2.6 The Runge-Kutta Method

it's actually one of many methods in a family of methods called the Runge-kutta Methods.

(Euler and Improved Euler are members of this family)

problem: solve $\frac{dy}{dx} = f(x,y) \quad y(x_0) = y_0$

Euler: $y_{n+1} = y_n + \underbrace{f(x_n, y_n)\, h}_{\text{beginning of interval only}}$

Improved Euler: $y_{n+1} = y_n + \frac{1}{2}\left[\underbrace{f(x_n, y_n)}_{\text{beginning}} + \underbrace{f(x_{n+1}, y_{n+1})}_{\text{end}}\right] h$

today: Runge-kutta (order 4) $\rightarrow$ also include midpoint

Runge-Kutta:

$$y_{n+1} = y_n + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

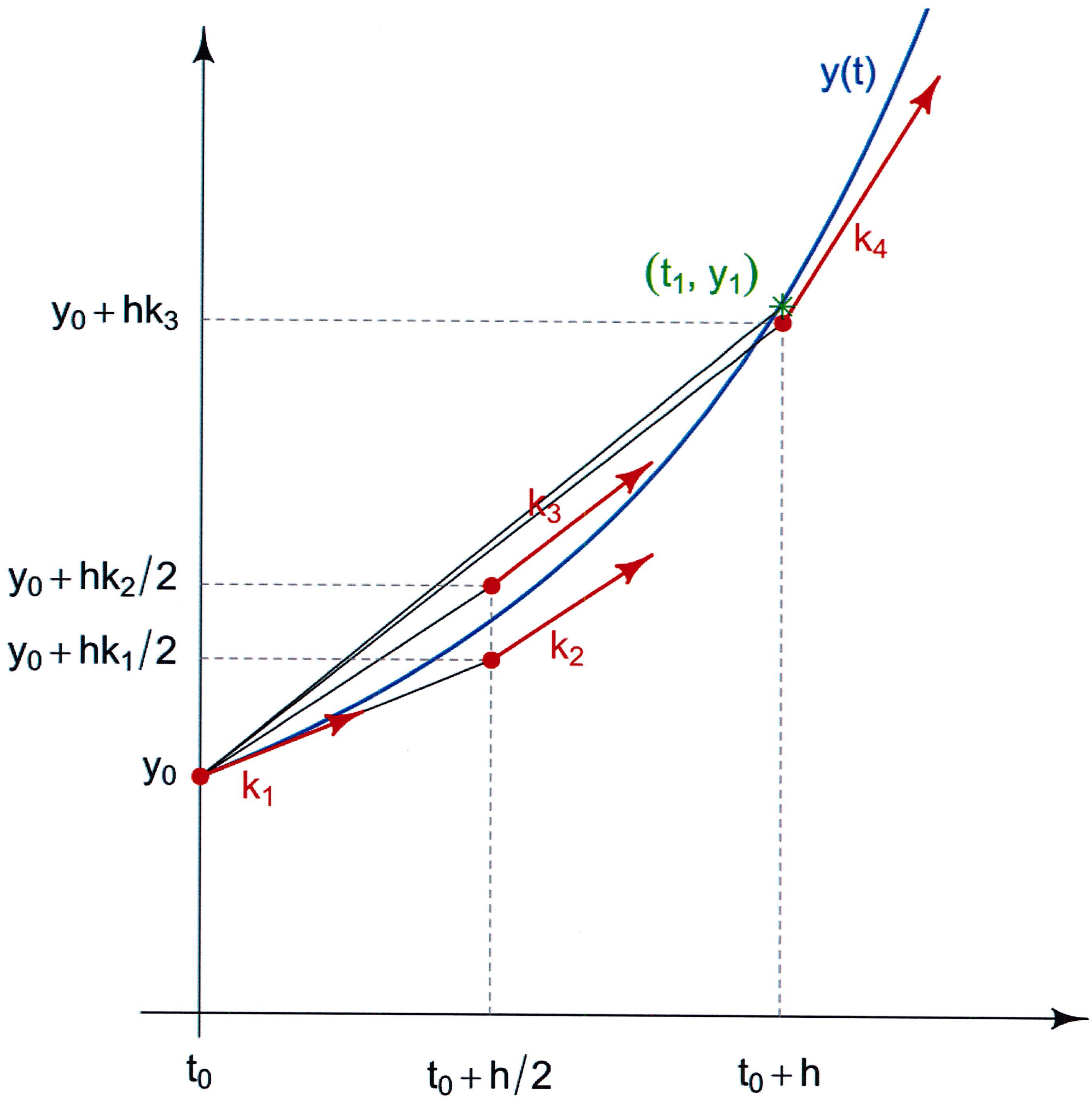
$k_1 = f(x_n, y_n)$ slope at left

$k_2 = f(x_n + \frac{1}{2}h,\ y_n + \frac{1}{2}h\,\underline{\underline{k_1}})$ slope at midpt using Euler to estimate y

$k_3 = f(x_n + \frac{1}{2}h,\ y_n + \frac{1}{2}h\,\underline{\underline{k_2}})$ slope at midpt using Backward Euler to estimate y

$k_4 = f(x_{n+1},\ y_n + h k_3)$ slope at right using Euler w/ slope at midpt

weighted average of these four slopes



y(t)
k4
(t1, y1)
y0 + hk3
k3
y0 + hk2/2
y0 + hk1/2
k2
y0
k1
t0
t0 + h/2
t0 + h

RK: analogous to Simpson's Rule in numerical integration
(Improved Euler is trapezoidal rule)

Advantage: ~~given~~ smaller errors for given step size h

Euler (forward/backward): local error proportional to h^2
cumulative proportional to h

Improved Euler / Heun's: local proportional to h^3
Cumulative proportional to h^2

RK: local proportional to h^5
cumulative proportional to h^4

<u>example</u> $y' = 1 - x + 4y \quad y(0) = 1$

using $h = 0.1$ to estimate $y(0.1)$

$x_0 = 0$

$y_0 = 1$

$x_1 = x_0 + h = 0.1$

$k_1 = f(x_0, y_0) = 1 - x_0 + 4y_0 = 1 - 0 + 4 = 5$

$k_2 = f(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}hk_1)$

$= 1 - (x_0 + \frac{1}{2}h) + 4(y_0 + \frac{1}{2}hk_1) = 5.95$

$k_3 = f(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}hk_2)$

$= 1 - (x_0 + \frac{1}{2}h) + 4(y_0 + \frac{1}{2}hk_2) = 6.14$

$k_4 = f(x_1, y_0 + hk_3) = 7.356$

$y_1 = y_0 + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) = 1.6089333$

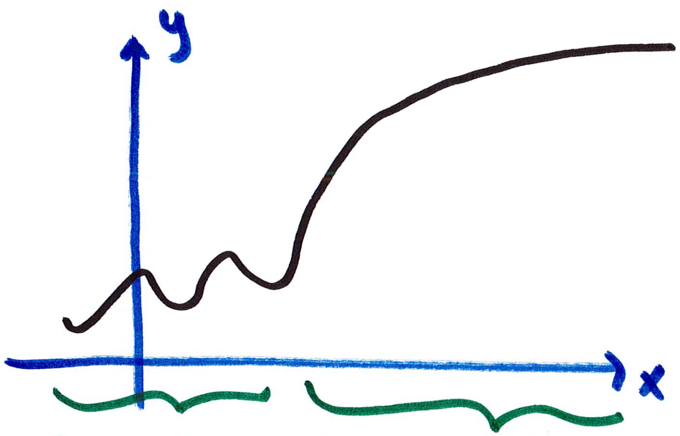
RK w/ $h=0.1$ $\rightarrow$ 1.6089333

exact $\rightarrow$ 1.6090418

RK w/ $h=0.05$ $\rightarrow$ 1.6090338
(2 steps, 8 stages)

Euler		Backward Euler	Heun's
$h=0.1$	$\rightarrow$ 1.5	$\rightarrow$ 1.81667	$\rightarrow$ 1.595
" $h=0.01$	$\rightarrow$ 1.59529	" " $\rightarrow$ 1.62366	" $\rightarrow$ 1.60886
" $h=0.005$	$\rightarrow$ 1.60206	" " $\rightarrow$ 1.61624	" $\rightarrow$ 1.609

step size:



fast changing — need "small" h

relatively straight — "large" h ok

modern computational tools (e.g. Matlab) can use adaptive step size

Matlab: $\underbrace{\text{ode45}}$, ode23, ode78, ode89, → all RK methods

uses 4th order and 5th order simultaneous

uses the difference at each location to estimate error and adjust step size

not all Matlab solvers are RK methods

e.g. ode113 → Adams-Bashforth-Moulton

this is a linear multi-step method → uses past histories to move forward

RK → single step : uses info at each point in isolation