## 3. Predictor - corrector method:

to avoid solving equations in implicit schemes:
predict: use an explicit method to estimate $\overline{Y}_{n+1}$
correct: in implicit scheme use $\overline{Y}_{n+1}$ in RHS, instead of $Y_{n+1}$.

## Taylor series method:

Assuming $y' = f(t,y)$ has smooth solutions, expand $y(t)$ about $t = t_n$:

$$y(t_n + h) = y(t_n) + y'(t_n)\cdot h + \frac{y''(t_n)}{2!}h^2 + \frac{y'''(t_n)}{3!}h^3 + \dots + \underbrace{\frac{y^{(m)}(t_n)}{m!}h^m}_{m\text{-th order}} + O(h^{m+1})$$

Now $y' = f(t,y) \Rightarrow y'(t_n) = f(t_n, Y_n)$

$$y'' = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial y}\cdot\frac{dy}{dt} \Rightarrow y''(t_n) = \frac{\partial f}{\partial t}(t_n, Y_n) + \frac{\partial f}{\partial y}(t_n, Y_n)\cdot f(t_n, Y_n)$$

$y'''(t) = \dots$ gets very messy...

For simple $f(t,y)$, can do differentiations directly, at least for low order terms, very messy...

e.g. $y' = y^2 - 3t$, $y(0) = 1$: 3rd order Taylor:

$$Y_{n+1} \approx y(t_{n+1}) \approx Y_n + y'(t_n)\cdot h + y''(t_n)\frac{h^2}{2!} + y'''(t_n)\frac{h^3}{3!}$$

with $y'(t_n) = f(t_n, Y_n) = Y_n^2 - 3t_n$

$$y''(t_n) = 2yy' - 3 = 2Y_n\cdot[Y_n^2 - 3t_n]$$

$$y'''(t_n) = 2(y'^2 + yy'') = 2[Y_n^2 - 3t_n]^2 + 2Y_n\cdot[\quad]$$

# Runge-Kutta methods

Idea: replace higher derivates in Taylor by f-evaluations.

e.g. 2nd order RK: start with 2nd order Taylor expansion of $y(t)$ about $t_n$:

$$y(t_{n+1}) = y(t_n) + h\cdot y'(t_n) + \frac{h^2}{2} y''(t_n) + O(h^3)$$

now $y' = f(t,y) \Rightarrow y'' = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial y}\cdot y' = f_t + f f_y$

$$\Rightarrow y(t_{n+1}) = y(t_n) + \underbrace{hf + \frac{h^2}{2}\left[f_t + f f_y\right]} + O(h^3)$$

Runge, in 1904, noticed that these terms resemble terms in Taylor expansion of $f(t,y)$: expand $f(t_n+\alpha, y_n+\beta)$ about $(t_n, y_n)$:

$$h\cdot f(t_n+\alpha, y_n+\beta) = h\cdot f + \underbrace{h\cdot\alpha\cdot\frac{\partial f}{\partial t} + h\cdot\beta\cdot\frac{\partial f}{\partial y}} + h\cdot O(h^2)$$

matching terms $\Rightarrow$ $h\cdot\alpha = \frac{h^2}{2}$, $h\cdot\beta = \frac{h^2}{2}\cdot f$ $\Rightarrow \alpha = \frac{h}{2}$, $\beta = \frac{h}{2} f$ up to $O(h^3)$

$\Rightarrow$ 2nd order RK: $Y_{n+1} = Y_n + h\cdot f\left(t_n + \frac{h}{2}, Y_n + \frac{h}{2} f(t_n, Y_n)\right)$ explicit single step

better view: $K_1 = f(t_n, Y_n)$

$Y_{n+1} = Y_n + h\cdot f\left(t_n + \frac{h}{2}, Y_n + \frac{h}{2}\cdot K_1\right)$

2 stages f-evaluations

Viewed as Predictor-Corrector: $Y^{pred} = Y_n + h\cdot f(t_n, Y_n)$ (forward Euler)

$$Y_{n+1} = Y_n + h\cdot f\left(t_n + \frac{h}{2}, \frac{Y_n + Y^{pred}}{2}\right)$$

For higher orders, many choices for $\alpha$, $\beta$. exist.

Expansions get very long and messy... up to 10th order... in 1990s

RK methods are characterized by order $p$ of accuracy and number of stages $s$ (# of f-evaluations = cost)

RK1 = RK($p$ = 1, $s$ = 1) $\equiv$ Forward Euler

RK2 = RK(2,2) (above)

best known: classical RK4 = RK(4,4)

For orders $p > 4$ need $s > p$, so 4th order is "optimal"

Explicit and implicit versions exist

Classical RK4: $Y_{n+1} = Y_n + \frac{h}{6}\left[K_1 + 2K_2 + 2K_3 + K_4\right]$

where
$$\begin{aligned} K_1 &= f(t_n, Y_n) \\ K_2 &= f(t_n + \tfrac{h}{2}, \; Y_n + \tfrac{h}{2}K_1) \\ K_3 &= f(t_n + \tfrac{h}{2}, \; Y_n + \tfrac{h}{2}K_2) \\ K_4 &= f(t_n + h, \; Y_n + hK_3) \end{aligned}$$

$4^{th}$ order, 4-stage, single step, explicit

This is the original Runge-Kutta method, developed in 1904.

One of the best ODE solvers, for non-stiff ODEs.

In Matlab: rk4

General form of RK: $Y_{n+1} = Y_n + \Delta t \cdot \sum_{k=1}^{s} b_k \cdot f(t_n + c_k \Delta t, \; Y_{nk})$

$c_k = \sum_{\ell=1}^{s} a_{k\ell}$, with $Y_{nk} = \Delta t \cdot \sum_{\ell=1}^{s} a_{k\ell} \cdot f(t_n + c_\ell \Delta t, \; Y_{nk})$, $k = 1, \dots, s$

usually presented by a Butcher array:
$$\begin{array}{c|ccc} c_1 & a_{11} & \cdots & a_{1s} \\ \vdots & \cdots & \cdots & \cdots \\ c_s & a_{s1} & \cdots & a_{ss} \\ \hline & b_1 & \cdots & b_s \end{array} \qquad c_k = \sum_{\ell=1}^{s} a_{k\ell}$$

Explicit if $[a_{k\ell}]$ is lower triangular, else implicit

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$ explicit Euler (forward)

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$$ implicit backward Euler

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$ explicit trapezoidal $p = 2$, $s = 2$

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$ classical RK4

RKF45 (Runge-Kutta-Fehlberg, 1960): uses an RK4 (not classical) & an RK5 to estimate error, using only 6 evaluations, for adaptiveness (adjust $\Delta t$, half or double it). Order 4, 6 evaluations / step.

One of the best adaptive schemes.

Always worth trying RK4 and RKF45 to compare with other integrators.

## Multistep methods

use $s$ steps to update $Y_{n+s}$, achieve order $s$: $\mathcal{O}(h^s)$

Adams-Bashforth: explicit, Adams-Moulton: implicit

e.g. 2-step Adams-Bashforth: uses $Y_n$, $Y_{n+1}$ for $Y_{n+2}$: $2^{nd}$ order, explicit

$$Y_{n+2} = Y_{n+1} + \frac{3}{2} h \cdot f(t_{n+1}, Y_{n+1}) - \frac{1}{2} h f(t_n, Y_n)$$

e.g. 2-step Adams-Moulton: uses $Y_n, Y_{n+1}, Y_{n+2}$ for $Y_{n+2}$: $3^{rd}$ order, implicit

$$Y_{n+2} = Y_{n+1} + h \cdot \left[\frac{5}{12} f(t_{n+2}, Y_{n+2}) + \frac{8}{12} f(t_{n+1}, Y_{n+1}) - \frac{1}{12} f(t_n, Y_n)\right]$$

Often used predictor-corrector style: predict $Y_{n+s}$ by Adams-Bashforth
correct by Adams-Moulton, $\mathcal{O}(h^s)$ overall

Efficient for very high orders, good for adaptive

General form:

$$Y_{n+s} + a_{s-1} Y_{n+s-1} + \dots + a_0 Y_n =$$
$$= h \cdot \left[b_s \cdot f(t_{n+s}, Y_{n+s}) + b_{s-1} \cdot f(t_{n+s-1}, Y_{n+s-1}) + \dots + b_0 \cdot f(t_n, Y_n)\right]$$

coefficients $a_i$, $b_i$ chosen for high order and easy implementation.
Explicit if $b_s = 0$, else implicit.