Lecture 6: Explicit Runge-Kutta Methods

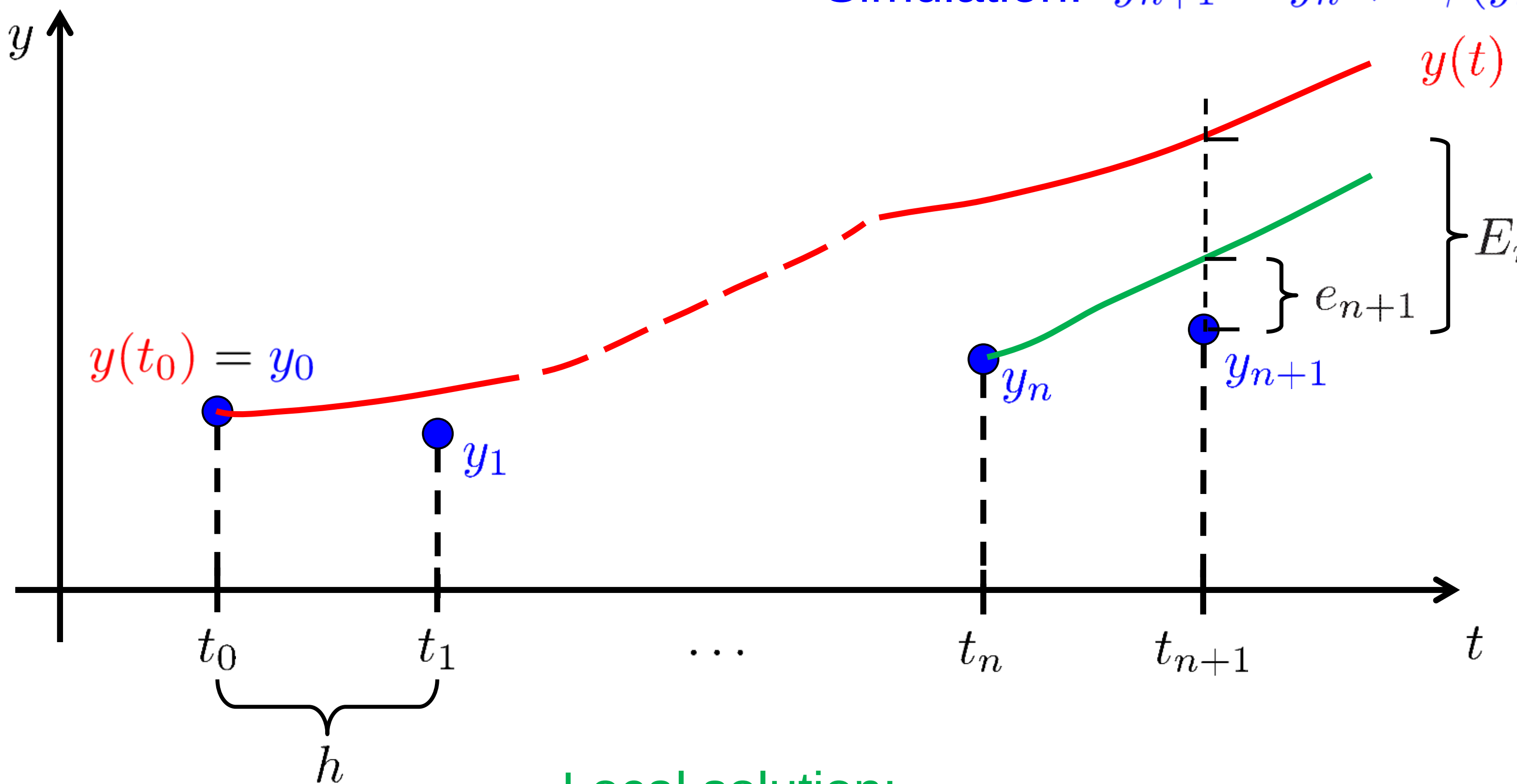
Explicit Runge-Kutta (ERK) methods, and their order and stability

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

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

Linearization

(14.2.4)

- System $\dot{y} = f(y,t)$, $y = (y_1, \ldots, y_d)^{\mathsf{T}}$
- Linearize around operating point y^*: $\Delta\dot{y} = J\Delta y, \quad J = \left.\frac{\partial f}{\partial y}\right|_{y=y^*}$
- Diagonalize: $Jm_i = \lambda_i m_i, \quad \text{where} \quad \begin{cases} m_i : \text{eigenvectors of } J \\ \lambda_i : \text{eigenvalues of } J \end{cases}$
- Define $q = M^{-1}\Delta y$:

$$\dot{q} = M^{-1}J\Delta y = M^{-1}JMq = \Lambda q, \qquad \Lambda = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_d \end{pmatrix}$$

- That is, $\dot{q}_i = \lambda_i q_i$ from which we can find $\Delta y(t) = Mq = \sum_{i=1}^{d} q_i(t) m_i$

We can study properties of a method used to simulate the system $\Delta\dot{y} = J\Delta y$, by study properties of the method for the systems $\dot{q}_i = \lambda_i q_i, \quad i = 1, \ldots, d$.

Example linearization

- System:

$$\dot{y}_1 = y_2$$
$$\dot{y}_2 = -y_1^3 - cy_2$$

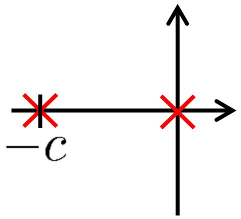
Linearization about $(y_1^*, y_2^*)^\mathsf{T}$:

$$\begin{pmatrix} \Delta\dot{y}_1 \\ \Delta\dot{y}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -3\left(y_1^*\right)^2 & -c \end{pmatrix} \begin{pmatrix} \Delta y_1 \\ \Delta y_2 \end{pmatrix}$$

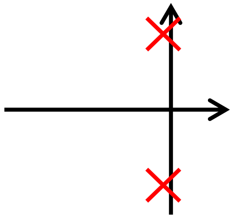
- Eigenvalues:

$$\lambda^2 + c\lambda + 3\left(y_1^*\right) = 0 \qquad \lambda_{1,2} = -\frac{c}{2} \pm \sqrt{\left(\frac{c}{2}\right)^2 - 3\left(y_1^*\right)^2}$$

$$y_1^* = 0: \quad \lambda_1 = 0,\ \lambda_2 = -c$$



$$y_1^* = \text{large}: \quad \lambda_{1,2} \to \pm j\omega_0$$



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:

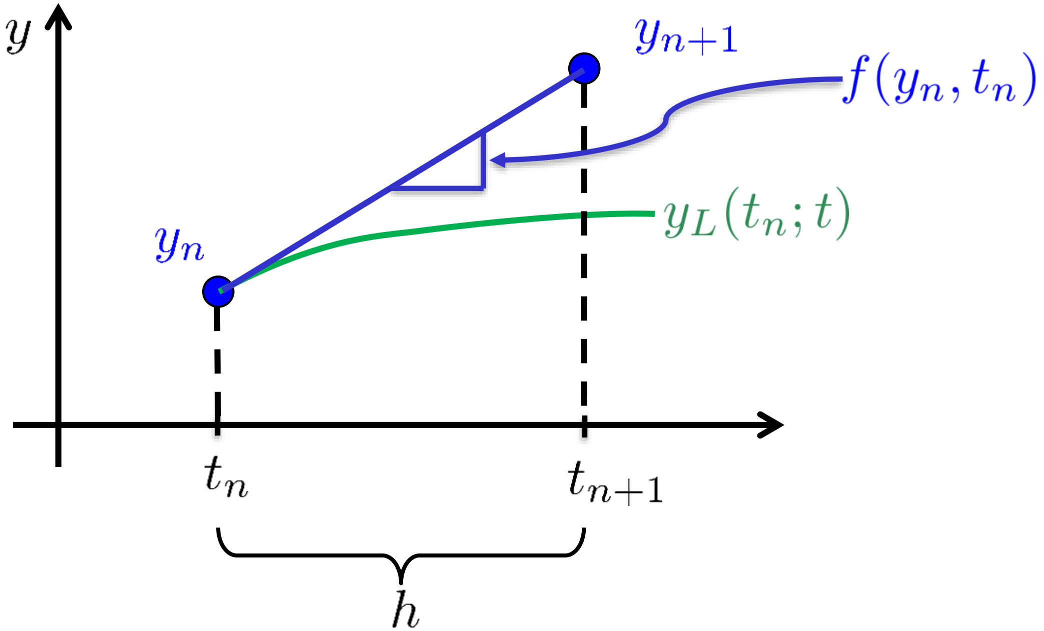
$$y_{n+1} = R(h\lambda)y_n$$

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

- The method is stable (for test system!) if $|y_{n+1}| \leq |y_n|$

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

Simplest method: Euler



- Slope:

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

- Euler's method: $y_{n+1} = y_n + hf(y_n, t) + \frac{h^2}{2}\frac{\mathrm{d}f(y_n, t)}{\mathrm{d}t} + \ldots + \frac{h^p}{p!}\frac{\mathrm{d}^{p-1}f(y_n, t)}{\mathrm{d}t^{p-1}} + O(h^{p+1})$

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

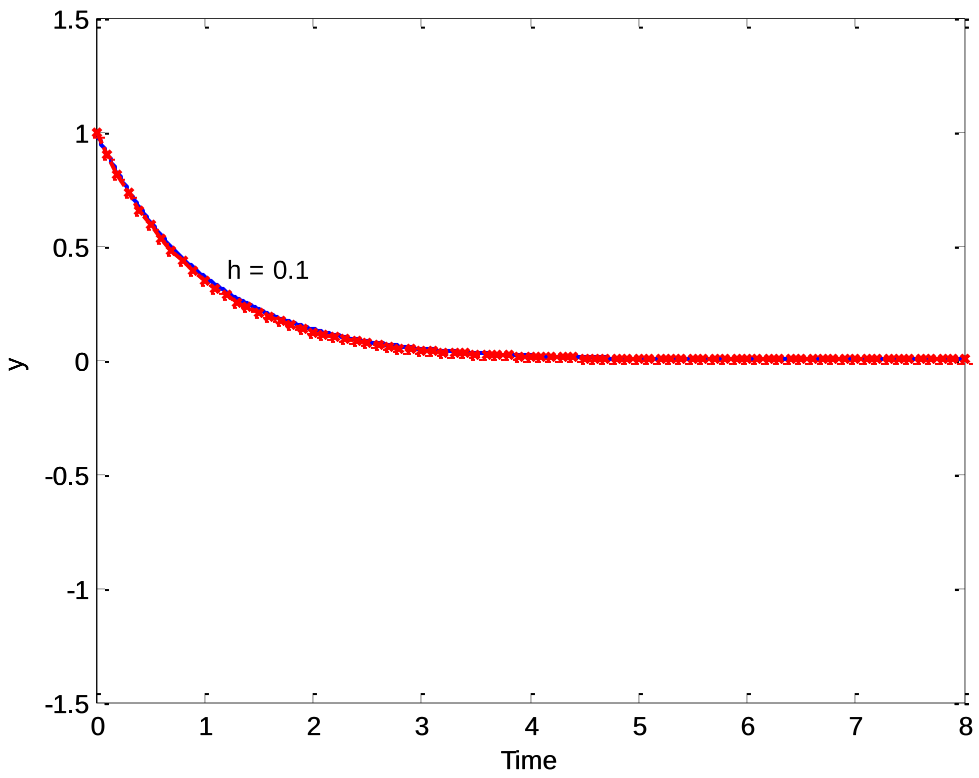
Example Euler's method

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	
…	…	…



Example: Euler's method stability

ODE:

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

Euler simulation:

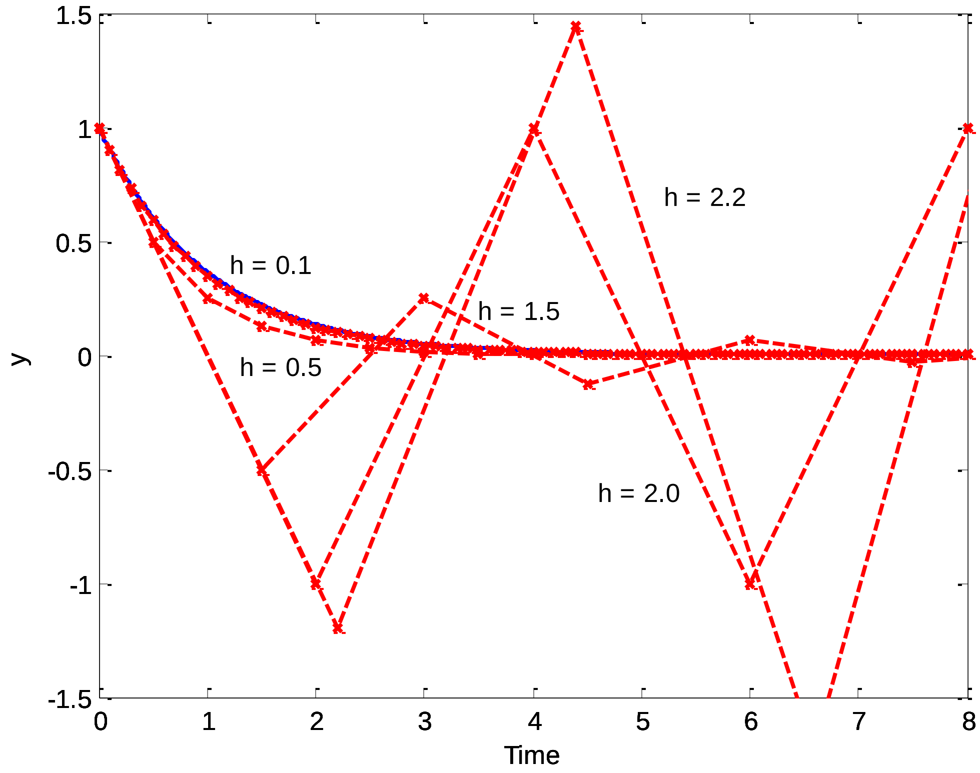
$$y_{n+1} = y_n + h\left(-y_n\right), \quad y_0 = 1$$

Example Euler's method

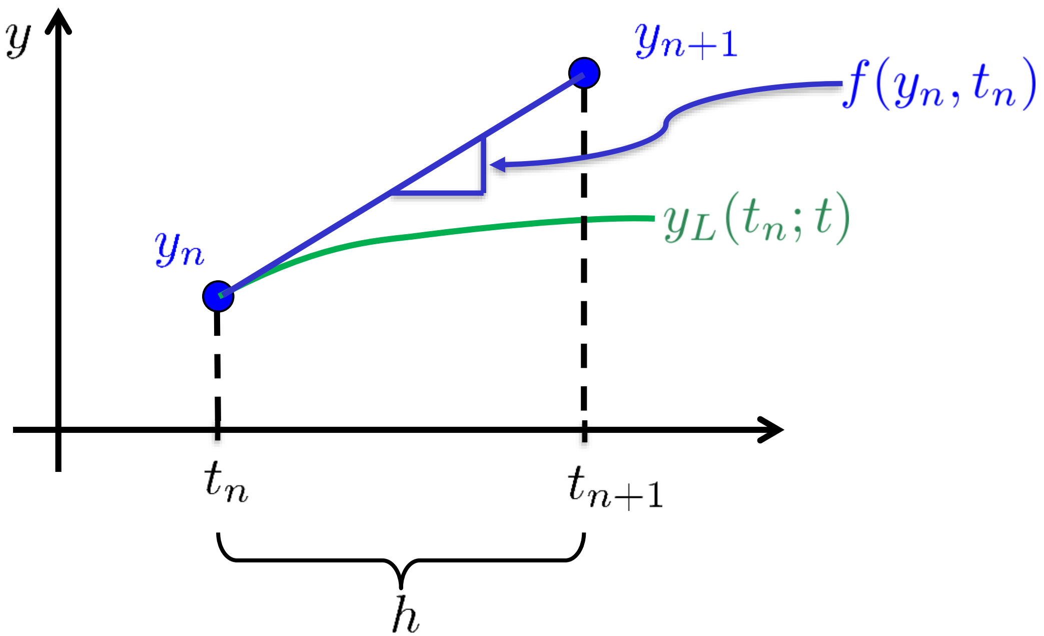
ODE: $$\dot{y} = -y, \quad y(0) = 1$$

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

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



Simplest method: Euler



- Slope:

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

- Euler's method:

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

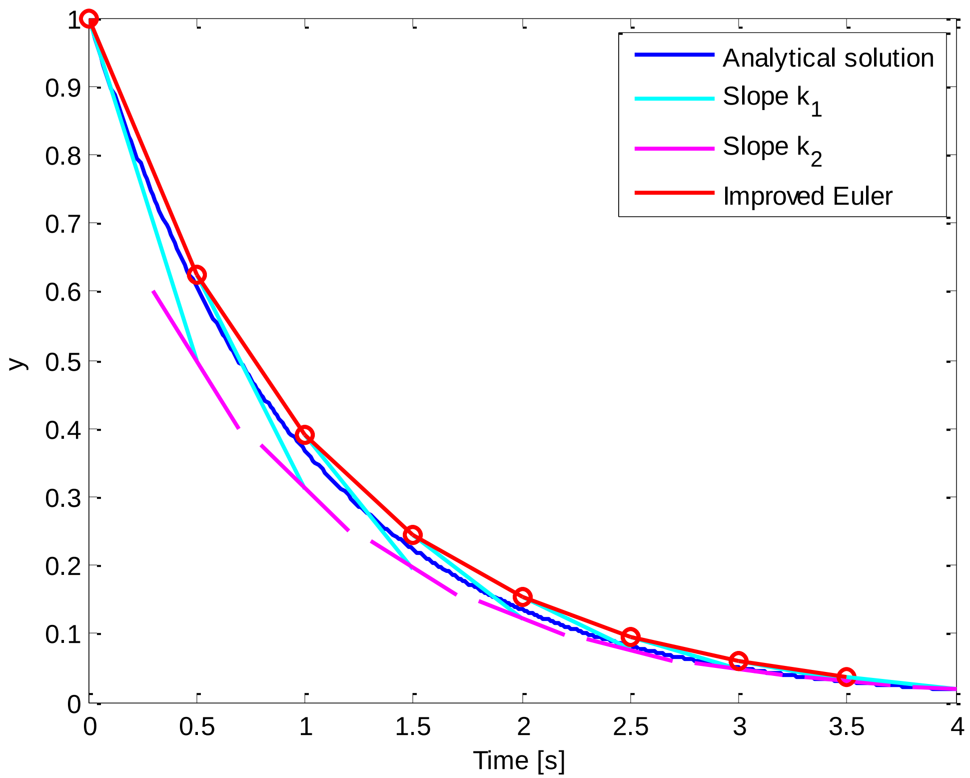
Euler method

Improved Euler illustration

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

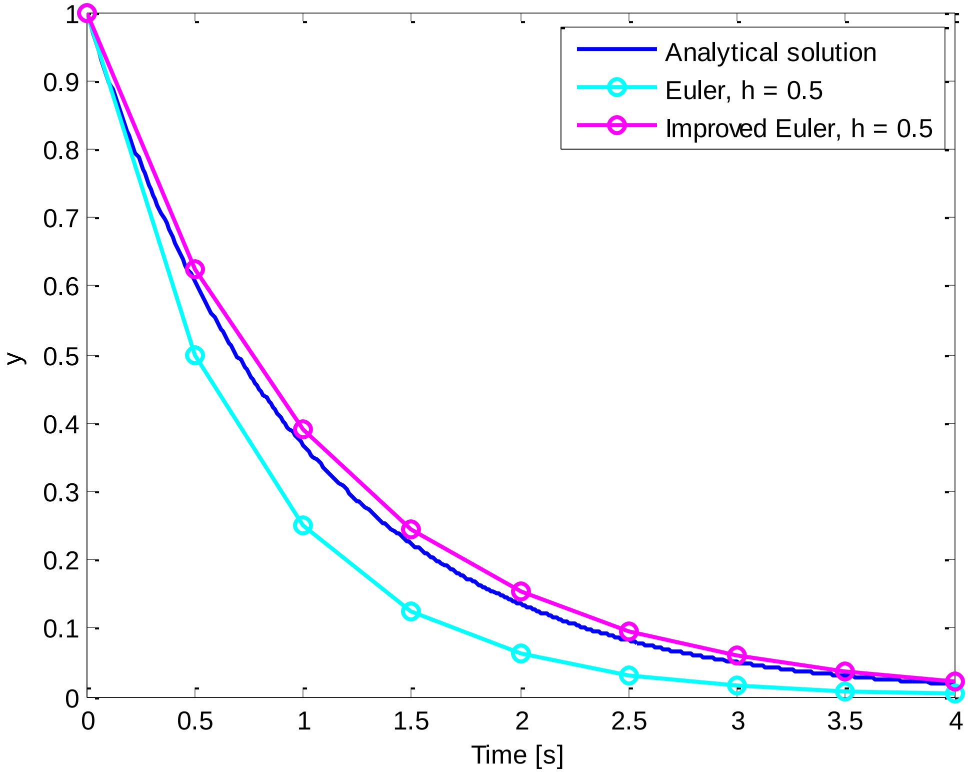
Improved Euler: $k_1 = f(y_n), \;\; k_2 = f(y_n + hk_1)$

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



Improved Euler vs Euler

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



Order of improved Euler method

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

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

- Taylor series expansion of k_2:

$$k_2 = f(y_n, t_n) + h\frac{df(y_n, t_n)}{dt} + \frac{h^2}{2}\frac{d^2 f(y_n, t_n)}{dt^2} + O(h^3)$$

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

Order of improved Euler method

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

Stability of improved Euler method

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

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

- Test system: $\dot{y} = \lambda y$

Accuracy and stability

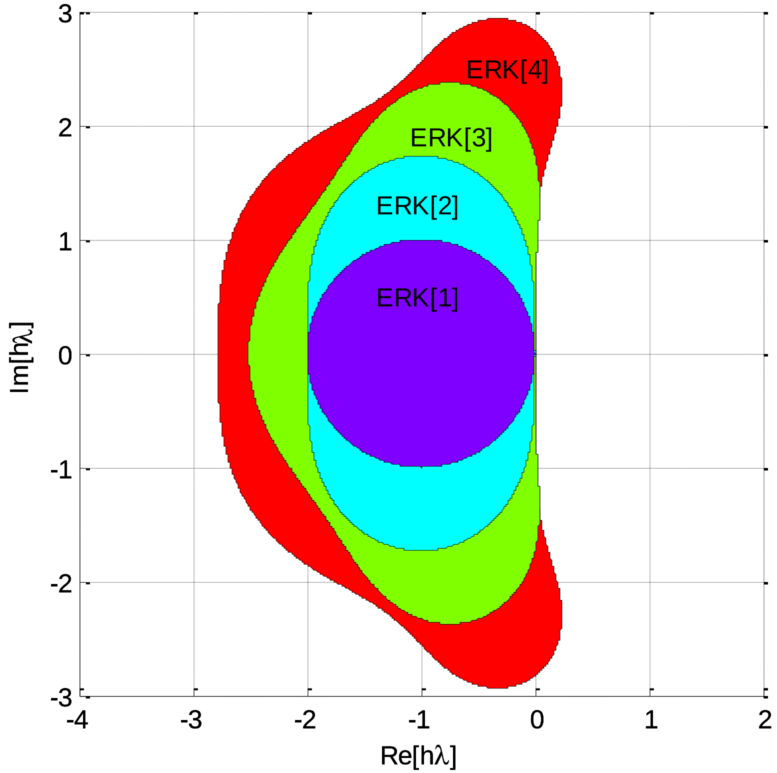
- Lots of different methods, with different complexity. How to quantify their behaviour?

- 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!

Stability regions for ERK methods



Explicit Runge-Kutta method I

Explicit Runge-Kutta method II

Butcher array

Butcher array: Examples

1. Explicit Euler:

$$\begin{aligned} k_1 &= f(y_n, t_n) \\ y_{n+1} &= y_n + hk_1 \end{aligned}$$

2. Improved Euler:

$$\begin{aligned} k_1 &= f(y_n, t_n) \\ k_2 &= f(y_n + hk_1, t_n + h) \\ y_{n+1} &= y_n + {}^h\!/_2(k_1 + k_2) \end{aligned}$$

3. Heun's method:

$$\begin{aligned} k_1 &= f(y_n, t_n) \\ k_2 &= f(y_n + {}^1\!/_3 hk_1, t_n + {}^1\!/_3 h) \\ k_3 &= f(y_n + {}^2\!/_3 hk_2, t_n + {}^2\!/_3 h) \\ y_{n+1} &= y_n + {}^1\!/_4 hk_1 + {}^3\!/_4 hk_3 \end{aligned}$$

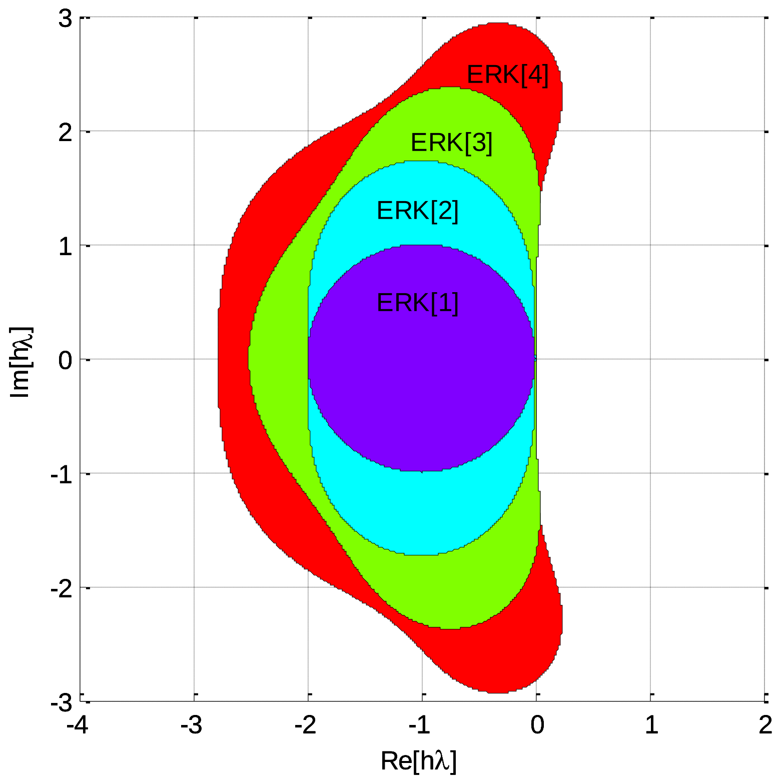
Butch array: Try yourself

- Write down the equations of the method!

- ERK 4:

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

Stability regions for ERK methods



Stability of ERK I

Stability of ERK II

Stability of ERK III

$$(I - h\lambda A)\kappa = \lambda \mathbf{1} y_n$$
$$y_{n+1} = y_n + hb^T\kappa$$

Stability of ERK IV

Homework

- Write the Butcher array for the improved Euler method (Slide 24)
- Write down the equations of the ERK4 method on slide 25.

- For Monday: Read 14.12 (only until 14.12.1)

- For Thursday: Read 14.5

Next lecture …

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 $\sigma \leq 4$:

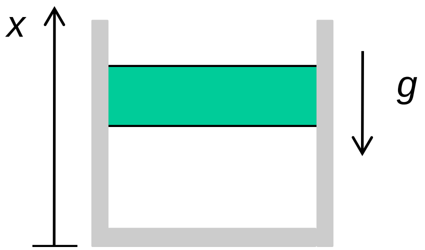
$$R_E(h\lambda) = 1 + h\lambda + \ldots + \frac{h^p\lambda^p}{p!}$$

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

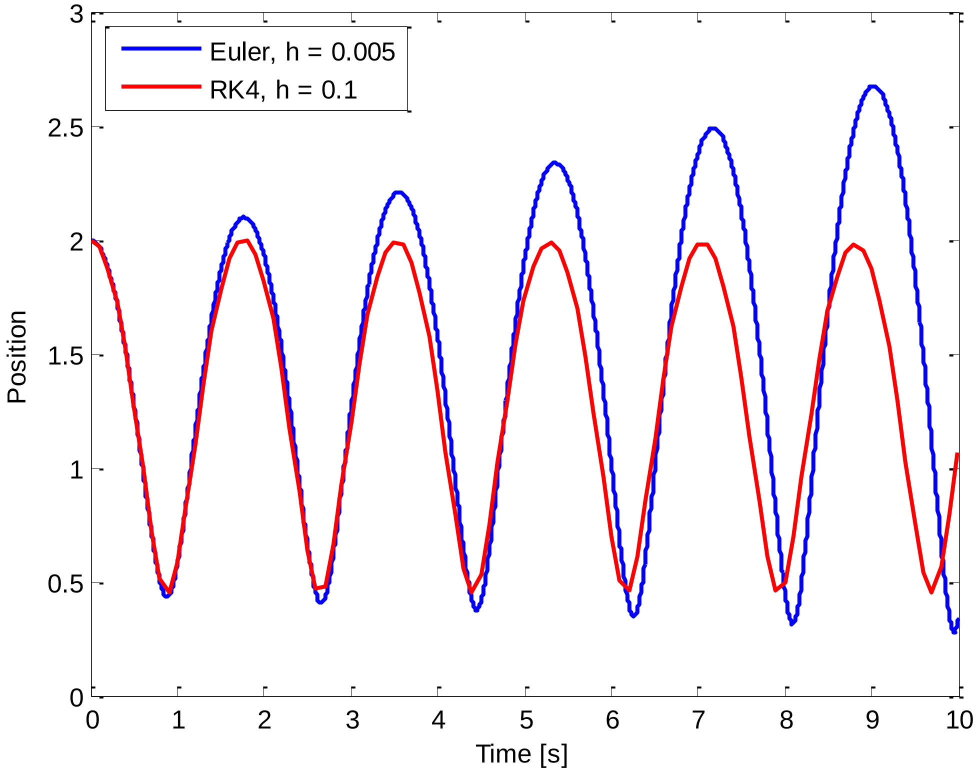
$$\begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \end{pmatrix} = \begin{pmatrix} y_2 \\ -g(1 - y_1^{-\kappa}) \end{pmatrix}$$

- Linearization about equilibrium ($y_1 = 1$) :

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

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