n+\frac{1}{2}}$).

$$\int_{t_{n-1}}^{t_{n+1}} y'(t)\,dt = \int_{t_{n-1}}^{t_{n+1}} f(t, y)\,dt$$

Multistep method

Node point

$$y(t_{n+1}) - y(t_{n-1}) = (t_{n+1} - t_{n-1})\, f(t_n, y_n)$$

$$y(t_{n+1}) = y(t_{n-1}) + 2h f(t_n, y_n).$$

<u>Question</u>: What do you notice?

<u>Answer</u>: It determines soln. of ODE at new point t_{n+1} from 2 previous node points.