Lecture 6: Explicit Runge-Kutta Methods

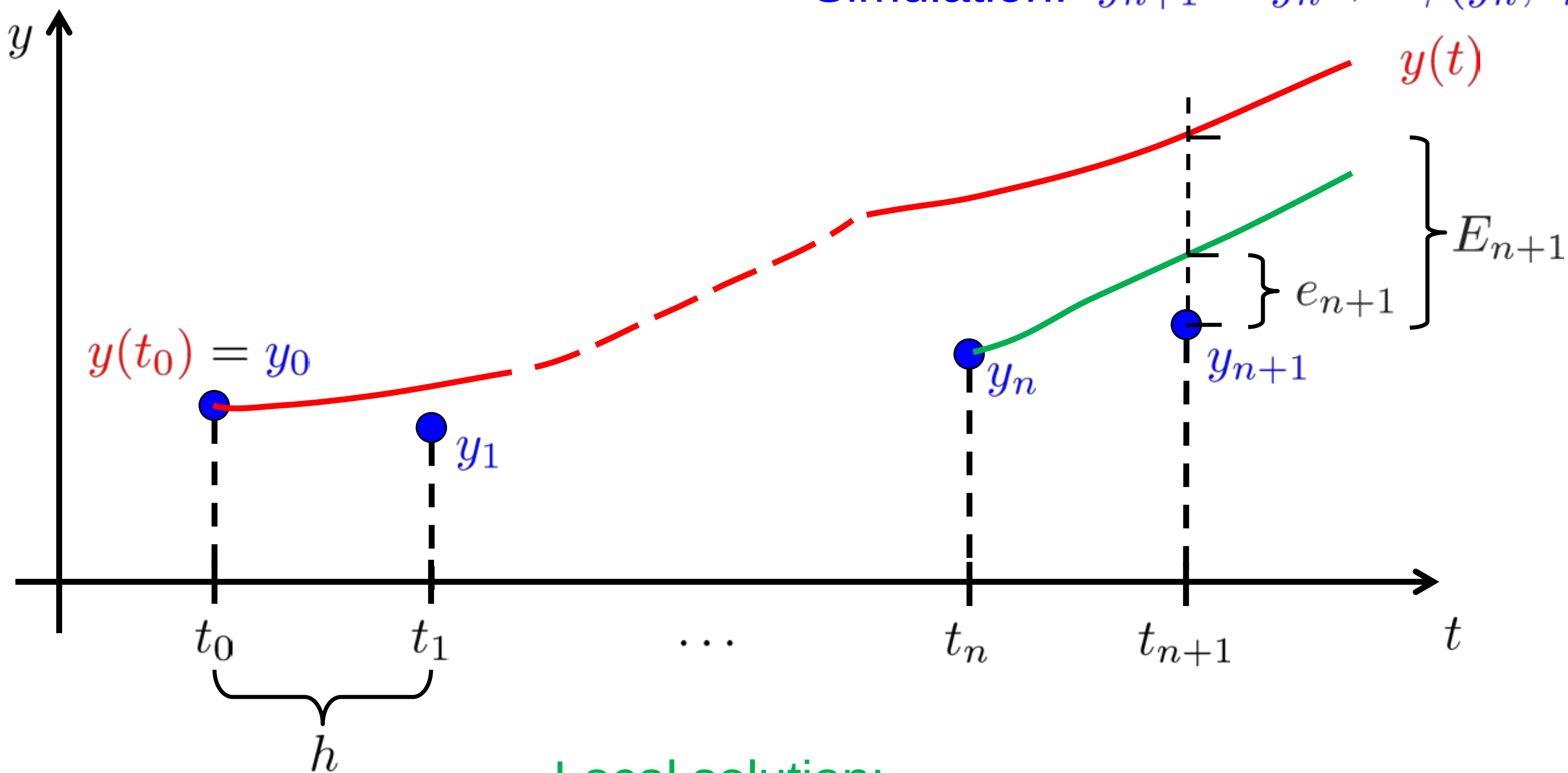
Explicit Runge-Kutta (ERK) methods, and their order and stability

Book: 14.3, 14.4

Recap: Notation

IVP: $\dot{y} = f(y,t), \quad y(t_0) = y_0$

Simulation: $y_{n+1} = y_n + h\phi(y_n, t_n)$

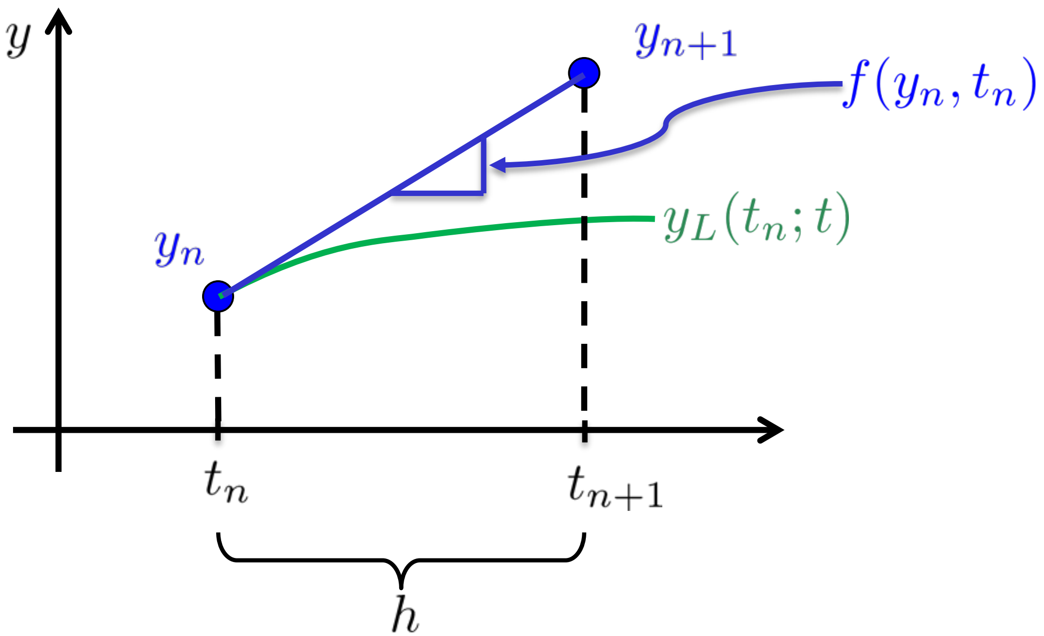


Local solution:

$$\dot{y}_L(t_n; t) = f(y_L(t_n; t), t), \quad y_L(t_n; t_n) = y_n$$

- Local error: $e_{n+1} = y_{n+1} - y_L(t_n; t_{n+1})$
- Global error: $E_{n+1} = y_{n+1} - y(t_{n+1})$
- If local error is $O(h^{p+1})$ then we say method is of order p

Simplest method: Euler



- Slope:

$$\frac{y_{n+1} - y_n}{h} = f(y_n, t_n)$$

- Euler's method:

$$y_{n+1} = y_n + hf(y_n, t_n)$$

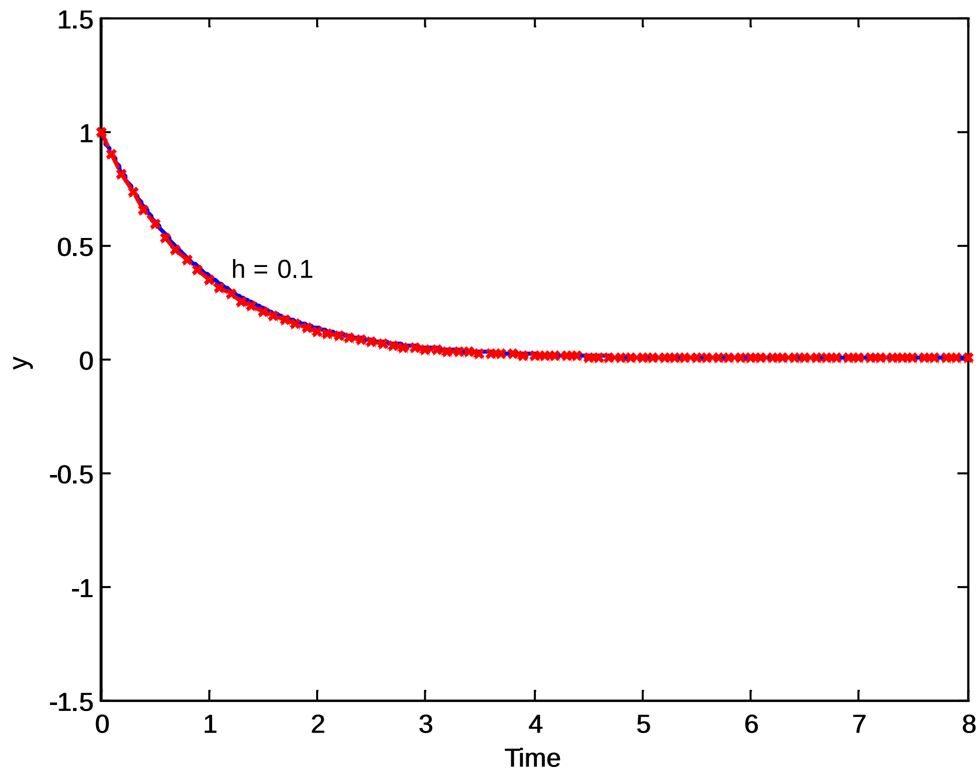
Example Euler's method

ODE: $$\dot{y} = -y, \quad y(0) = 1$$

Euler simulation: $$y_{n+1} = y_n + h\left(-y_n\right), \quad y_0 = 1$$

Example, *h* = 0.1:

n	***t**$_n$*	***y**$_n$*
0	0	1
1	0.1	
2	0.2	
3	0.3	
4	0.4	
…	…	…

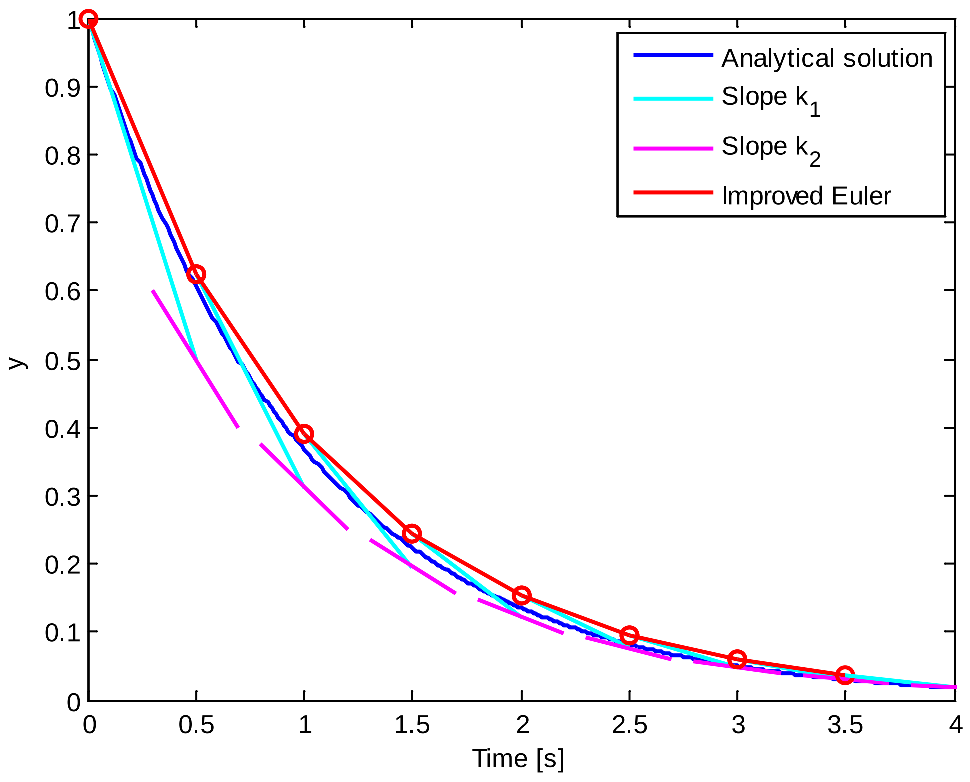


Improved Euler illustration

$$\dot{y} = -y, \quad y(0) = 1$$

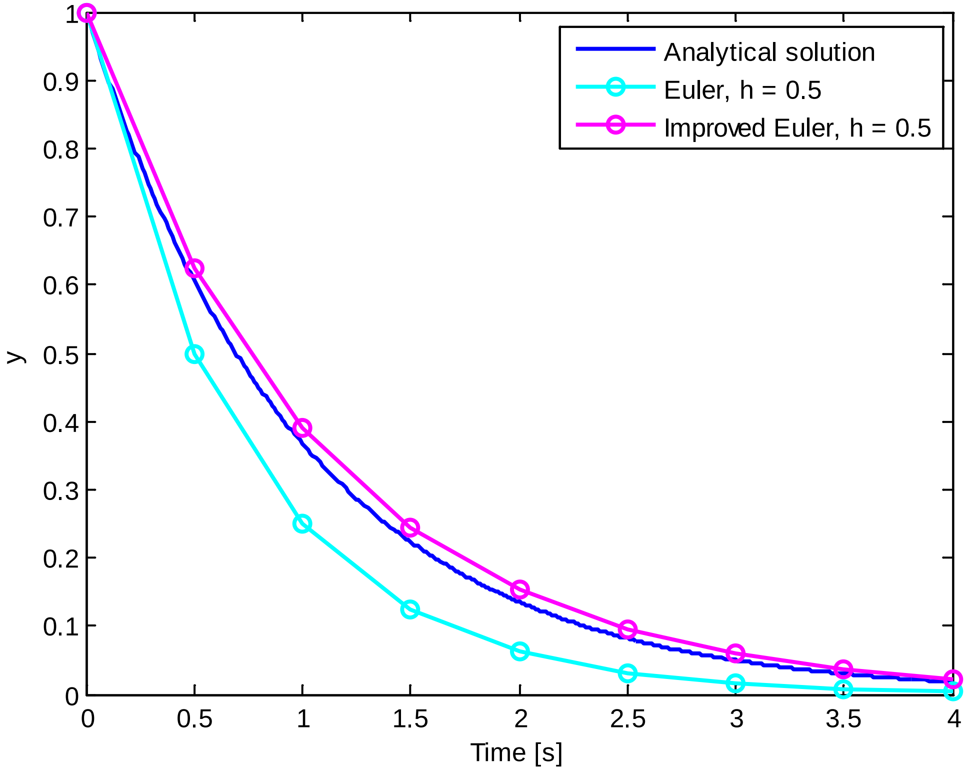
Improved Euler: $$k_1 = f(y_n), \quad k_2 = f(y_n + hk_1)$$

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



Improved Euler vs Euler

$$\dot{y} = -y, \quad y(0) = 1$$



Accuracy and stability

- Lots of different methods, with different complexity. How to quantify their behaviour?

- Two aspects are important: accuracy and numerical stability.
  - Accuracy: How does the local error vary with step-size?
  - Numerical stability: How to avoid that the simulation diverges?

- What decides accuracy and numerical stability?
  - Accuracy: Method and choice of step-size
  - Stability: Method, system eigenvalues, and choice of step-size

- Why are we interested in both accuracy and numerical stability?
  - We always need stability, but stability not enough: Many stable methods are not very accurate!

Recap: Order (accuracy)

- Given IVP:

$$\dot{y} = f(y,t), \quad y(0) = y_0$$

- One-step method:

$$y_{n+1} = y_n + h\phi(y_n, t_n), \quad h = t_{n+1} - t_n$$

- If we can show that

$$y_{n+1} = y_n + hf(y_n, t) + \frac{h^2}{2}\frac{\mathrm{d}f(y_n, t)}{\mathrm{d}t} + \ldots + \frac{h^p}{p!}\frac{\mathrm{d}^{p-1}f(y_n, t)}{\mathrm{d}t^{p-1}} + O(h^{p+1})$$

- Then:
  - Local error is $O(h^{p+1})$
  - Method is order p

Recap: Test system, stability function

- One step method:

$$y_{n+1} = y_n + h\phi(y_n, t_n)$$

- Apply it to scalar test system:

$$\dot{y} = \lambda y$$

- We get:

$$y_{n+1} = R(h\lambda)y_n$$

where $R(h\lambda)$ is stability function

- The method is stable (for test system!) if

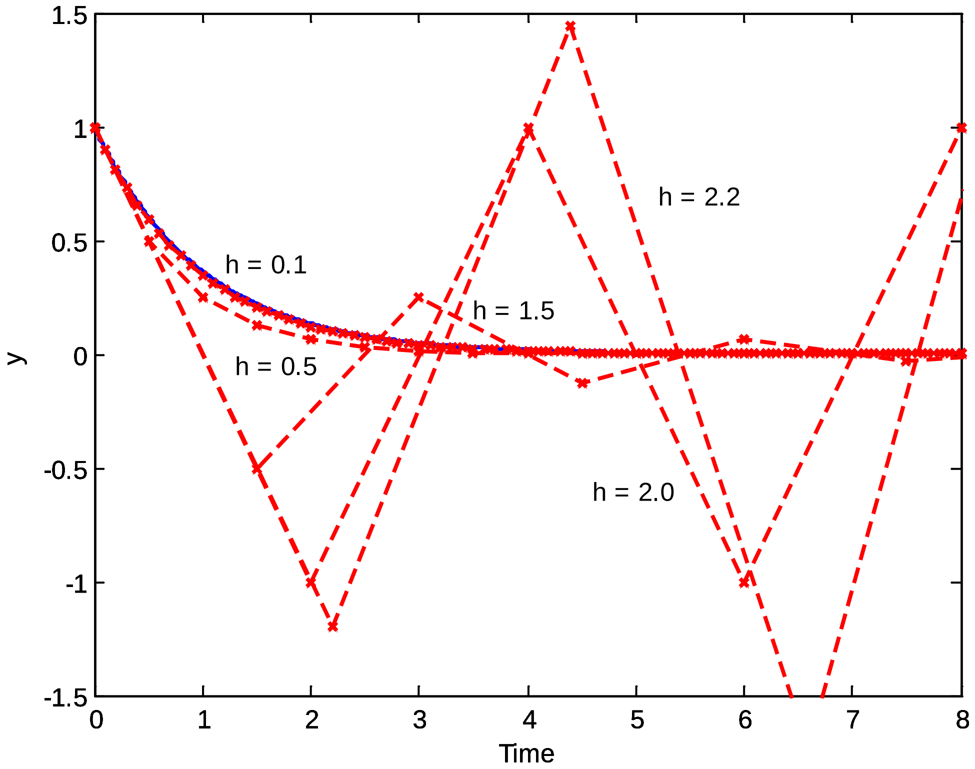
$$|R(h\lambda)| \leq 1$$

Example Euler's method

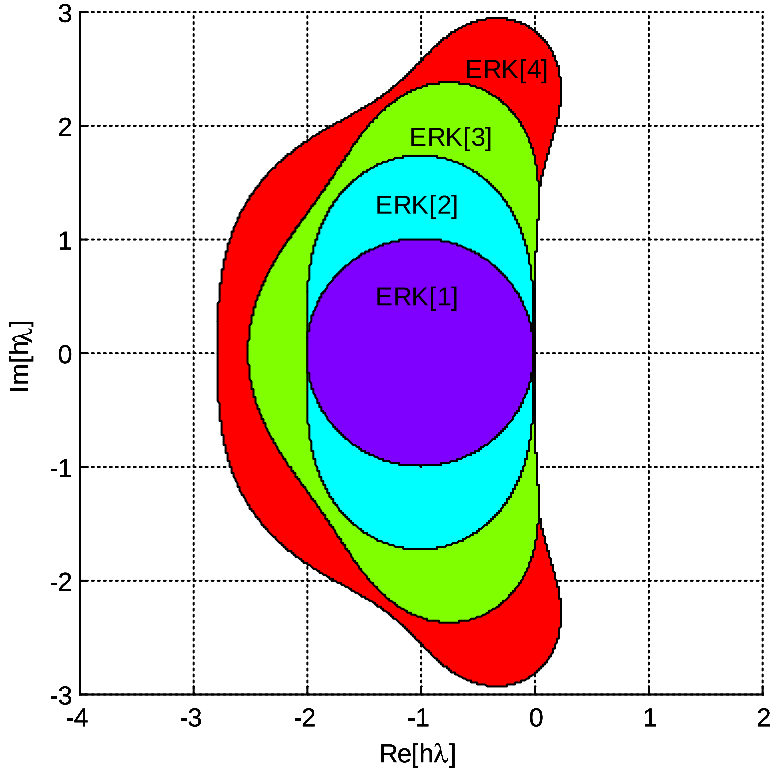
ODE: $$\dot{y} = -y, \quad y(0) = 1$$

Euler simulation: $$y_{n+1} = y_n + h\left(-y_n\right), \quad y_0 = 1$$

Stability: $$|R(h\lambda)| = |1 - h| \leq 1 \Rightarrow 0 \leq h \leq 2$$



Stability regions for ERK methods



Fact: For $1 \leq \sigma \leq 4$, one can devise ERK methods with order $p = \sigma$

- For these methods, per definition

$$y_{n+1} = y_n + hf(y_n, t_n) + \ldots + \frac{h^p}{p!} \frac{\mathrm{d}^{p-1}}{\mathrm{d}t^{p-1}} f(y_n, t_n) + O(h^{p+1})$$

- For test system $\dot{y} = \lambda y$,

$$\begin{aligned} y_{n+1} &= y_n + h\lambda y_n + \ldots + \frac{h^p \lambda^p}{p!} y_n + O(h^{p+1}) \\ &= \left(1 + h\lambda + \ldots + \frac{h^p \lambda^p}{p!}\right) y_n + O(h^{p+1}) \end{aligned}$$

- From before, we know that $y_{n+1} = R_E(h\lambda) y_n$, where $R_E(h\lambda)$ is polynomial of degree less than or equal to $\sigma = p$

That is: For ERK methods with order $p = \sigma$, for $\sigma \leq 4$:

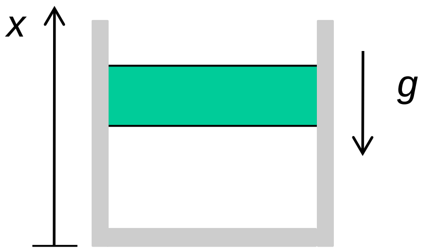
$$R_E(h\lambda) = 1 + h\lambda + \ldots + \frac{h^p \lambda^p}{p!}$$

Case: Pneumatic spring

- Model from Newton's 2nd law:

$$\ddot{x} + g(1 - x^{-\kappa}) = 0$$

"mass-spring-damper with nonlinear spring and no damping"



- On states-space form $\dot{y} = f(y,t)$

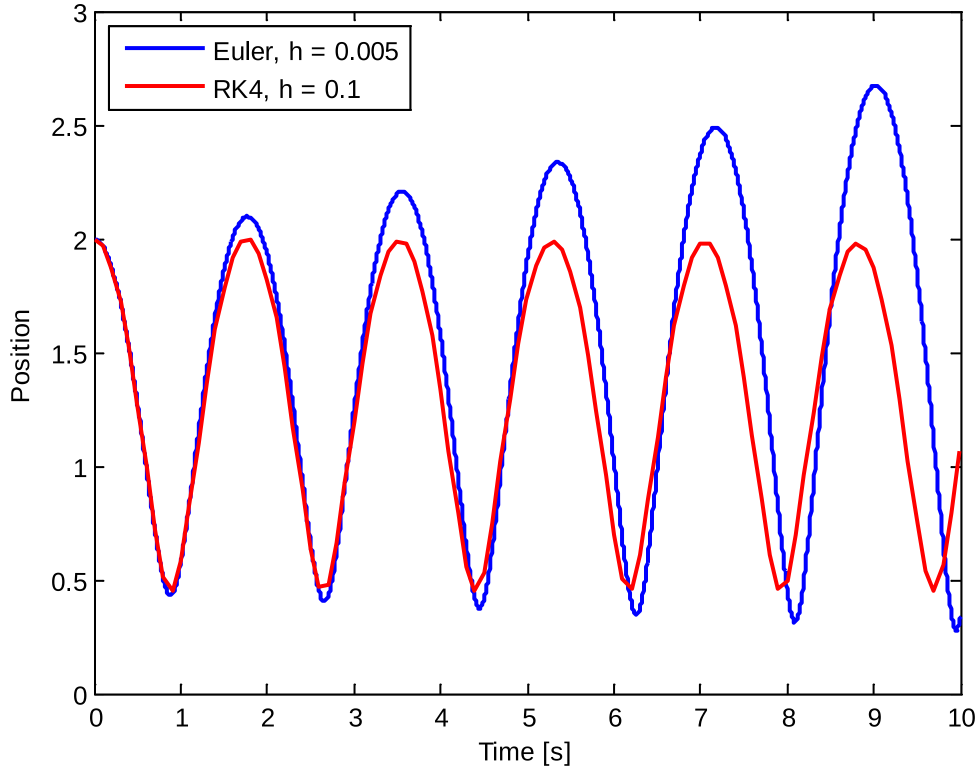
$$\begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \end{pmatrix} = \begin{pmatrix} y_2 \\ -g(1 - y_1^{-\kappa}) \end{pmatrix}$$

- Linearization about equilibrium:

$$\frac{\partial f}{\partial y} = \begin{pmatrix} 0 & 1 \\ -g\kappa & 0 \end{pmatrix}, \qquad \lambda_{1,2} = \pm j\omega_0, \quad \omega_0 = \sqrt{g\kappa} \approx 3.7$$

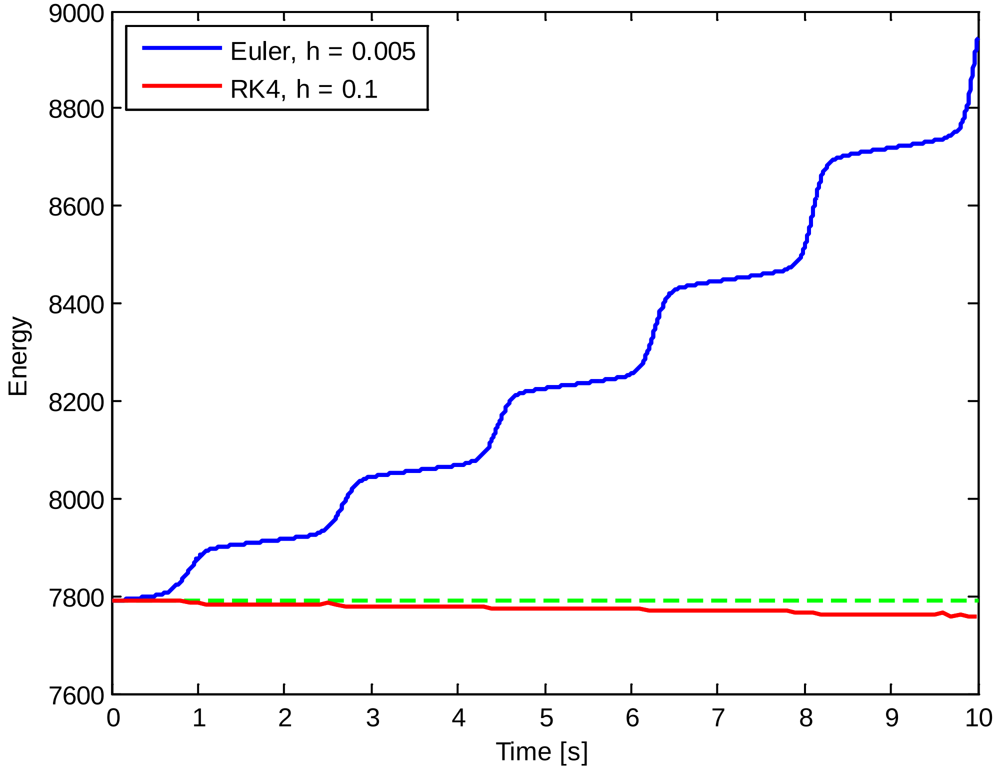
Simulation

Euler: 2000 function evaluations
RK4: 400 function evaluations



- Stability, RK4
  - Theoretical: $\omega_0 h \approx 2.83 \quad \Rightarrow \quad h \approx 0.76$
  - Practically: $h \approx 0.52$

Accuracy: Energy should be constant



Kahoot

- https://play.kahoot.it/#/k/5919b1ba-e564-400e-b63e-d9b2d3fa75cc