Lecture 9: Stability, Padé approximations

- Stability of Runga-Kutta methods
  - **A-stability**
  - Aliasing
  - **L-stability**
  - Padé approximations
  - (Nonlinear stability: AN-, B- and algebraic stability)

Book: 14.6

Explicit Runge-Kutta (ERK) methods

- IVP: $\dot{y} = f(y,t), \quad y(0) = y_0$
- ERK:

$$
\begin{aligned}
k_1 &= f(y_n, t_n) \\
k_2 &= f(y_n + h a_{21} k_1, t_n + c_2 h) \\
k_3 &= f(y_n + h(a_{31} k_1 + a_{32} k_2), t_n + c_3 h) \\
&\vdots \\
k_\sigma &= f(y_n + h(a_{\sigma,1} k_1 + a_{\sigma,2} k_2 + \ldots + a_{\sigma,\sigma-1} k_{\sigma-1}), t_n + c_\sigma h) \\
y_{n+1} &= y_n + h(b_1 k_1 + b_2 k_2 + \ldots + b_\sigma k_\sigma)
\end{aligned}
$$

- Butcher array:

$\mathbf{c}$	$\mathbf{A}$
	$\mathbf{b}^\mathsf{T}$

0					
c_2	a_{21}				
c_3	a_{31}	a_{32}			
$\vdots$	$\vdots$	$\vdots$	$\ddots$		
c_σ	$a_{\sigma,1}$	$a_{\sigma,2}$	$\cdots$	$a_{\sigma,\sigma-1}$	
	b_1	b_2	$\ldots$	$b_{\sigma-1}$	b_σ

Recap: Test system, stability function

- One step method (typically: Runge-Kutta):

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

- Apply it to scalar test system:

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

- We get:

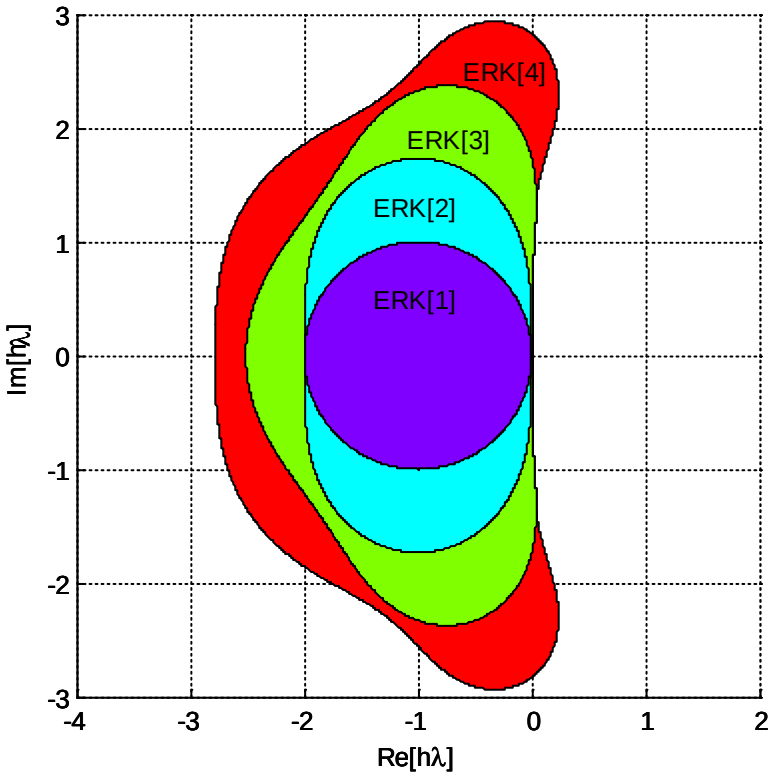
$$y_{n+1} = R(h\lambda)y_n$$

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

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

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

Stability regions for ERK methods



Implicit Runge-Kutta (IRK) methods

- IVP: $\dot{y} = f(y,t), \quad y(0) = y_0$
- IRK:

$$
\begin{aligned}
k_1 &= f(y_n + h(a_{1,1}k_1 + a_{1,2}k_2 + \ldots + a_{1,\sigma}k_\sigma), t_n + c_1 h) \\
k_2 &= f(y_n + h(a_{2,1}k_1 + a_{2,2}k_2 + \ldots + a_{2,\sigma}k_\sigma), t_n + c_2 h) \\
k_3 &= f(y_n + h(a_{3,1}k_1 + a_{3,2}k_2 + \ldots + a_{3,\sigma}k_\sigma), t_n + c_3 h) \\
&\vdots \\
k_\sigma &= f(y_n + h(a_{\sigma,1}k_1 + a_{\sigma,2}k_2 + \ldots + a_{\sigma,\sigma}k_\sigma), t_n + c_\sigma h) \\
y_{n+1} &= y_n + h(b_1 k_1 + b_2 k_2 + \ldots + b_\sigma k_\sigma)
\end{aligned}
$$

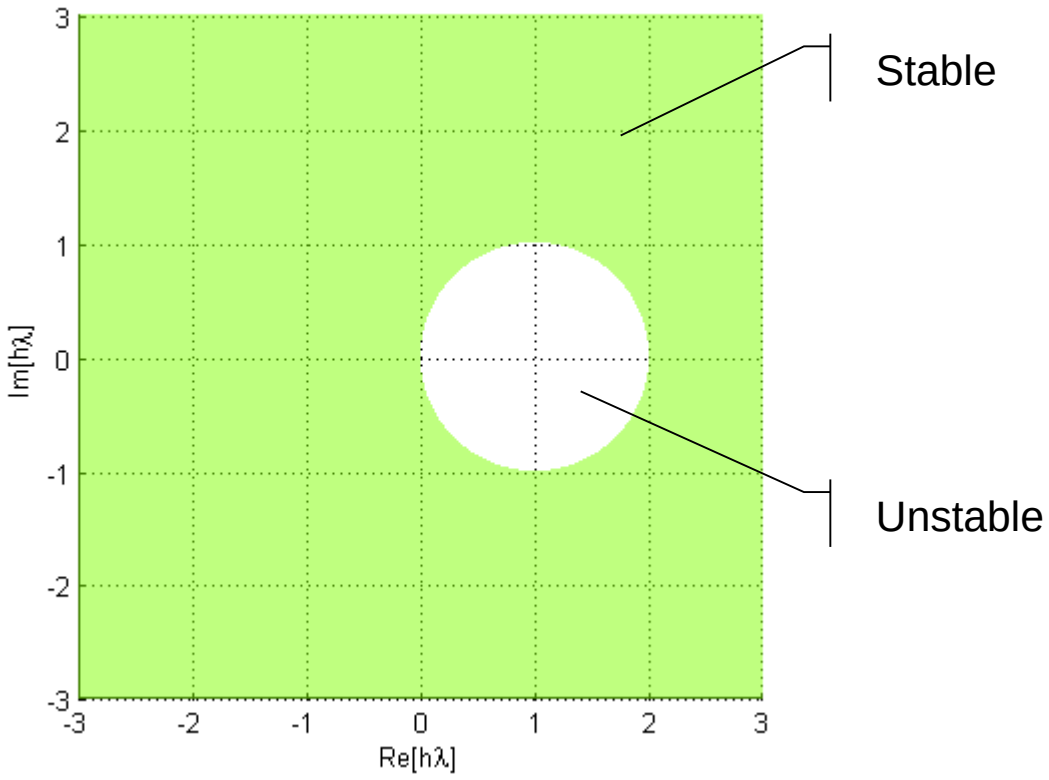
- Butcher array:

$\mathbf{c}$	$\mathbf{A}$
	$\mathbf{b}^\mathsf{T}$

c_1	a_{11}	a_{12}	$\cdots$	$a_{1,\sigma-1}$	$a_{1,\sigma}$
c_2	a_{21}	a_{22}	$\cdots$	$a_{2,\sigma-1}$	$a_{2,\sigma}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
c_σ	$a_{\sigma,1}$	$a_{\sigma,2}$	$\cdots$	$a_{\sigma,\sigma-1}$	$a_{\sigma,\sigma}$
	b_1	b_2	$\ldots$	$b_{\sigma-1}$	b_σ

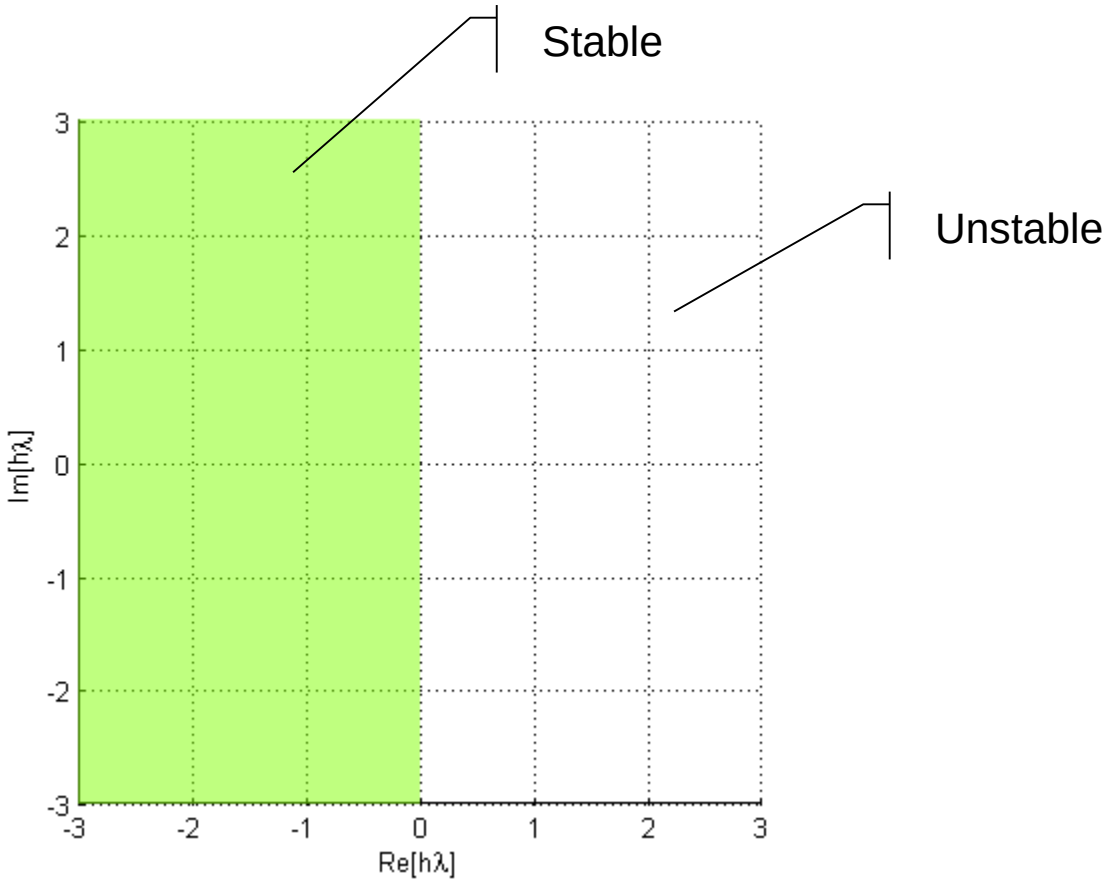
Stability regions for implicit methods

- Implicit Euler



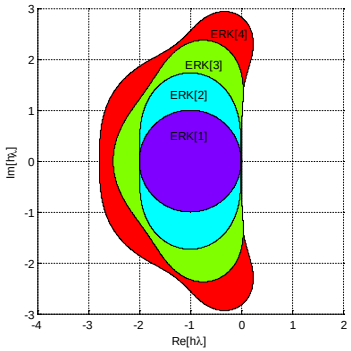
Stability regions for implicit methods

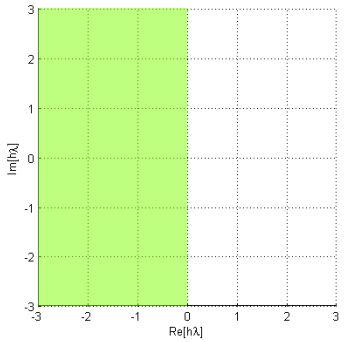
- Trapezoidal rule and implicit midpoint rule:



Why use IRK methods?

- IRK methods are much more complex (since we have to solve a set of nonlinear equations for each step) than ERK methods, so why use them?
  - Accuracy? Stability?
- Not because of accuracy
  - Even if an IRK method may have higher accuracy for a given number of stages, it is easy and cheap to achieve same accuracy for ERK by increasing the number of stages
- It's because of the much larger stability region!
- When is this important?
  - **Stiff systems!**





Example: ”Lambert’s problem”

- IVP:

$$\dot{u} = \frac{1}{100} - (\frac{1}{100} + u + v)(1 + (u + 1000)(u + 1)), \quad u(0) = 0$$
$$\dot{v} = \frac{1}{100} - (\frac{1}{100} + u + v)(1 + v^2), \quad v(0) = 0$$

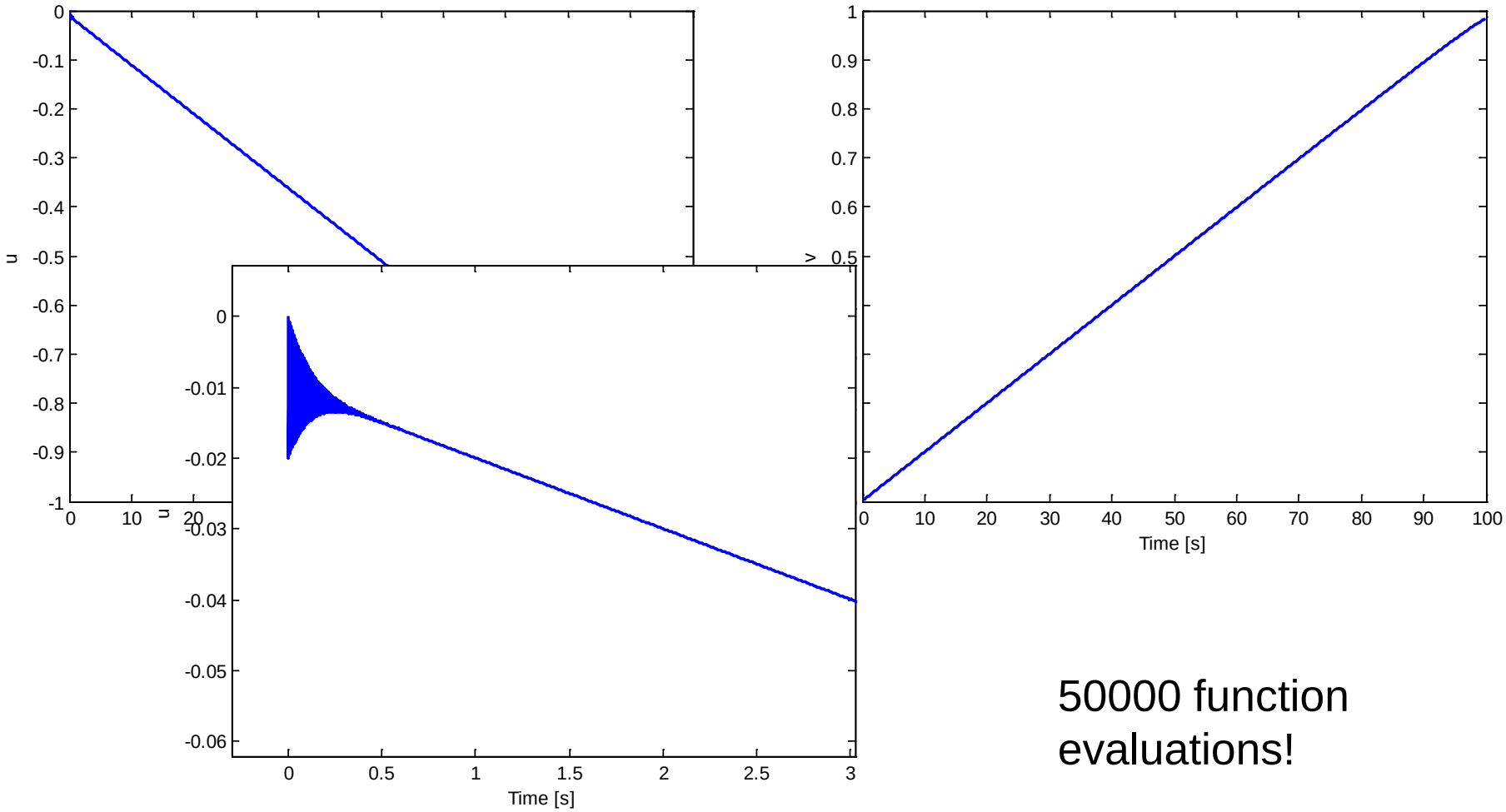
- Task: Simulate from t = 0 s til t = 100 s

- Eigenvalues:

$$(u, v) = (0, 0) \Rightarrow \lambda_1 \approx -1000, \lambda_2 \approx -0.01$$
$$(u, v) = (-.5, .5) \Rightarrow \lambda_1 \approx -500, \lambda_2 \approx -0.03$$
$$(u, v) = (-1, 1) \Rightarrow \lambda_1 \approx -11, \lambda_2 \approx -1$$

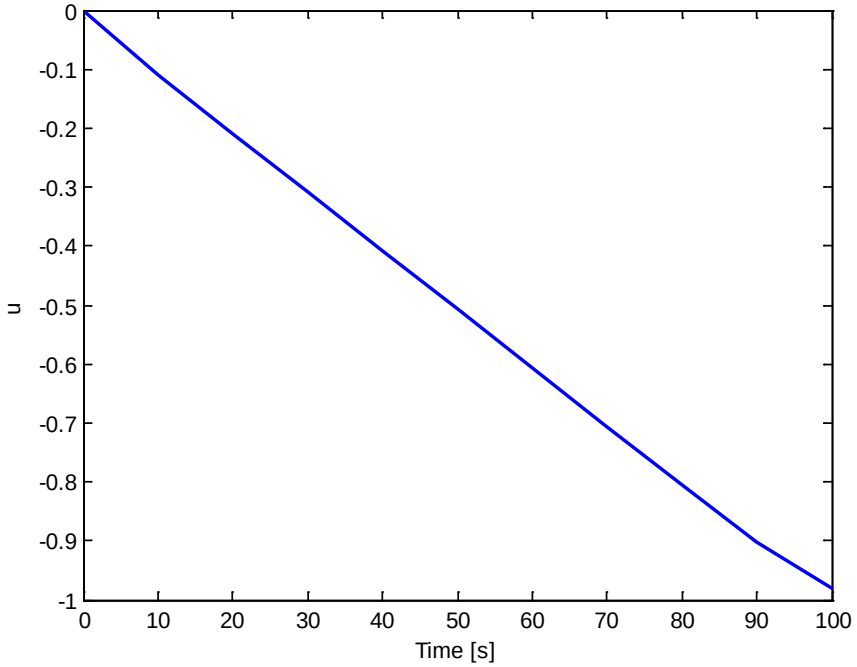
Using Euler (explicit), h = 0.002

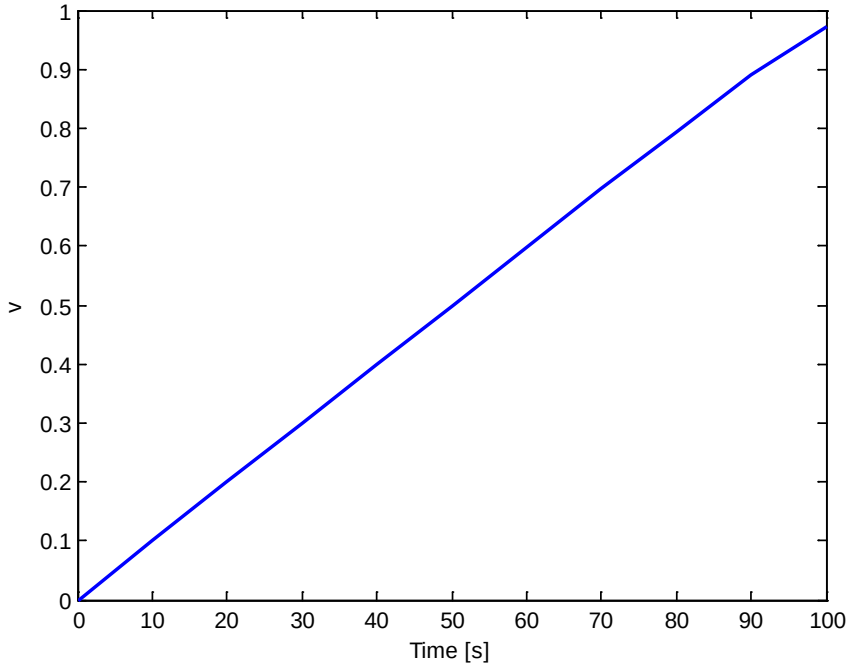
$$h|\lambda| \leq 2$$



50000 function evaluations!

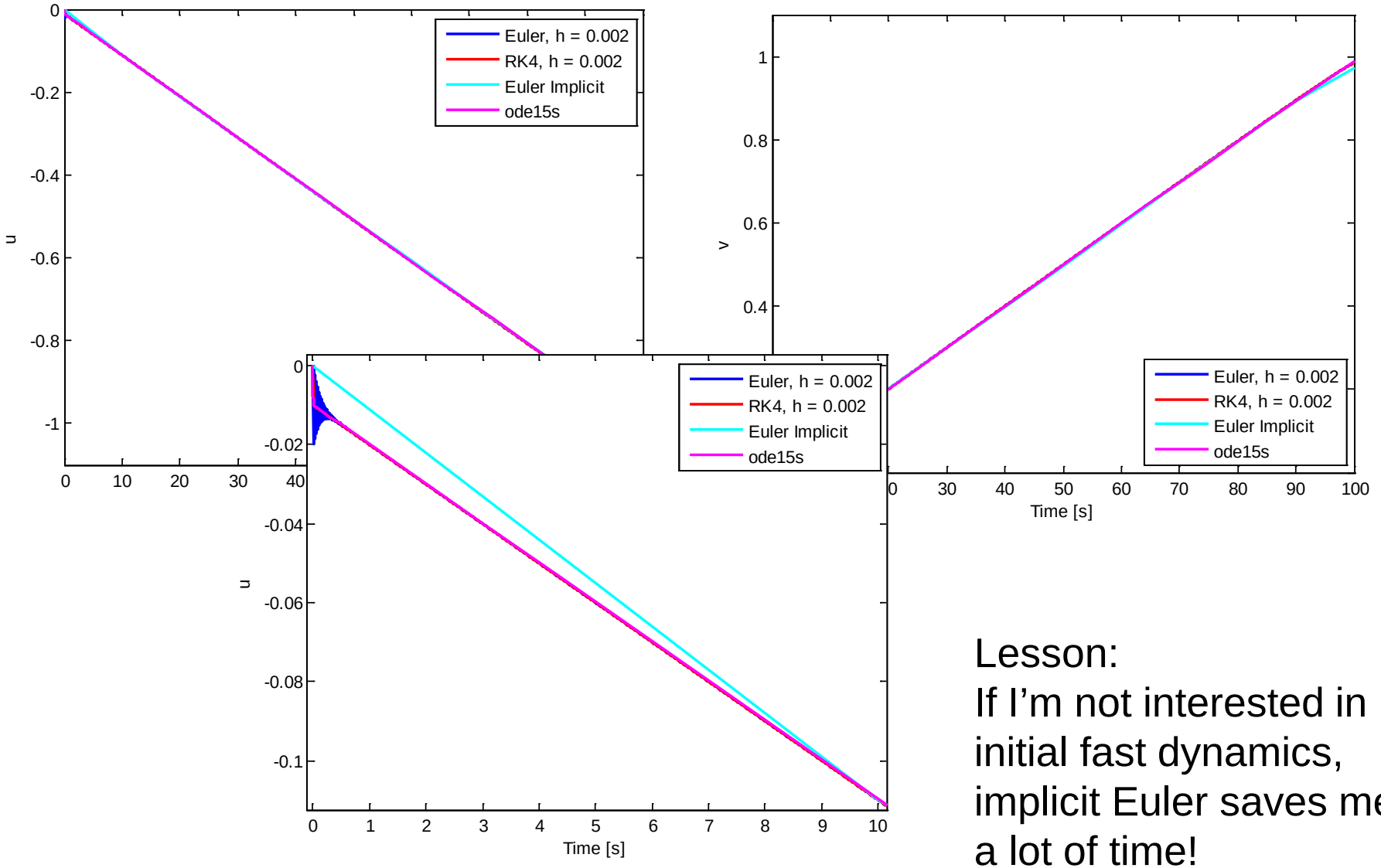
Attempt 3: Euler implicit, h = 10





149 function evaluations!
(dependent on solution algorithm)

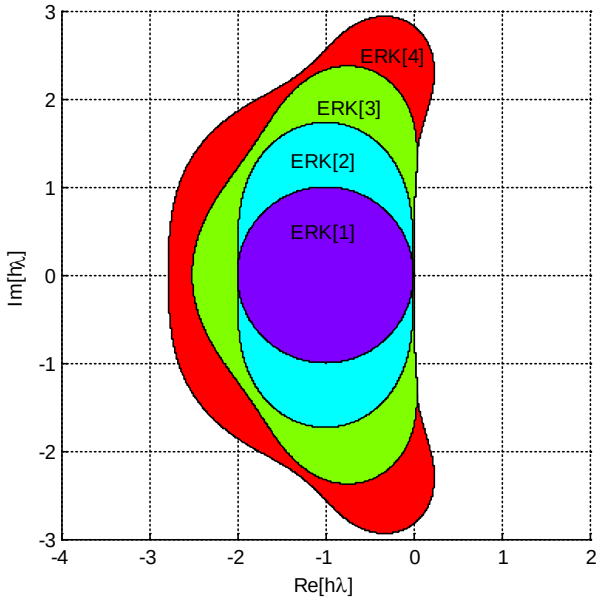
Comparisons



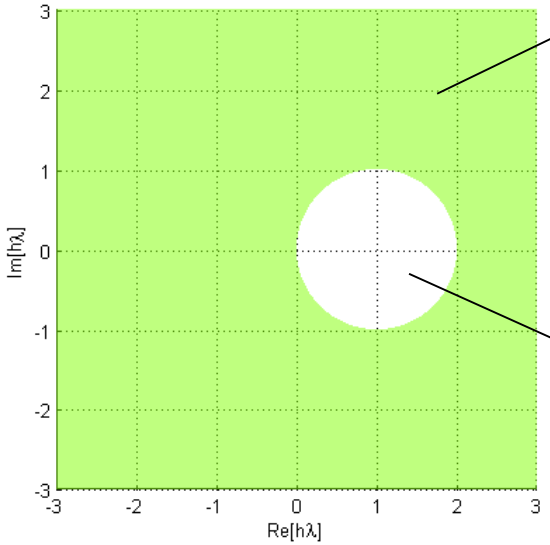
Lesson:
If I'm not interested in initial fast dynamics, implicit Euler saves me a lot of time!

ERK vs IRK stability regions

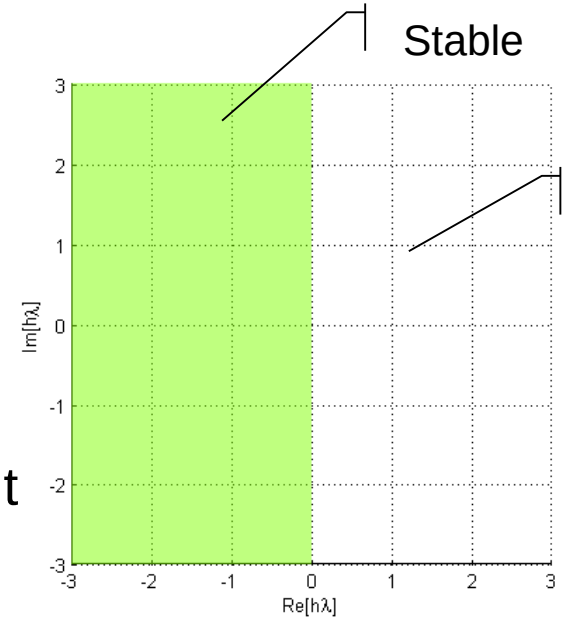
ERK-methods



Implicit Euler



Trapezoidal/
Implicit Midpoint



Padé approximations to e^s

m \ k	0	1	2	3
0	$\frac{1}{1}$	$\frac{1+s}{1}$	$\frac{1+s+\frac{1}{2}s^2}{1}$	$\frac{1+s+\frac{1}{2}s^2+\frac{1}{6}s^3}{1}$
1	$\frac{1}{1-s}$	$\frac{1+\frac{1}{2}s}{1-\frac{1}{2}s}$	$\frac{1+\frac{2}{3}s+\frac{1}{6}s^2}{1-\frac{1}{3}s}$	$\frac{1+\frac{3}{4}s+\frac{1}{4}s^2+\frac{1}{24}s^3}{1-\frac{1}{4}s}$
2	$\frac{1}{1-s+\frac{1}{2}s^2}$	$\frac{1+\frac{1}{3}s}{1-\frac{2}{3}s+\frac{1}{6}s^2}$	$\frac{1+\frac{1}{2}s+\frac{1}{12}s^2}{1-\frac{1}{2}s+\frac{1}{12}s^2}$	$\frac{1+\frac{3}{5}s+\frac{3}{20}s^2+\frac{1}{60}s^3}{1-\frac{2}{5}s+\frac{1}{20}s^2}$
3	$\frac{1}{1-s+\frac{1}{2}s^2-\frac{1}{6}s^3}$	$\frac{1+\frac{1}{4}s}{1-\frac{3}{4}s+\frac{1}{4}s^2-\frac{1}{24}s^3}$	$\frac{1+\frac{2}{5}s+\frac{1}{20}s^2}{1-\frac{3}{5}s+\frac{3}{20}s^2-\frac{1}{60}s^3}$	$\frac{1+\frac{1}{2}s+\frac{1}{10}s^2+\frac{1}{120}s^3}{1-\frac{1}{2}s+\frac{1}{10}s^2-\frac{1}{120}s^3}$

- $m = 0$: Explicit Runge-Kutta methods with $p = \sigma$
- $m = k$: Gauss, Lobatto IIIA/IIIB (incl. implicit mid-point, trapezoidal)
- $m = k+1$: Radau-methods (incl. implicit Euler)
- $m = k+2$: Lobatto IIIC

Padé approximations to e^s

m \ k	0	1	2	3
0	$\frac{1}{1}$	$\frac{1+s}{1}$	$\frac{1+s+\frac{1}{2}s^2}{1}$	$\frac{1+s+\frac{1}{2}s^2+\frac{1}{6}s^3}{1}$
1	$\frac{1}{1-s}$	$\frac{1+\frac{1}{2}s}{1-\frac{1}{2}s}$	$\frac{1+\frac{2}{3}s+\frac{1}{6}s^2}{1-\frac{1}{3}s}$	$\frac{1+\frac{3}{4}s+\frac{1}{4}s^2+\frac{1}{24}s^3}{1-\frac{1}{4}s}$
2	$\frac{1}{1-s+\frac{1}{2}s^2}$	$\frac{1+\frac{1}{3}s}{1-\frac{2}{3}s+\frac{1}{6}s^2}$	$\frac{1+\frac{1}{2}s+\frac{1}{12}s^2}{1-\frac{1}{2}s+\frac{1}{12}s^2}$	$\frac{1+\frac{3}{5}s+\frac{3}{20}s^2+\frac{1}{60}s^3}{1-\frac{2}{5}s+\frac{1}{20}s^2}$
3	$\frac{1}{1-s+\frac{1}{2}s^2-\frac{1}{6}s^3}$	$\frac{1+\frac{1}{4}s}{1-\frac{3}{4}s+\frac{1}{4}s^2-\frac{1}{24}s^3}$	$\frac{1+\frac{2}{5}s+\frac{1}{20}s^2}{1-\frac{3}{5}s+\frac{3}{20}s^2-\frac{1}{60}s^3}$	$\frac{1+\frac{1}{2}s+\frac{1}{10}s^2+\frac{1}{120}s^3}{1-\frac{1}{2}s+\frac{1}{10}s^2-\frac{1}{120}s^3}$

L-stable L-stable A-stable

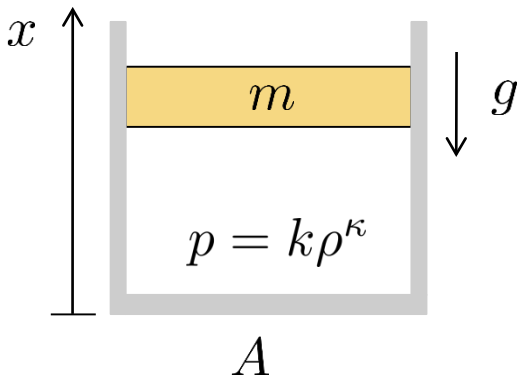
- $m = 0$: Explicit Runge-Kutta methods with $p = \sigma$
- $m = k$: Gauss, Lobatto IIIA/IIIB (incl. implicit mid-point, trapezoidal)
- $m = k+1$: Radau-methods (incl. implicit Euler)
- $m = k+2$: Lobatto IIIC

Pneumatic spring example, again

- Model from Newton's 2nd law:

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

"mass-spring-damper with nonlinear spring"



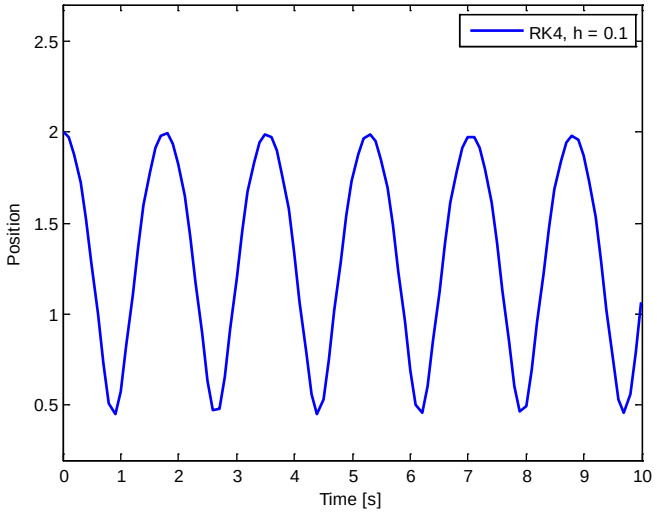
- On state-space form $\dot{y} = f(y, t)$

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

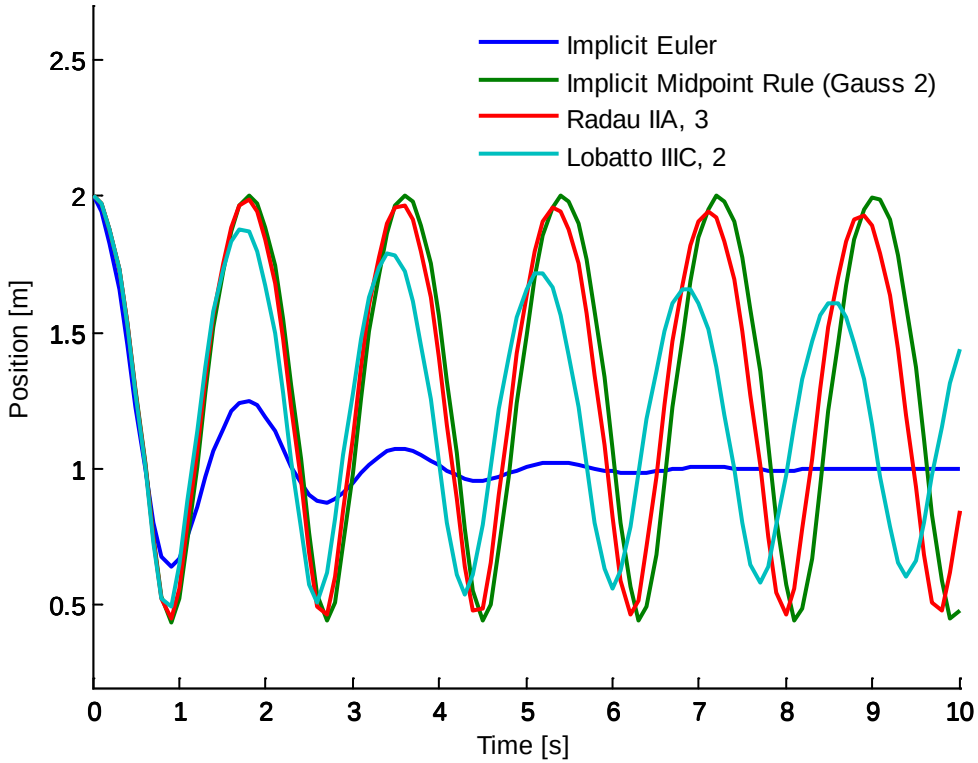
- Linearization about equilibrium:

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

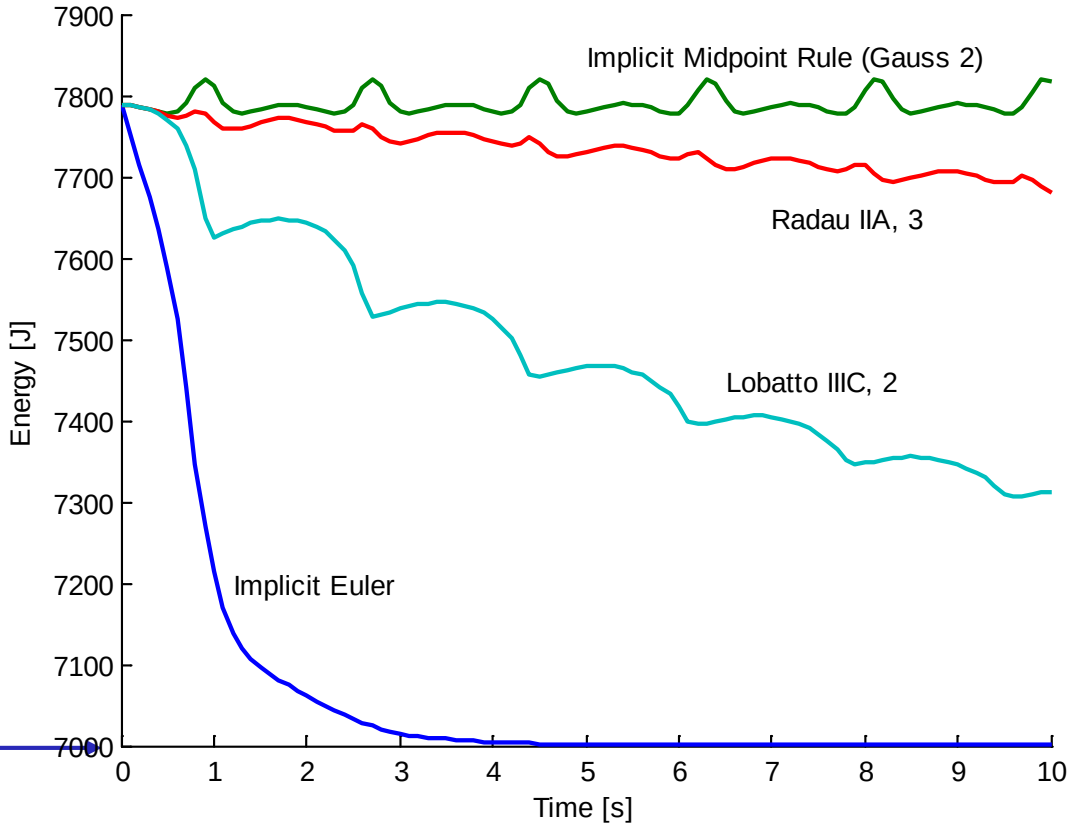
Simulation



h = 0.5 (stability limit for RK4)



Energy



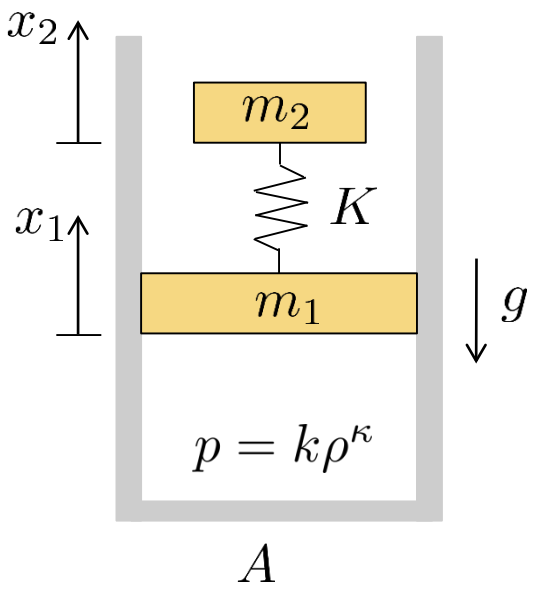
Equilibrium energy

Pneumatic spring with resonant load

- Equations of motion (Newton's law):

$$\ddot{x}_1 + g\left(1 - \frac{m_1 + m_2}{m_1} x_1^{-\kappa}\right) + \frac{\omega_2^2}{2}(x_1 - x_2) = 0$$

$$\ddot{x}_2 + g + \frac{\omega_2^2}{2}(x_2 - x_1) = 0$$



- Linearization around equilibrium:

$$J = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -g\frac{m_1+m_2}{m_1}\kappa\left(x_1^*\right)^{-(\kappa-1)} - \frac{\omega_2^2}{2} & 0 & \frac{\omega_2^2}{2} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{\omega_2^2}{2} & 0 & -\frac{\omega_2^2}{2} & 0 \end{pmatrix}$$

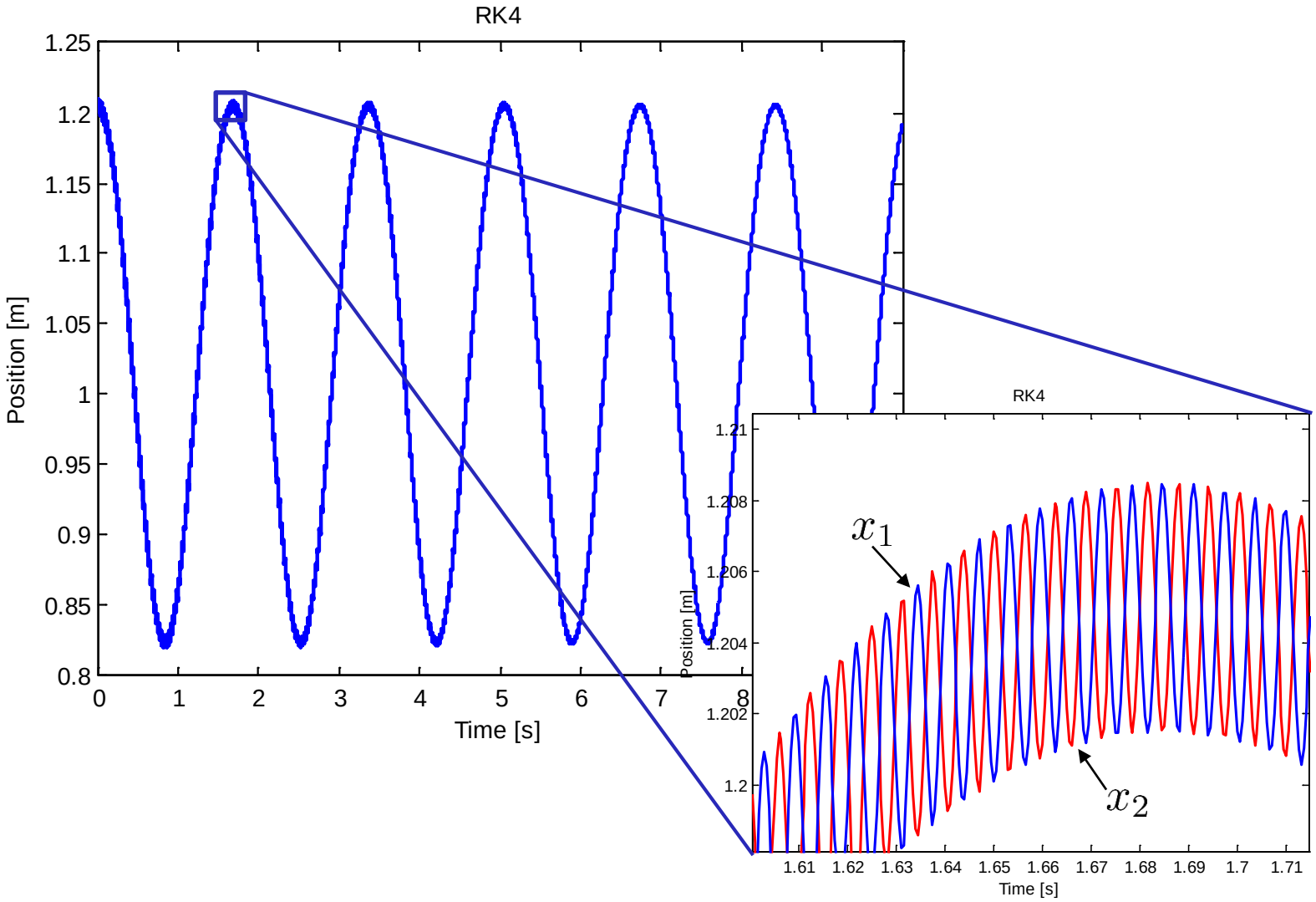
- Eigenvalues:

$$\lambda_{1,2} = \pm j\omega_1, \quad \omega_1 = 3.7 \text{ rad/s}$$

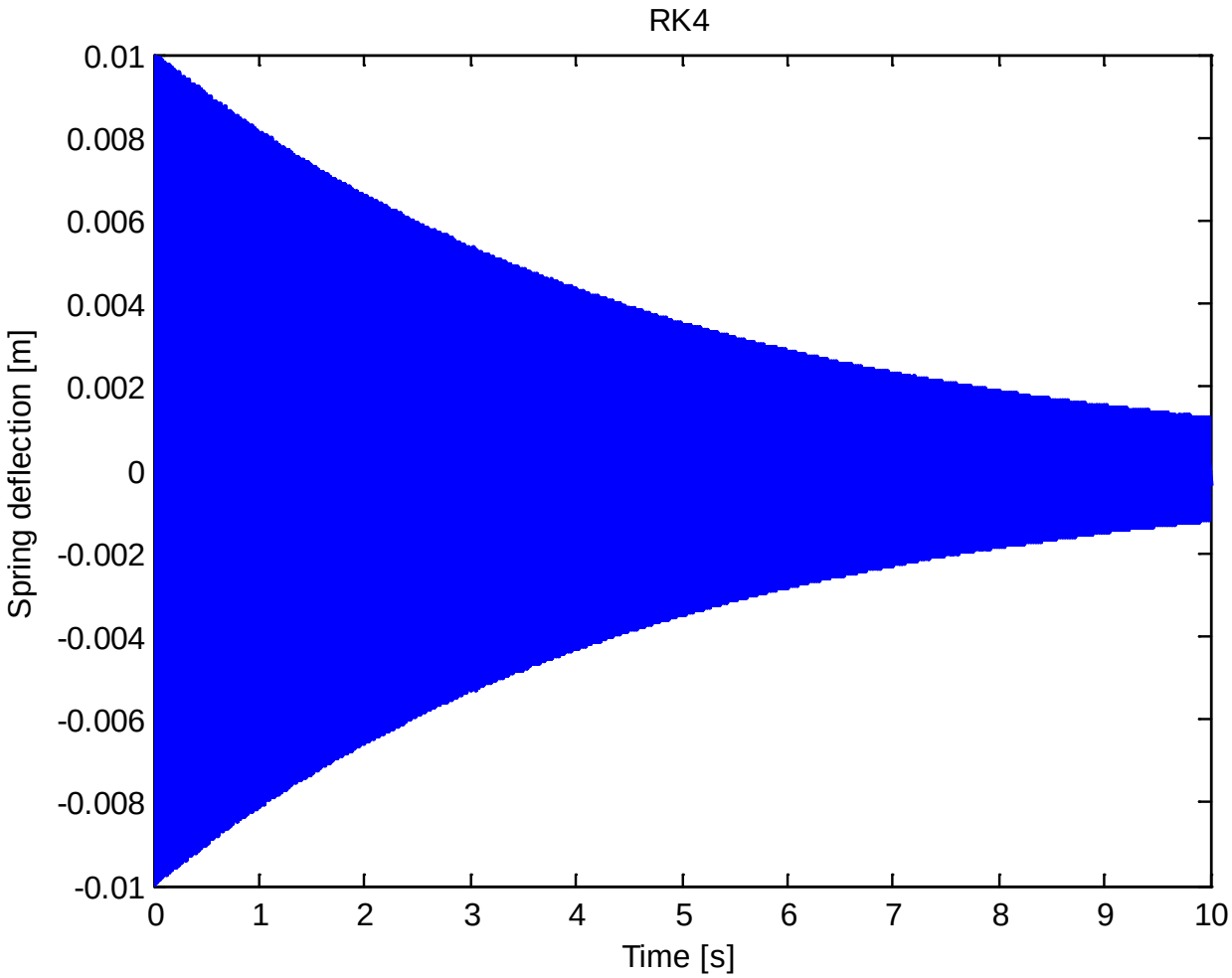
$$\lambda_{3,4} = \pm j\omega_2, \quad \omega_2 = 1000 \text{ rad/s}$$

Position of the two masses

RK4 with time step h = 0.0005

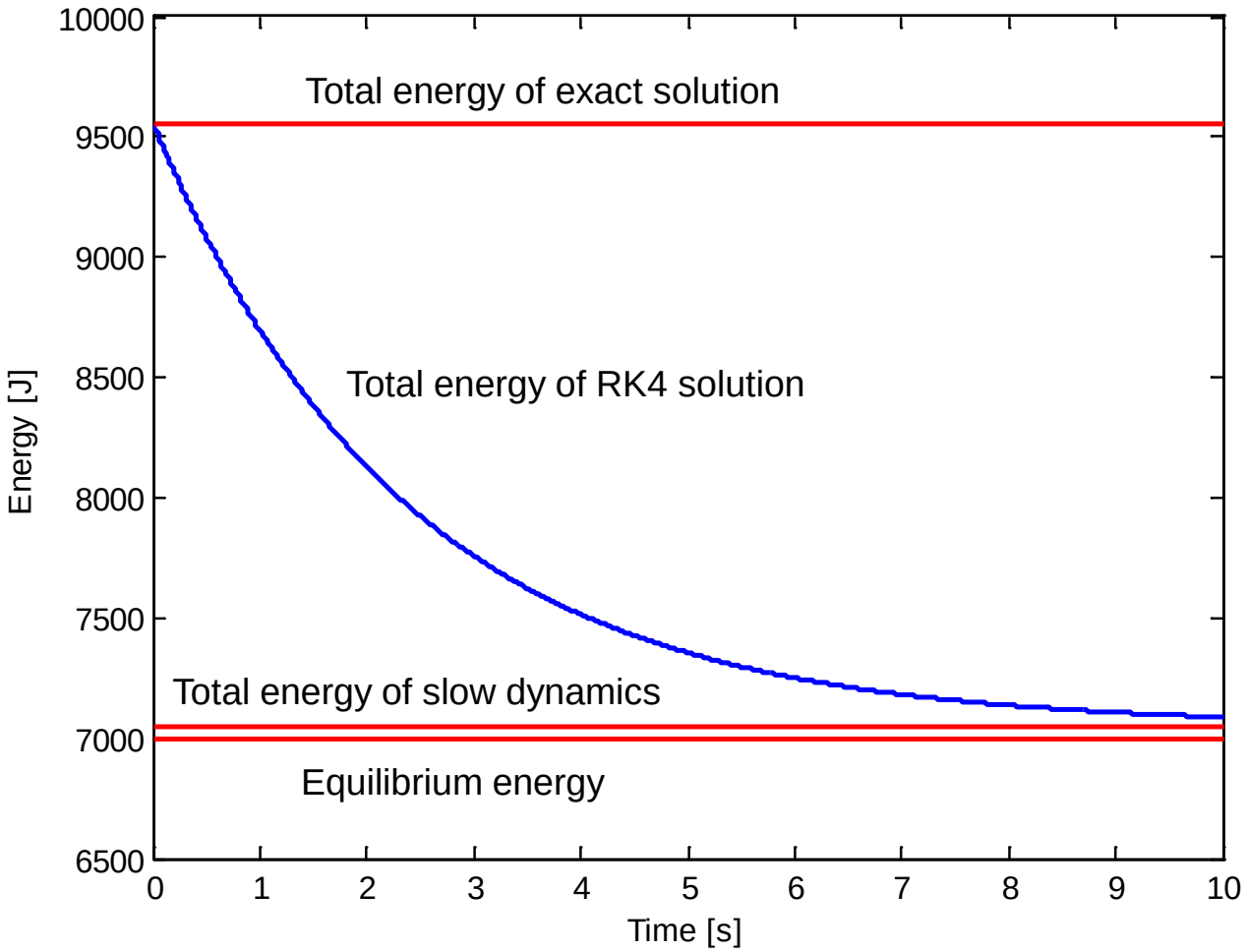


Spring deflection, RK4 with h = 0.0005



- Oscillation is lightly damped by integration method

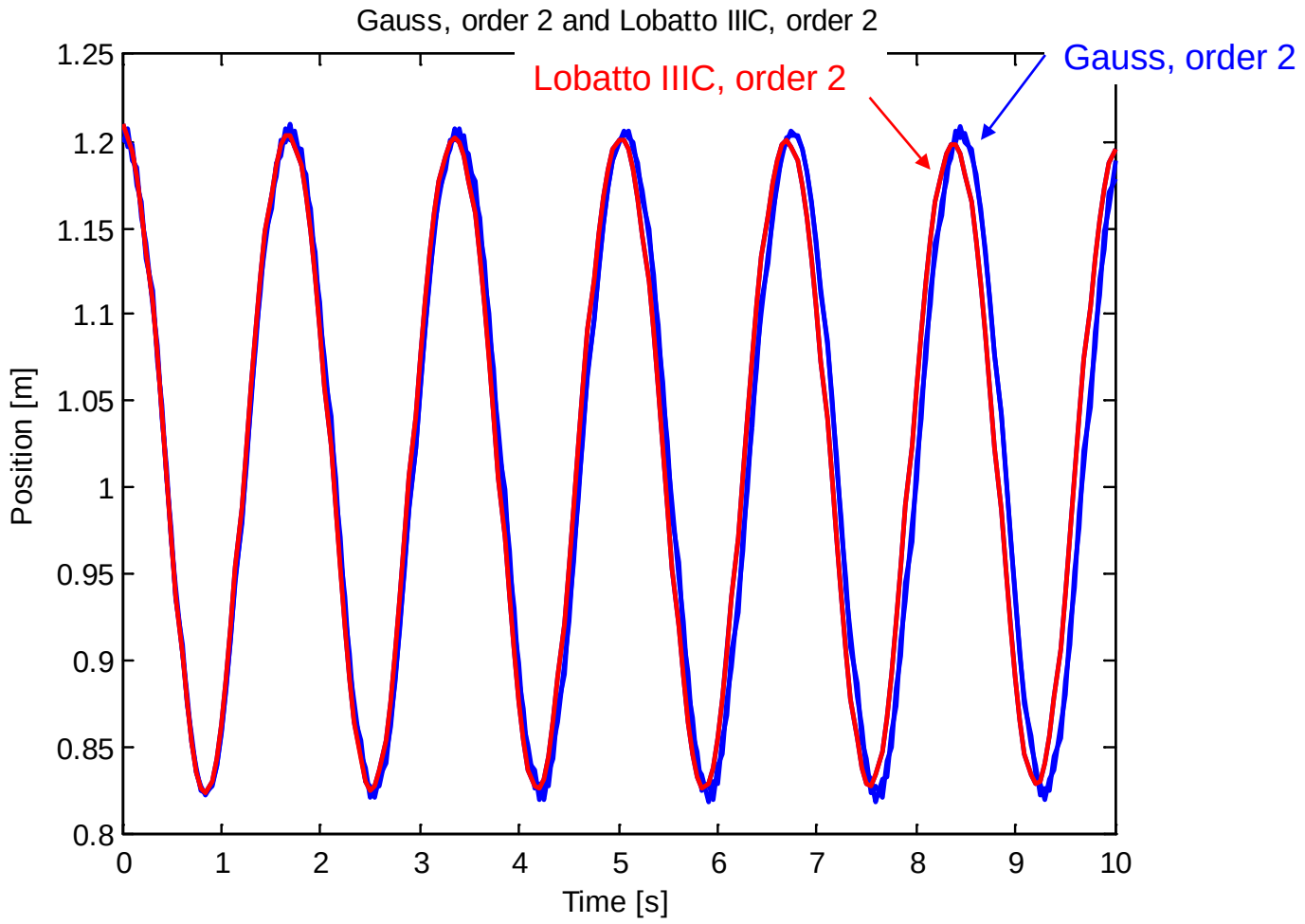
Energy of RK4 solution, h = 0.0005



- Energy related to fast dynamics slowly damped out

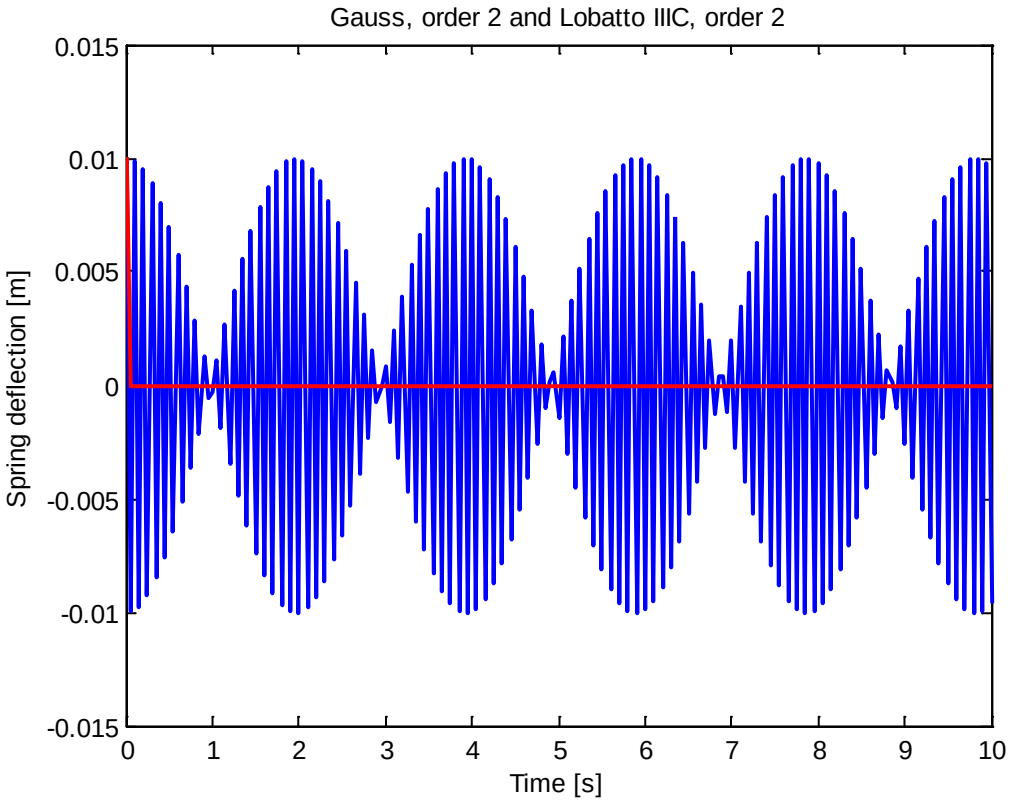
Position of the two masses

Gauss, order 2 and Lobatto IIIC, order 2, h = 0.05



Spring deflection

Gauss, order 2 and Lobatto IIIC, order 2, h = 0.05



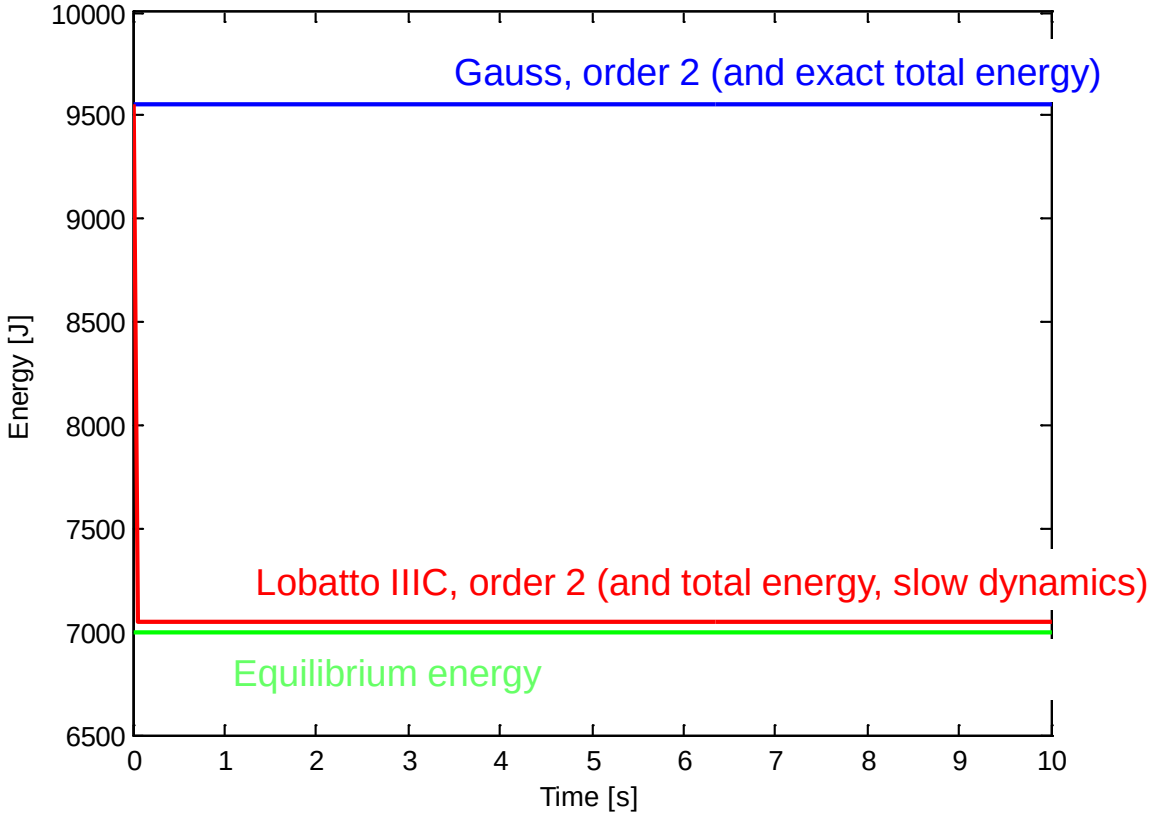
Gauss, order 2

Lobatto IIIC, order 2

- Gauss method gives no damping, but shifts fast dynamics and energy to frequencies below Nyquist frequency, $\omega_N = \frac{\pi}{h} = \frac{\pi}{0.05} = 62.8$
- Lobatto IIIC dampens out fast dynamics in one step

Total energies

Gauss, order 2 and Lobatto IIIC, order 2, h = 0.05



- Gauss does not dampen energies at all (same as exact total energy)
- Lobatto IIIC dampens out energy associated with fast dynamics in very few steps, to the energy of slow dynamics

Kahoot

- https://play.kahoot.it/#/k/694ab821-e4e0-421a-a5d6-0d297fd2cf1c