Initial Value Problem

Raja Damanik, M.Sc.

Problem

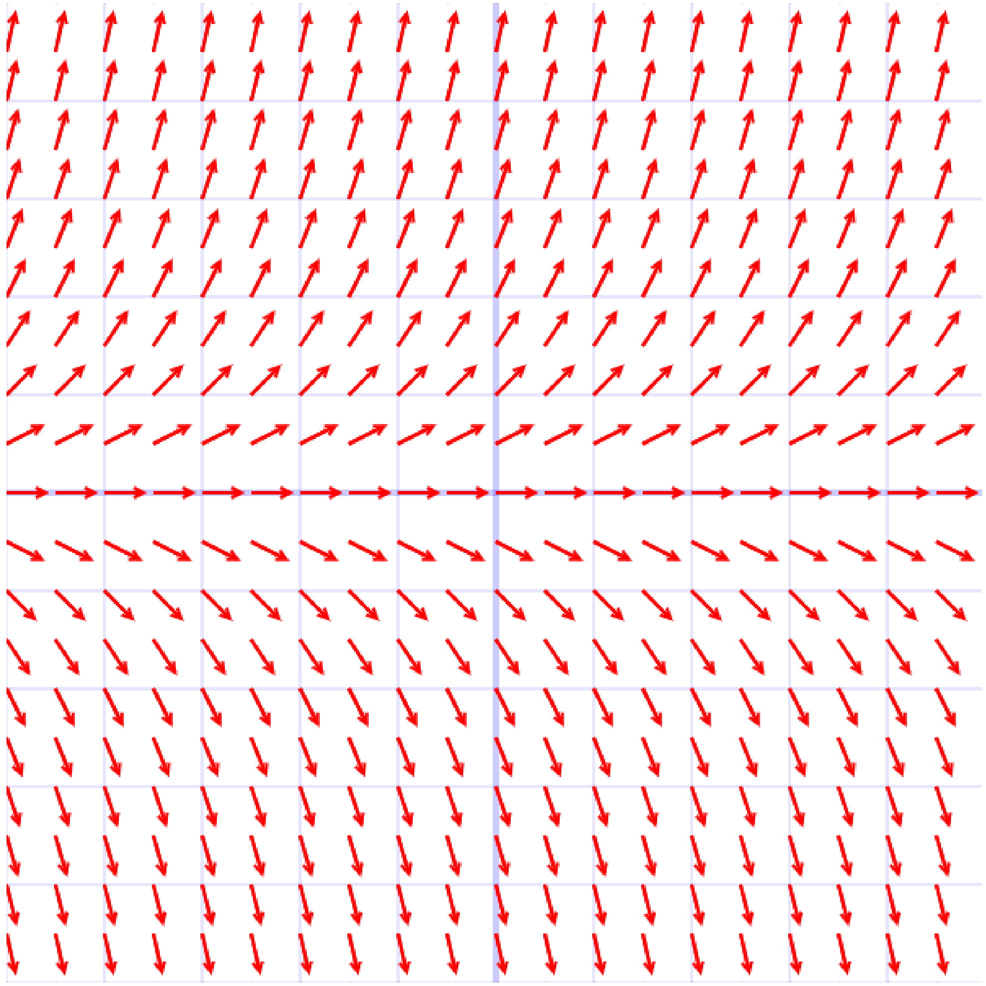
- Think of y be a function of x.
- An ordinary differential equation (ODE) is an equation that tells us about how y changes for certain value of x and y.
  - Example 1: How many humans are there in 10 years if now there are 1B humans?
  - Example 2: How many rabbits and wolves are there after some time if now there are 100 rabbits and 20 wolves? (Check: Lotka-Volterra prey-predator model)

Problem

- The simplest: first order ordinary differential equation
$$y' = f(x, y).$$
  - Example 1: $y' = y$
  - Example 2: $y' = y\cos(x) + \sqrt{xy}$
- More sophisticated:
  - Second-order, third-order, ... ODE
  - Stochastic Differential Equation (SDE)
  - Partial Differential Equation (PDE) -> real world modelling ☺

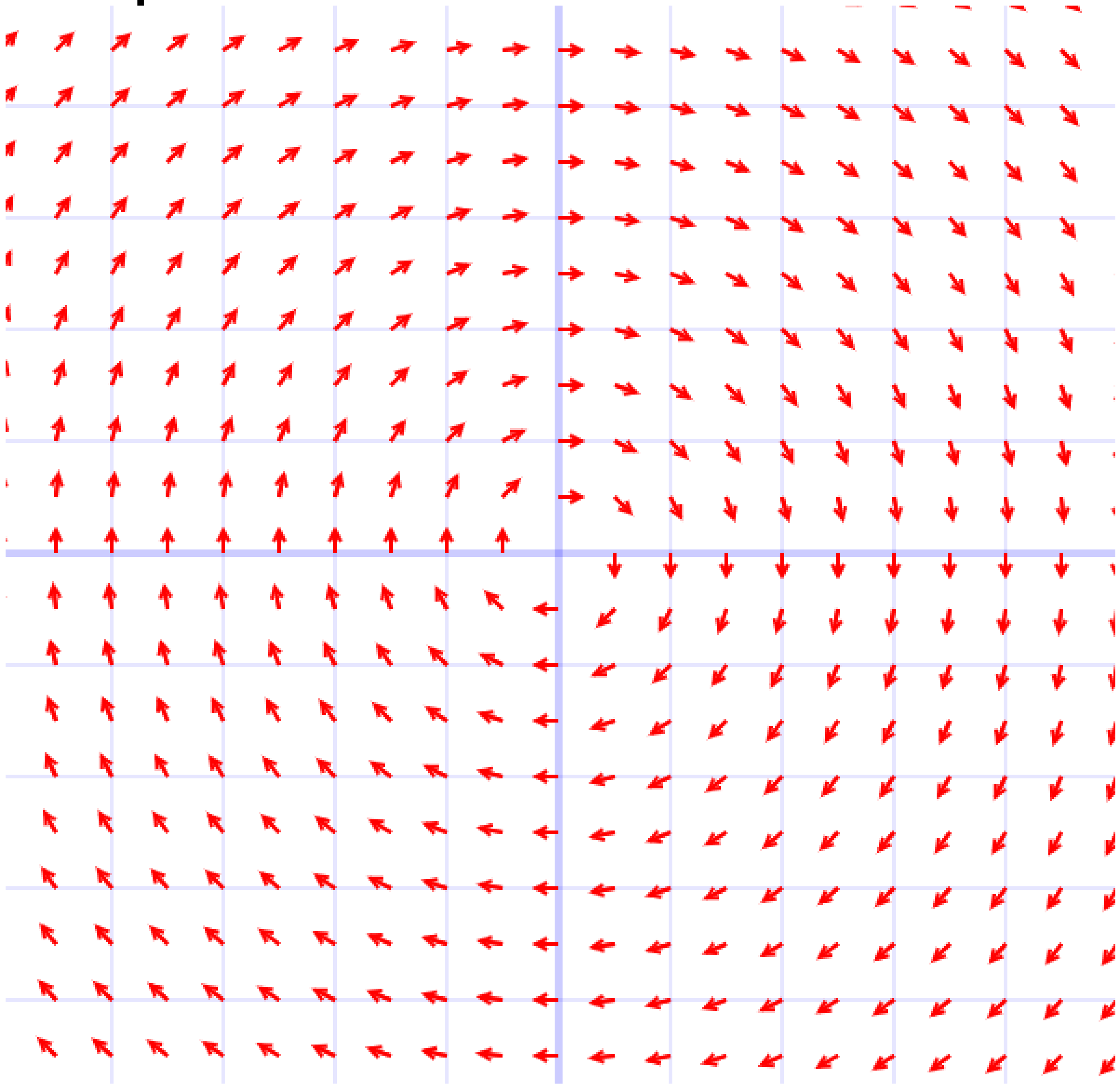
Gradient Field as Phase Space

- Let $y' = f(x, y)$ be our ODE.
- At each point (x, y), we can draw a small vector with gradient $f(x, y)$.
- Example with $y' = y$



Gradient Field as Phase Space

- Let $y' = f(x, y)$ be our ODE.
- At each point (x, y), we can draw a small vector with gradient $f(x, y)$.
- Example with $y' = -\frac{x}{y}$.



Play with gradient field

- http://user.mendelu.cz/marik/EquationExplorer/vectorfield.html

Initial Value Problem

- Given a differential equation
$$y' = f(x, y)$$
with a specified value of $y(x_0) = x_0$.
- This will generate a curve of y starting at $y(x_0)$ and following the gradient field of $y' = f(x, y)$.
- This is called **initial value problem (IVP).**

Numerical Method for IVP

- Euler Method
- Backward Euler Method
- Implicit Euler Method
- Runge-Kutta Method
  - RK-2
  - RK-3
  - RK-4
  - RKF
  - etc.

Numerical Method for IVP

- Main idea:

DISCRETIZE!

Euler Method

- Given an IVP $y' = \boldsymbol{f(x, y)}$ with $y(x_0) = y_0$.
- Fix a step-size h.
- Compute new (x_n, y_n) at each point:
  - Compute

$$x_{n+1} = x_n + h$$
$$y_{n+1} = y_n + h\boldsymbol{f(x_n, y_n)}$$

Euler Method: The Why

- Just using a straight line (with a suitable gradient!) to go to the next point.

Backward Euler Method: The Why

- Also use a straight line to go to the next point, but the gradient matches at the next point. ☺

Backward Euler Method

- Given an IVP $y' = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y})$ with $y(x_0) = y_0$.
- Fix a step-size h.
- Compute new (x_n, y_n) at each iteration:
  - Compute
$$\boldsymbol{x_{n+1}} = x_n + h$$
  - Find y_{n+1} that satisfies:
$$\boldsymbol{y_{n+1}} = y_n + h\boldsymbol{f}(\boldsymbol{x_{n+1}}, \boldsymbol{y_{n+1}})$$

Backward Euler Method

- Given an IVP $y' = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y})$ with $y(x_0) = y_0$.
- Fix a step-size h.
- Compute new (x_n, y_n) at each iteration:
  - Compute
$$\boldsymbol{x_{n+1}} = x_n + h$$
  - Find y_{n+1} that satisfies:
$$\boldsymbol{y_{n+1}} = y_n + h\boldsymbol{f}(\boldsymbol{x_{n+1}}, \boldsymbol{y_{n+1}})$$

might be painful to solve, might need non-linear equation solve

Implicit Euler Method

- Use some parameter θ.
- Compute $x_{n+1} = x_n + h$.
- Find y_{n+1} that satisfies

$$y_{n+1} = y_n + h[\theta f(x_n, y_n) + (1 - \theta) f(x_{n+1}, y_{n+1})]$$

- Use “average gradient”.

Runge-Kutta Method

- It turns out using Euler's method is very ez: just generate new data points (x_n, y_n) at each iteration, using some kind of "gradient approximation".
- Runge-Kutta develops this idea even further.
- Will introduce Butcher tableau.

Butcher tableau

- A table for specifying the “mixing-coefficients to approximate the gradient”.
- Connecting derivative of a function and tree-graphs.
- Due to John C. Butcher.

List of Runge-Kutta methods

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

RK-2 (Heun's method)

$$
\begin{array}{c|ccc}
0 & 0 & 0 & 0 \\
1/2 & 1/2 & 0 & 0 \\
1 & -1 & 2 & 0 \\
\hline
 & 1/6 & 2/3 & 1/6
\end{array}
$$

RK-3

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

RK-4

List of Runge-Kutta methods

0						
1/4	1/4					
3/8	3/32	9/32				
12/13	1932/2197	−7200/2197	7296/2197			
1	439/216	−8	3680/513	-845/4104		
1/2	−8/27	2	−3544/2565	1859/4104	−11/40	
	16/135	0	6656/12825	28561/56430	−9/50	2/55

RK-Fehlberg method (trunctaed version)

How to read the Butcher tableau

$$
\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{21} & & & & \\
c_3 & a_{31} & a_{32} & & & \\
\vdots & \vdots & & \ddots & & \\
c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & \\
\hline
 & b_1 & b_2 & \cdots & b_{s-1} & b_s
\end{array}
$$

Butcher tableau

Compute:

$$k_1 = f(x_n, y_n)$$

$$k_2 = f\big(x_n + hc_2, y_n + h(a_{21}k_1)\big)$$

$$k_3 = f\big(x_n + hc_3, y_n + h(a_{31}k_1 + a_{32}k_2)\big)$$

…

$$k_s = f(x_n + hc_s, y_n + h\big(a_{s1}k_1 + \cdots + a_{s,s-1}k_{s-1}\big)$$

$$\boldsymbol{y_{n+1} = y_n + h(b_1 k_1 + b_2 k_2 + \cdots + b_s k_s)}$$

RK-2

$$x_{n+1} = x_n + h$$

$$k_1 = f(x_n, y_n)$$
$$k_2 = f(x_n + h, y_n + hk_1)$$
$$y_{n+1} = y_n + h\left(\frac{1}{2}k_1 + \frac{1}{2}k_2\right).$$

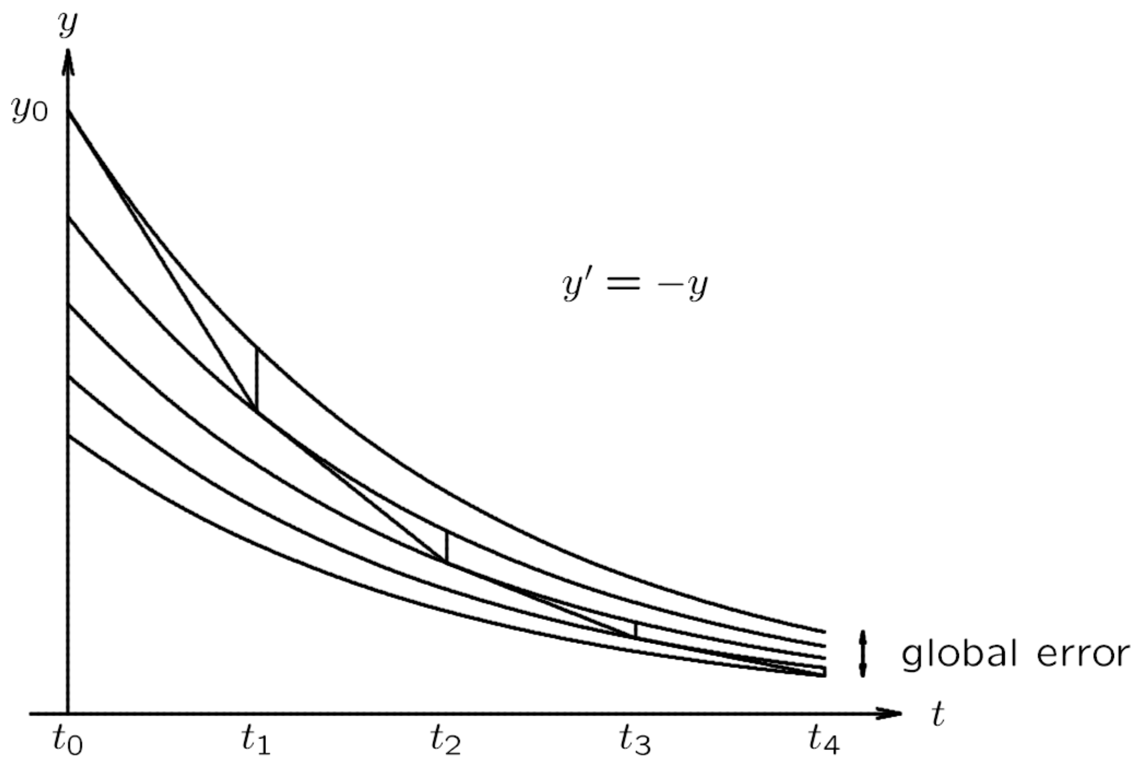
$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \end{array}$$

Euler method as Runge-Kutta instance

- Euler method can be written with Butcher tableau as well, so it is an instance of Runge-Kutta method.
- Backward Euler too.
- Implicit Euler too.

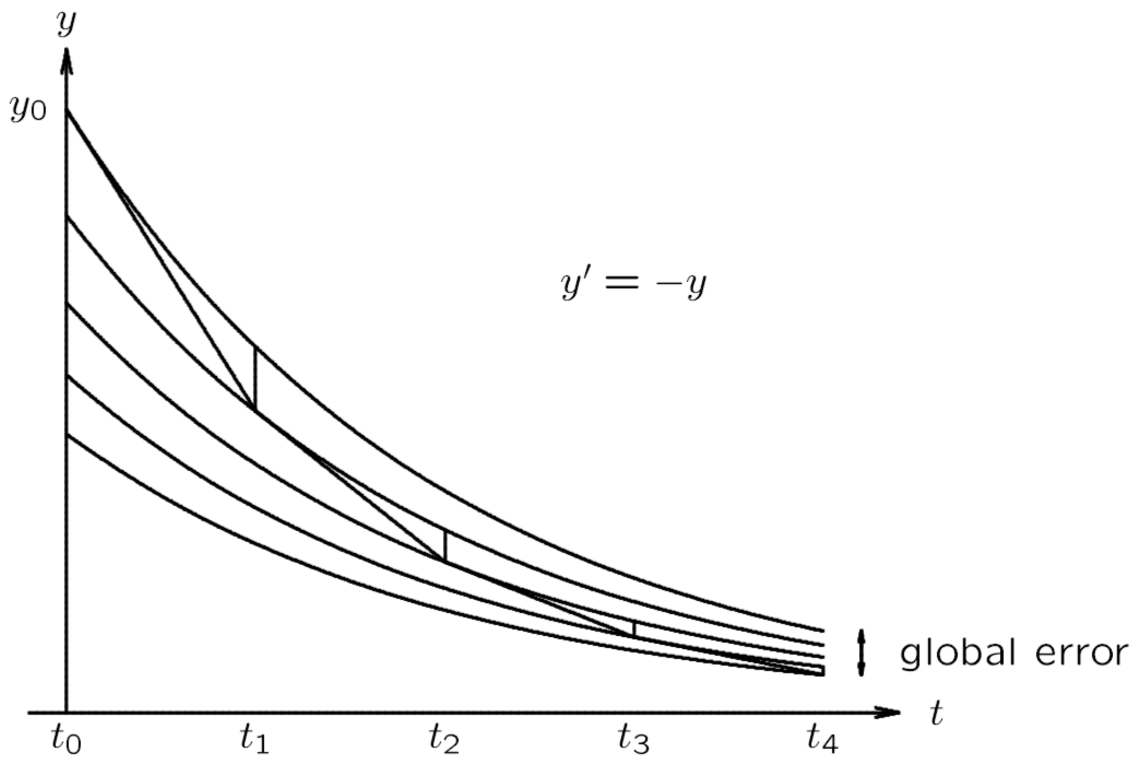
Accuracy

- Local Error
- Global Error



Accuracy

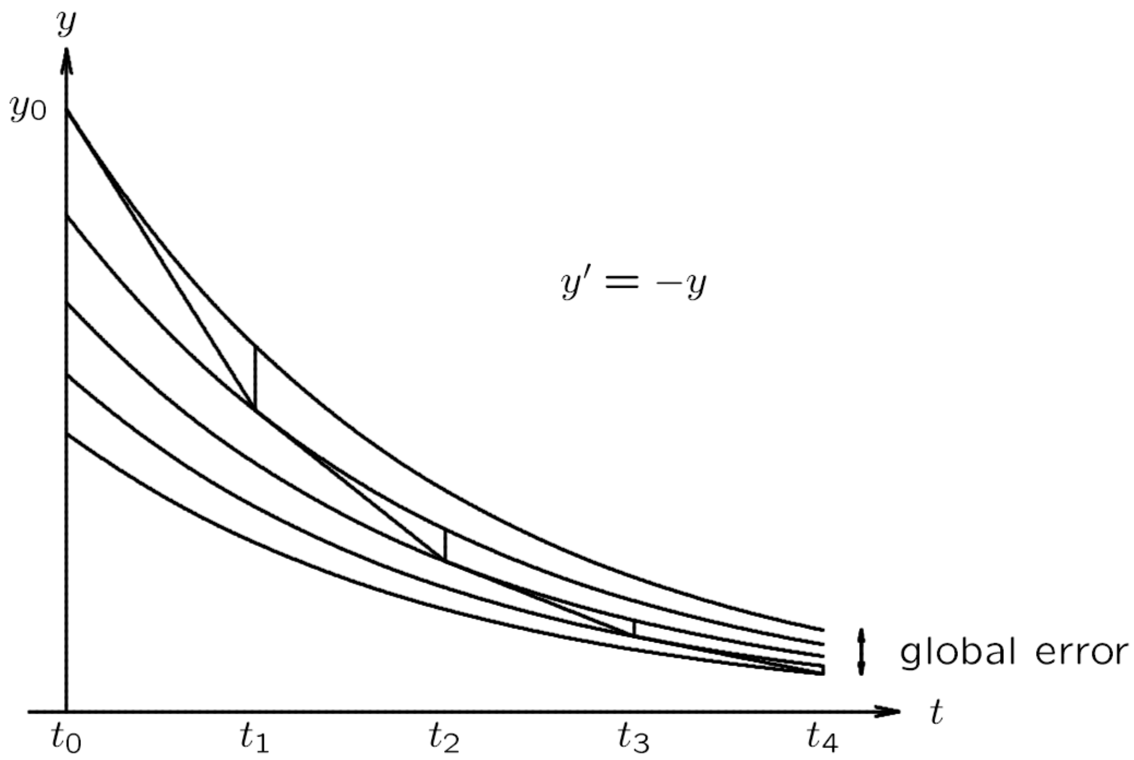
- Local Error:
  - One-step error
  - The error of y_{i+1} compared to the exact solution of IVP
    $y' = f(x, y) \quad y(x_i) = y_i$
- Global Error



Accuracy

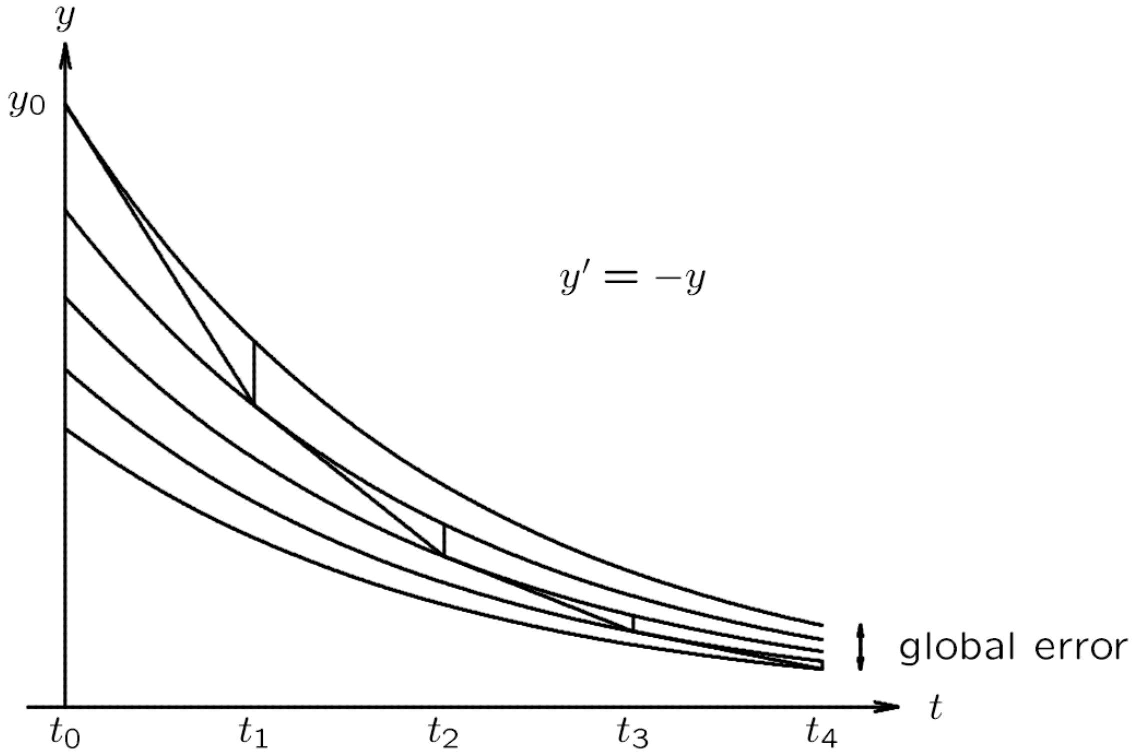
- Local Error:
  - One-step error
  - The error of y_{i+1} compared to the $y(x_{i+1})$ exact solution of IVP $y' = f(x, y) \quad y(x_i) = y_i$
- Global Error
  - Multi-step error (globally)
  - The maximum error of y_{n+1} compared to the $y(x_{n+1})$ exact solution of IVP

$$y' = f(x, y) \quad y(x_0) = y_0$$



Accuracy

- If a method has local error $O(h^{p+1})$, then it has global error $O(h^p)$.
- We call a method has **order p** when it has local error $O(h^{p+1})$.
- RK-2 has local error $O(h^3)$ and global error $O(h^2)$, etc.



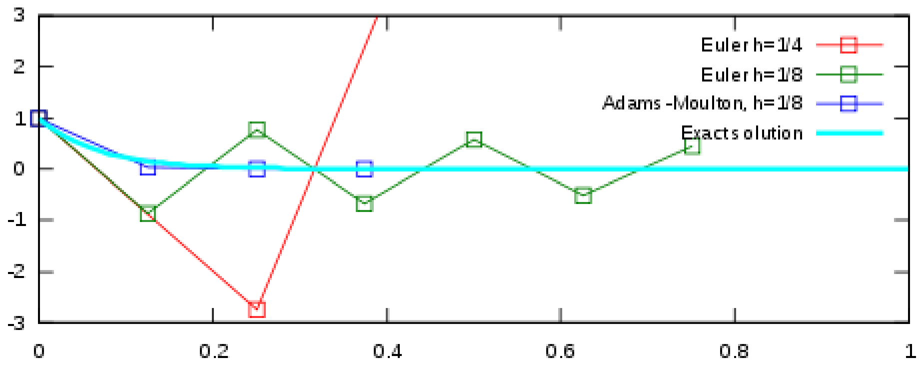
Runge-Kutta: Accuracy vs Number of stages

- Runge-Kutta has s intermediate steps, e.g:
  - RK-2 has 2 intermediate steps
  - RK-3 has 3 intermediate steps
- If a Runge-Kutta has order p, then it has been proved that $s \geq p$ and if $p \geq 5$, then $s \geq p + 1$. But the bound might be not sharp, e.g.

Implicit RK

- Why bother with implicit method?
- Search: Stiff differential equation, e.g.

$$y' = -15y \qquad y(0) = 1$$



Variable Step-Size

- Rule of thumbs:
  - Smaller steps lead to more accurate solution, but more costly to compute
- Thus, need to employ just the right size of the step
  - Subject to the desired level of accuracy
- Need an error estimate
  - For controlling the step-size

Variabel Step-Size Method

- Use a pair of RK of order p & p+1
- share the same k
- error estimate is easily available

$$y_{n+1}^{p} = y_n + h\sum_{i=1}^{m}\hat{b}_i k_i + O(h^p)$$

$$y_{n+1}^{p+1} = y_n + h\sum_{i=1}^{m} b_i k_i + O(h^{p+1})$$

$$E_{est} = h\sum_{i=1}^{m}(\hat{b}_i - b_i)k_i$$

RK-4-5 (Runge-Kutta Fehlberg)

$$\begin{array}{c|cccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\
\frac{3}{8} & \frac{3}{32} & \frac{9}{32} & 0 & 0 & 0 & 0 \\
\frac{12}{13} & \frac{1932}{2197} & \frac{-7200}{2197} & \frac{7296}{2197} & 0 & 0 & 0 \\
1 & \frac{439}{216} & -8 & \frac{3680}{513} & \frac{-845}{4104} & 0 & 0 \\
\frac{1}{2} & \frac{-8}{27} & 2 & \frac{-3544}{2565} & \frac{1859}{4104} & \frac{-11}{40} & 0 \\
\hline
\hat{b}_i & \frac{25}{16} & 0 & \frac{1408}{2565} & \frac{2197}{4104} & \frac{-1}{5} & 0 \\
b_i & \frac{16}{135} & 0 & \frac{6656}{12825} & \frac{28561}{56430} & \frac{-9}{50} & \frac{2}{55}
\end{array}$$

Variable Step-Size

- Given an IVP: y_0, $f(t,y)$, t_0, and t_f (target)
- Set $h=h_0$; accuracy level TOL
- Compute E_{est}
  - If E_{est} < TOL accept the solution y_1 and proceed till $t=t_f$. If E_{est} is too small, increase h.
  - Else, halve h and recompute E_{est}

Thank You!

- Do not hesitate to ask question!