2.6 The Runge-Kutta Method

large family of numerical methods

we will look at 4th order Runge-Kutta (RK4) (MATLAB "ode45" is a version of this)

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

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

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

RK4: sample slopes at the beginning, end, and middle.

RK4 Algorithm

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

h: step size (fixed)

$k_1 = f(x_n, y_n)$ beginning

$k_2 = f(x_n + \frac{1}{2}h,\ y_n + \frac{1}{2}hk_1)$ slope at mid-point using Euler w/ k_1 as slope

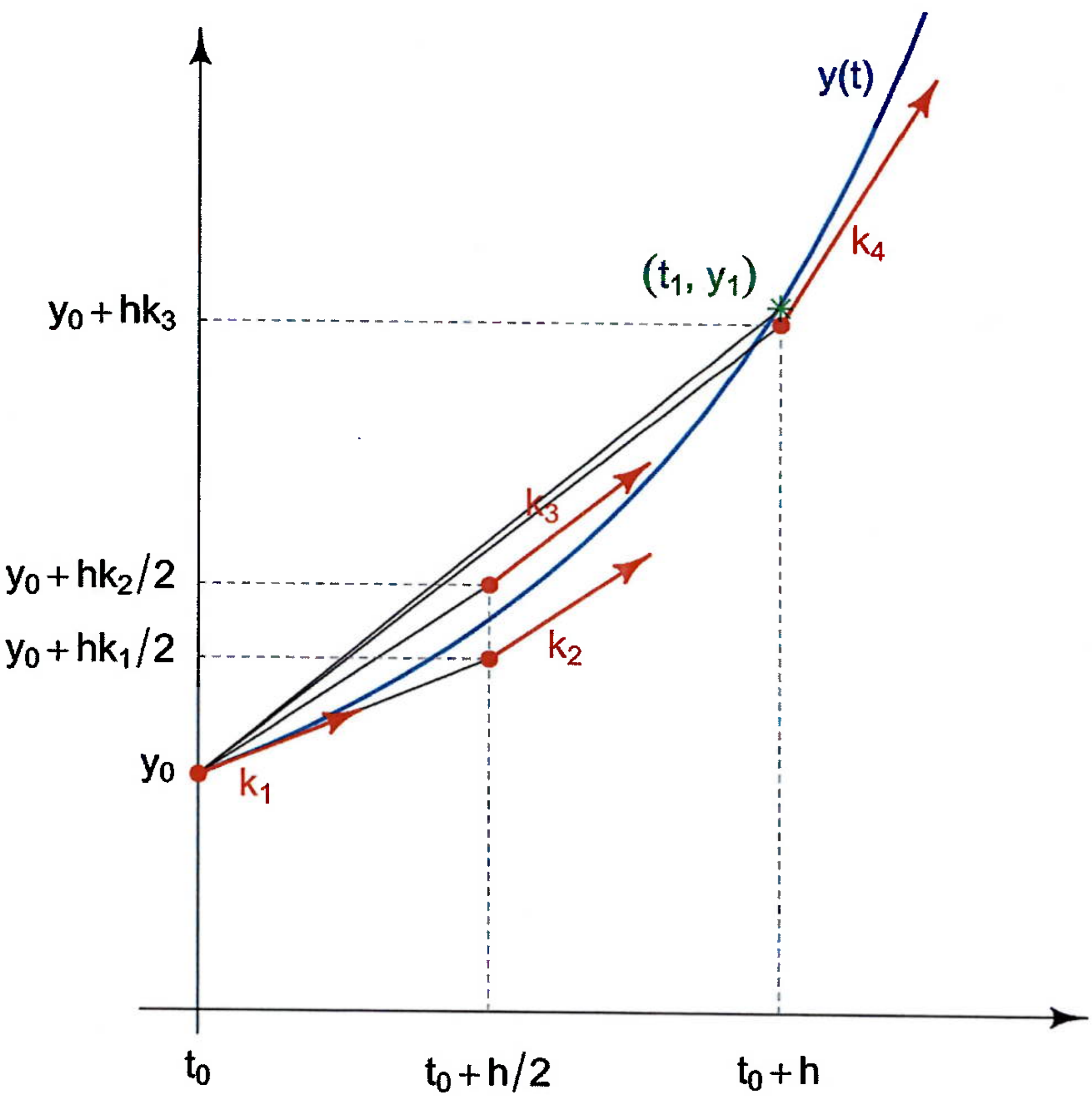
$k_3 = f(x_n + \frac{1}{2}h,\ y_n + \frac{1}{2}hk_2)$ updated slope at ~~po.~~ mid-point (basically backward Euler)

$k_4 = f(x_{n+1},\ y_n + hk_3)$ slope at end using Euler w/ slope k_3

weighted average (more weight on mid-point slopes)

then to the end point

(repeat)



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

advantages of RK4 over Euler

- stability (weigh mid-point more because extrapolation is usually not as good)

- minimizes the error

Euler: local error ~~propo~~ proportional to h^2

cumulative error " " h

Improved Euler (RK2): local proportional to h^3

cumulative " " h^2

RK4: local proportional to h^5

cumulative " " h^4

Same h → MUCH better estimate from RK4 example $y' = 1 - x + 4y \quad y(0) = 1$

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

$x_0 = 0$

$y_0 = 1$

$x_1 = 0 + 0.1$ (target)

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

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

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

$k_3 = f(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}hk_2) = 6.14$

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

now use weighted avg to predict y_1

$$y_1 = y_0 + \frac{h}{6}\left[k_1 + 2k_2 + 2k_3 + k_4\right] = 1.6089333$$

accuracy?

RK4 w/ $h=0.1 \rightarrow 1.6089333$

exact $\rightarrow 1.6090418$

RK4 w/ $h=0.05 \rightarrow 1.6090338$

Euler $h=0.1 \rightarrow 1.5$

$h=0.01 \rightarrow 1.59529$ 10 steps

$h=0.005 \rightarrow 1.60206$ 20 steps

1 RK4 step (4 sub steps) is better than 20 Euler steps!

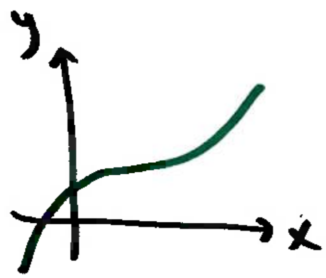
Imp. Euler $h=0.1 \rightarrow 1.595$

$h=0.01 \rightarrow 1.60886$ 10 steps w/ 2 sub steps each

RK4 beats Euler and Imp. Euler even w/ fewer total steps

ode45 in MATLAB is 4th/5th-order (program decides)

↳ uses adaptive step size (program decides h on the fly)



other in MATLAB

ode23
ode78
ode89

ode113 is NOT RK