Lecture 7: Implicit Runge-Kutta Methods

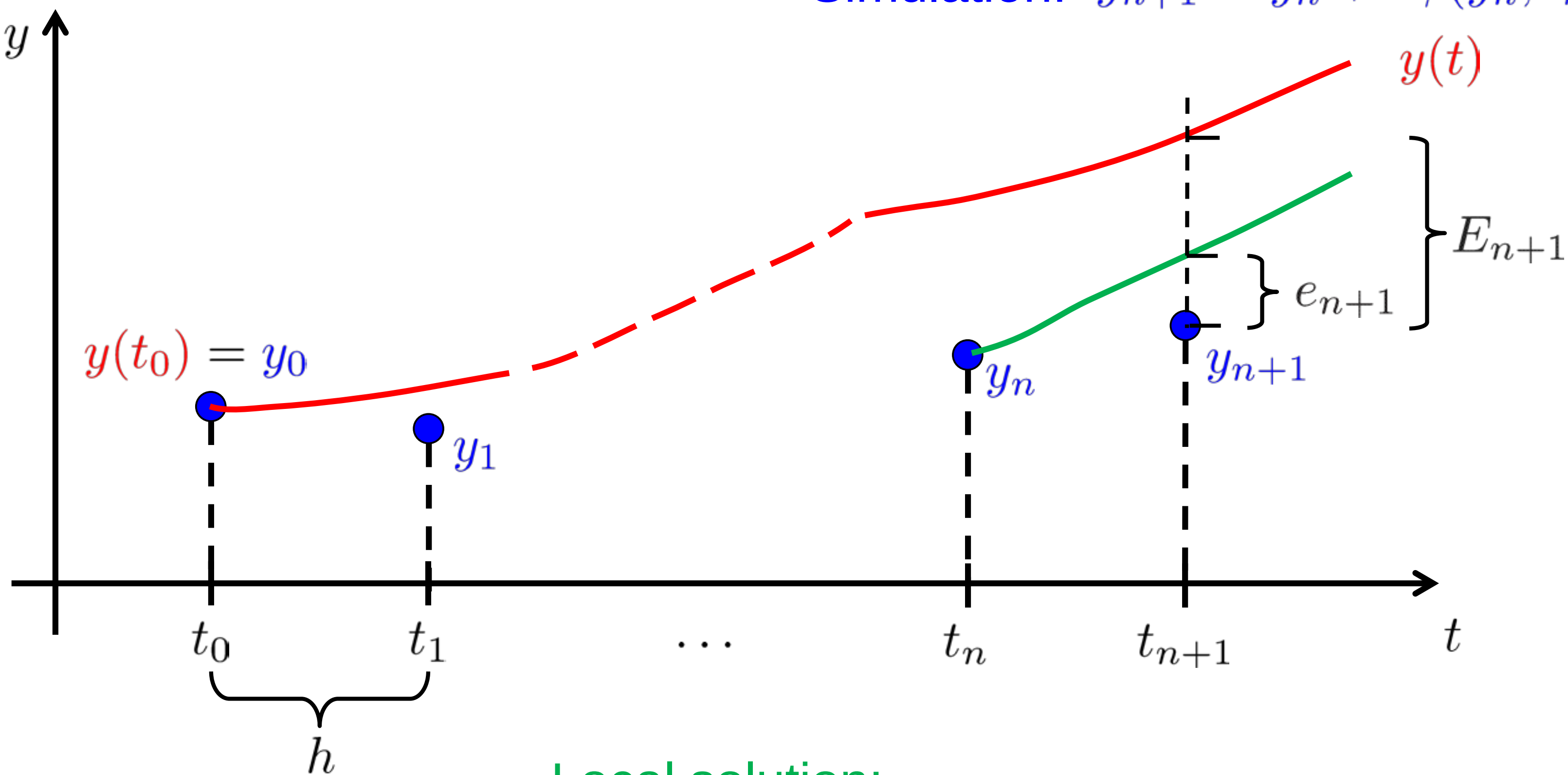
- Recap Explicit Runge-Kutta (ERK) methods
- Stiff systems
- Implicit Runge-Kutta (IRK) ODE solvers

Book: 14.5 (+ 14.8.1)

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

Recap: Explicit Runge-Kutta (ERK) methods

- IVP: $\dot{y} = f(y, t), \quad y(0) = y_0$
- One-step methods: $y_{n+1} = y_n + h\phi(y_n, t_n), \quad h = t_{n+1} - t_n$
- 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:

$$
\begin{array}{c|c}
\mathbf{c} & \mathbf{A} \\
\hline
 & \mathbf{b}^\mathsf{T}
\end{array}
\qquad
\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{21} & & & & \\
c_3 & a_{31} & a_{32} & & & \\
\vdots & \vdots & \vdots & \ddots & & \\
c_\sigma & a_{\sigma,1} & a_{\sigma,2} & \cdots & a_{\sigma,\sigma-1} & \\
\hline
 & b_1 & b_2 & \ldots & b_{\sigma-1} & b_\sigma
\end{array}
$$

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

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

Stability function for RK-methods

- Two alternative, equivalent expressions can be derived:
  - Either

$$R(h\lambda) = 1 + h\lambda \mathbf{b}^{\mathsf{T}} \left(\mathbf{I} - h\lambda \mathbf{A}\right)^{-1} \mathbf{1}$$

  - or

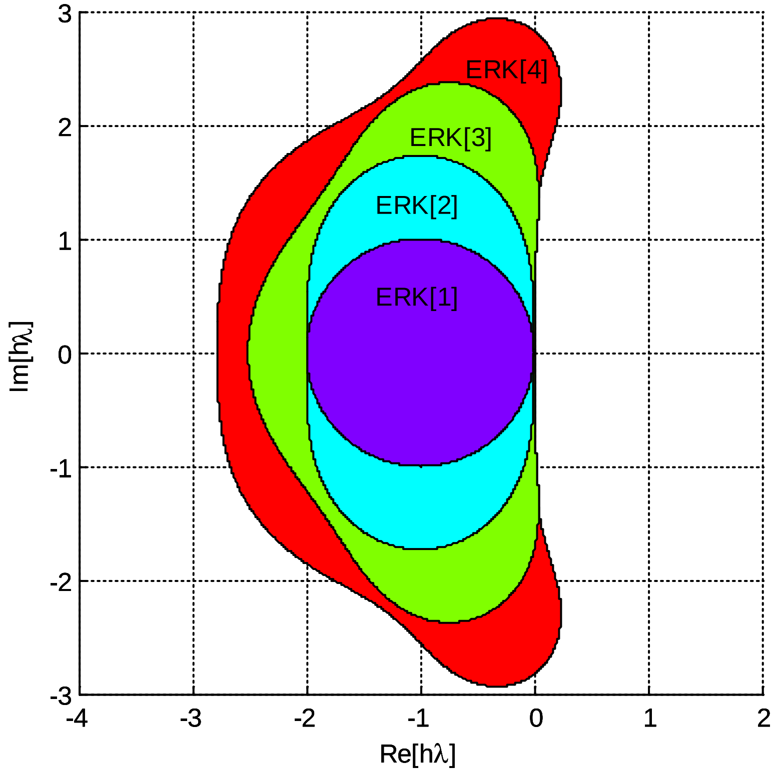
$$R(h\lambda) = \frac{\det\left[\mathbf{I} - h\lambda\left(\mathbf{A} - \mathbf{1}\mathbf{b}^{\mathsf{T}}\right)\right]}{\det\left[\mathbf{I} - h\lambda\mathbf{A}\right]}$$

- The latter can be simplified for ERK (since A is lower triangular):

$$R_E(h\lambda) = \det\left[\mathbf{I} - h\lambda\left(\mathbf{A} - \mathbf{1}\mathbf{b}^{\mathsf{T}}\right)\right]$$

- Two observations can be made
  1. $|R_E(h\lambda)|$ will tend to infinity when $h\lambda$ goes to infinity.
  2. $R_E(h\lambda)$ is a polynomial in $h\lambda$ of order less than or equal to σ.

Stability regions for ERK methods



Order and stages

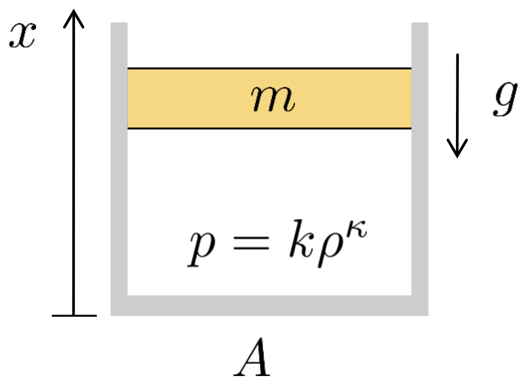
- For number of stages less than or equal to 4 it is possible to develop ERK methods (find combinations of a_{ij}, c_i, b_i) with order equal to number of stages. These are the ones that are used.

- These methods have stability function of the type

$$R_E(h\lambda) = 1 + h\lambda + \frac{(h\lambda)^2}{2} + ... + \frac{(h\lambda)^p}{p!}$$

- To obtain higher order, requires more stages:
  - Order 5 requires 6 stages
  - Order 6 requires 7 stages
  - Order 7 requires 9 stages
  - Order 8 requires 11 stages
  - …

ERK example: Pneumatic spring



- Model from Newton's 2nd law:

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

  "mass-spring-damper with nonlinear spring"

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

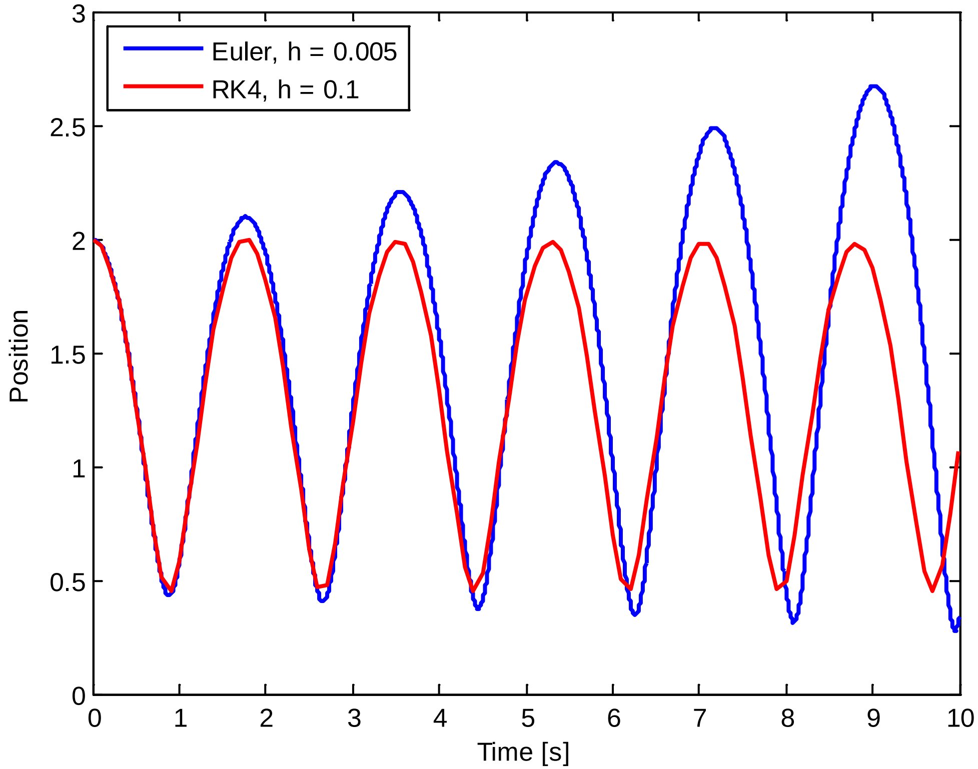
$$\begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \end{pmatrix} = \begin{pmatrix} y_2 \\ -g(1 - y_1^{-\kappa}) \end{pmatrix}$$

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

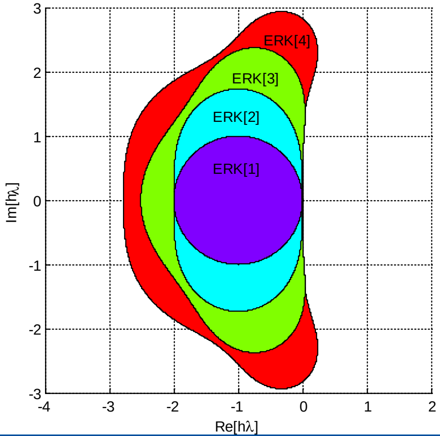
$$\frac{\partial f}{\partial y} = \begin{pmatrix} 0 & 1 \\ -g\kappa & 0 \end{pmatrix}, \qquad \lambda_{1,2} = \pm j\omega_0, \quad \omega_0 = \sqrt{g\kappa} \approx 3.7$$

Simulation

Euler: 2000 function evaluations
RK4: 400 function evaluations

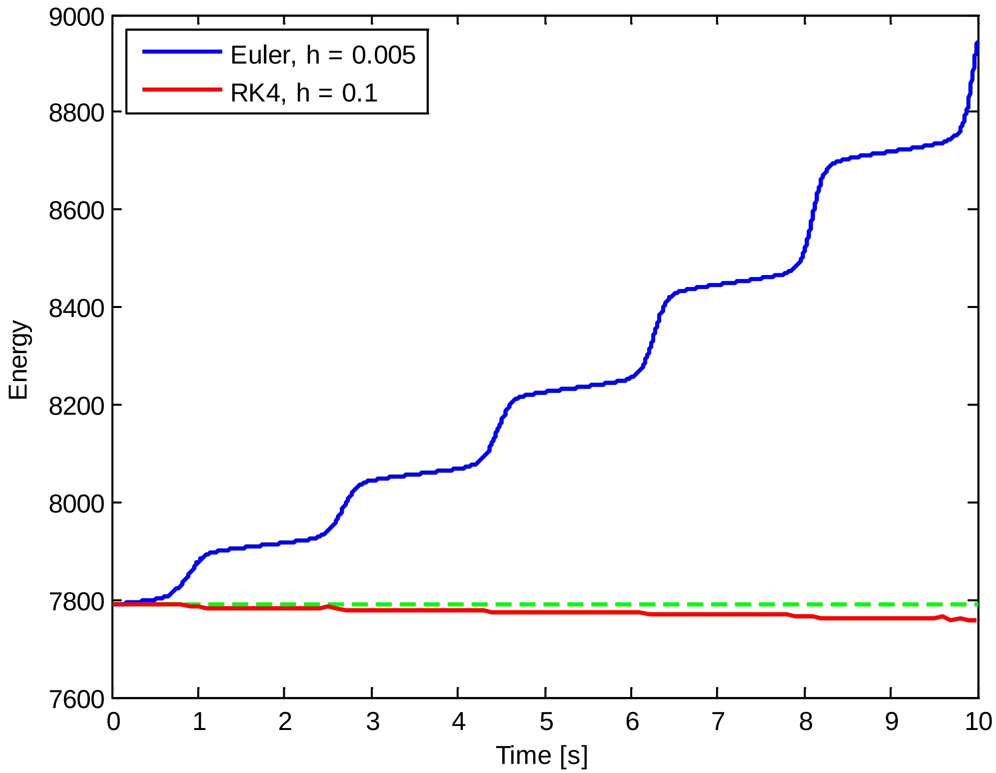


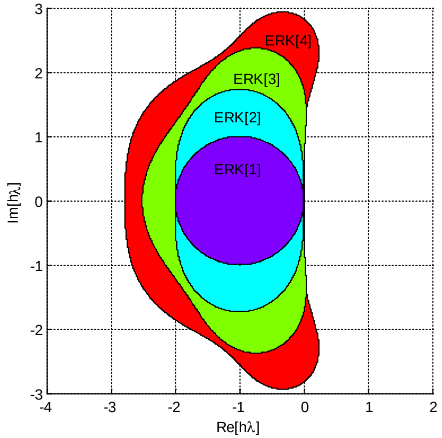
- Stability, RK4
  - Theoretical: $\omega_0 h \approx 2.83 \quad \Rightarrow \quad h \approx 0.76$
  - Practically: $h \approx 0.52$



Pneumatic spring: Accuracy

- Energy should be constant

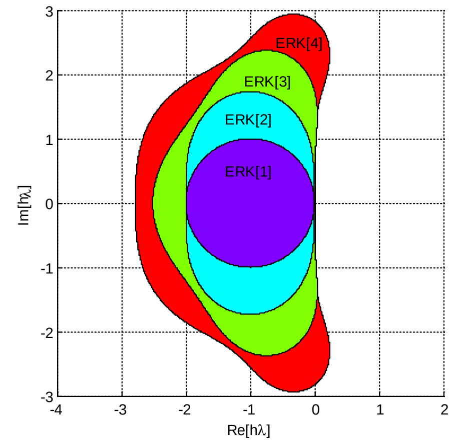




Kahoot

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

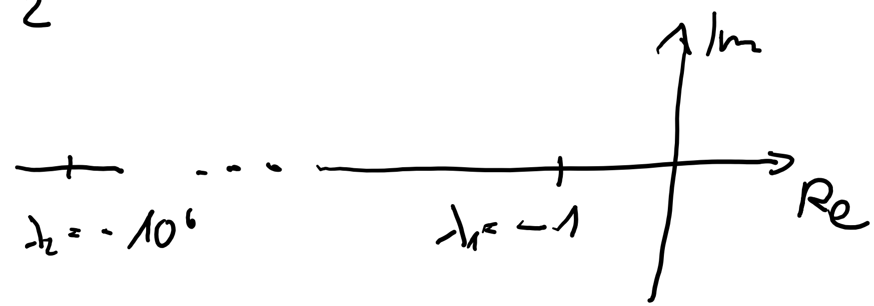
Motivation: Implicit RK



Example:

$$\dot{y}_1 = -y_1$$
$$\dot{y}_2 = -10^6 y_2$$

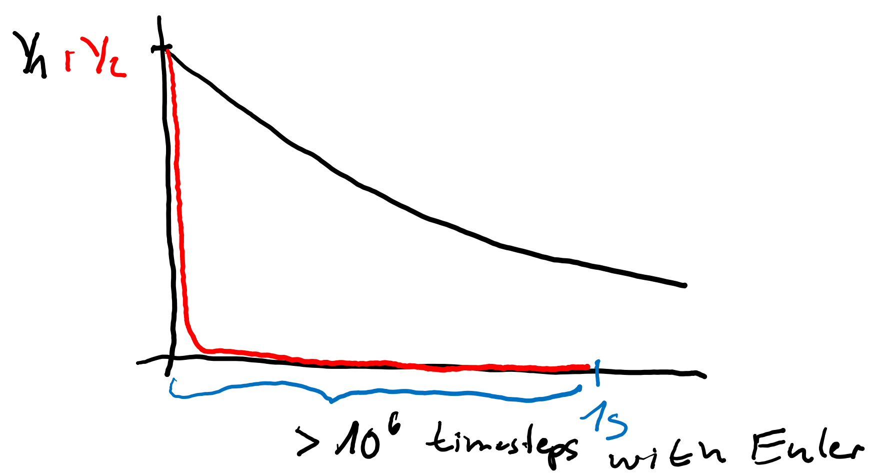
stability condition Euler: $h|\lambda| \leq 2$



$\lambda_1 \rightarrow \quad h \leq 2$

$\lambda_2 \rightarrow \quad h \leq 2 \cdot 10^{-6}$

We have to choose
$h < 2 \cdot 10^{-6}$



Motivation IRK: Stiff systems

Stiff system:

System that cannot be simulated effectively with explicit methods (system with large spread of eigenvalues of Jacobian)

Example: Curtiss-Hirschfelder

- IVP:

$$\dot{y} = -50(y - cos(t)) \qquad y(t_0) = 0$$

- Task Simulate from t = 0 s to t = 2 s

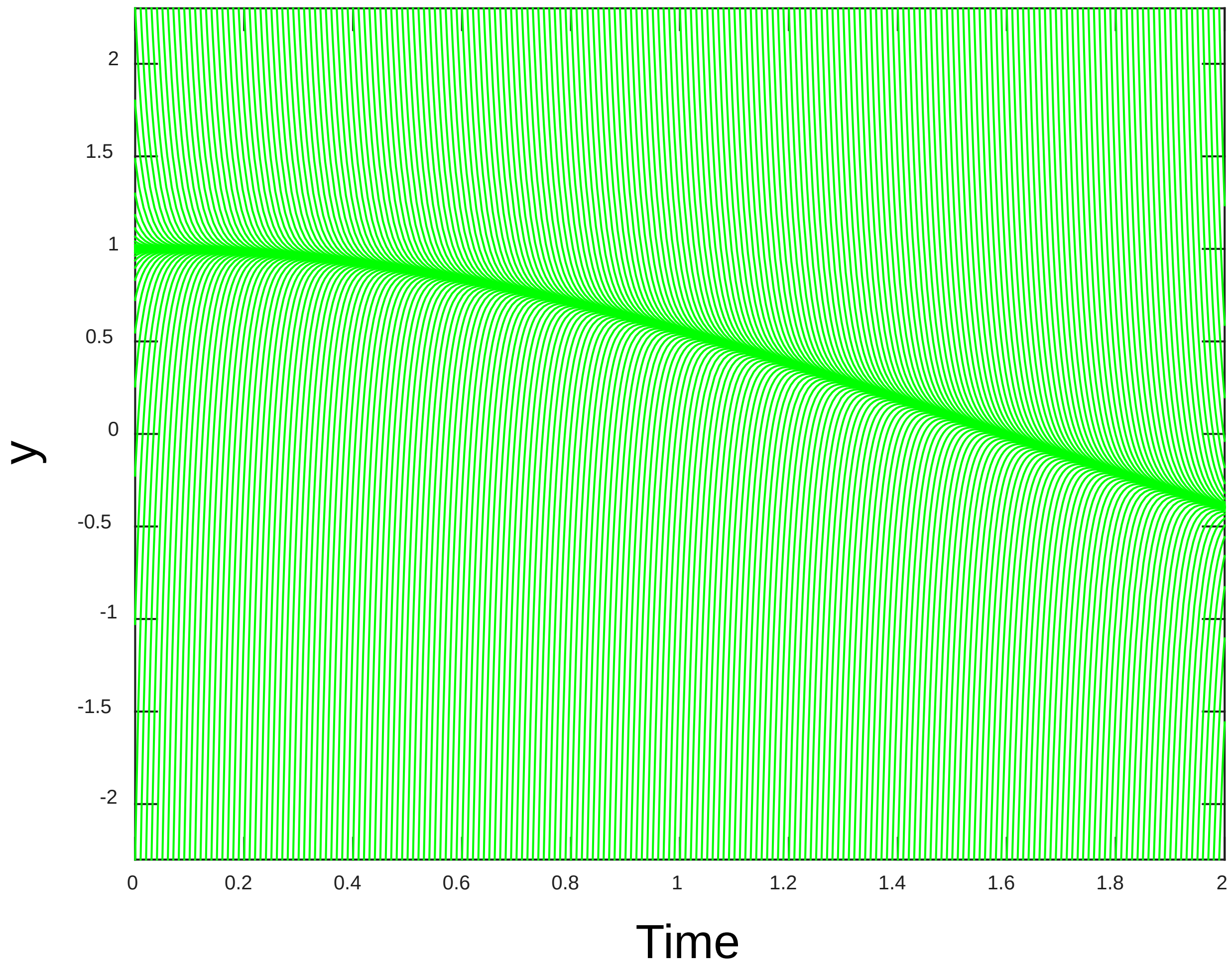
- Two widely different time scales:
  - Slow manifold

$$y^S(t) = cos(t)$$

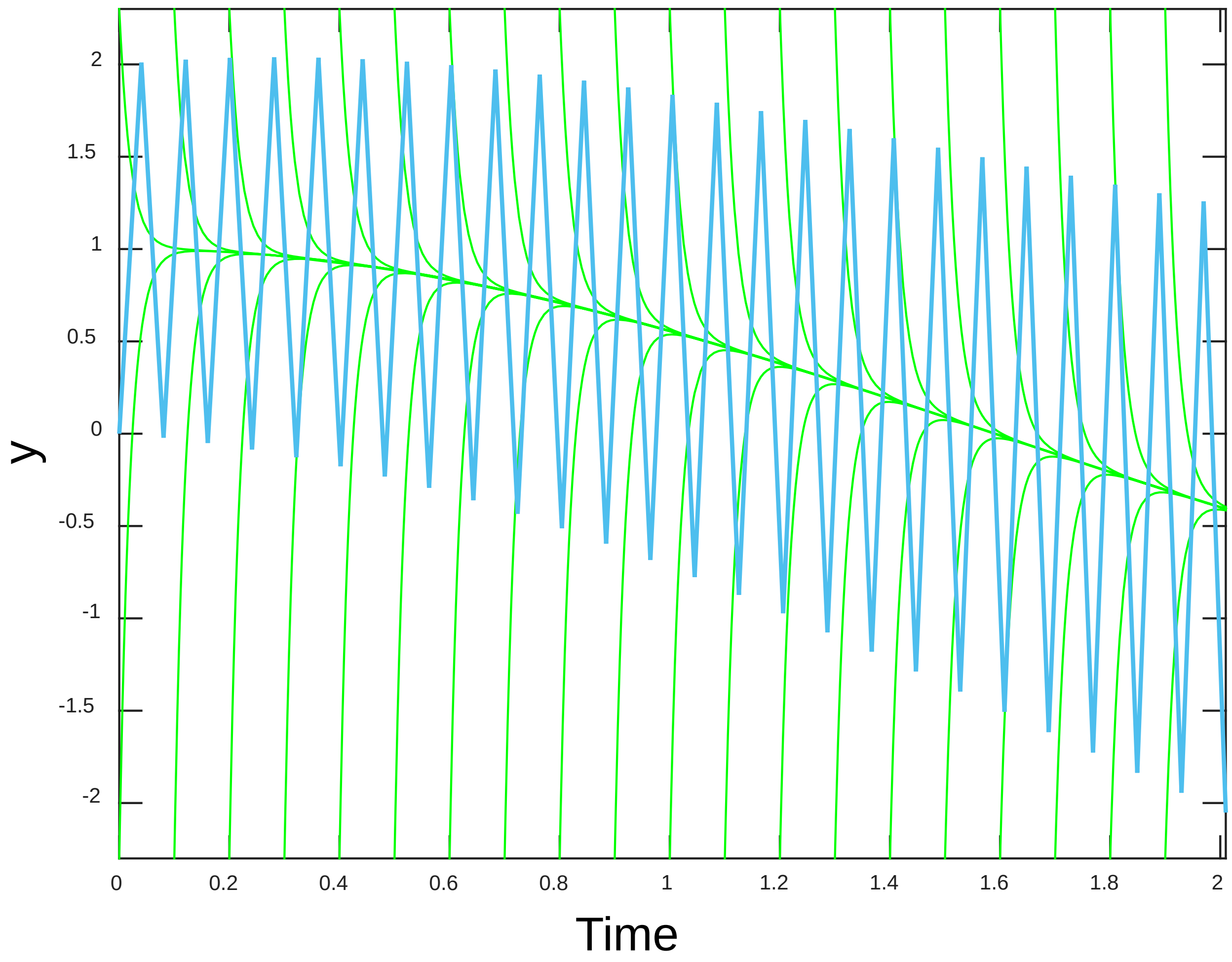
  - Strongly damped mode

$$\exp(-50t)$$

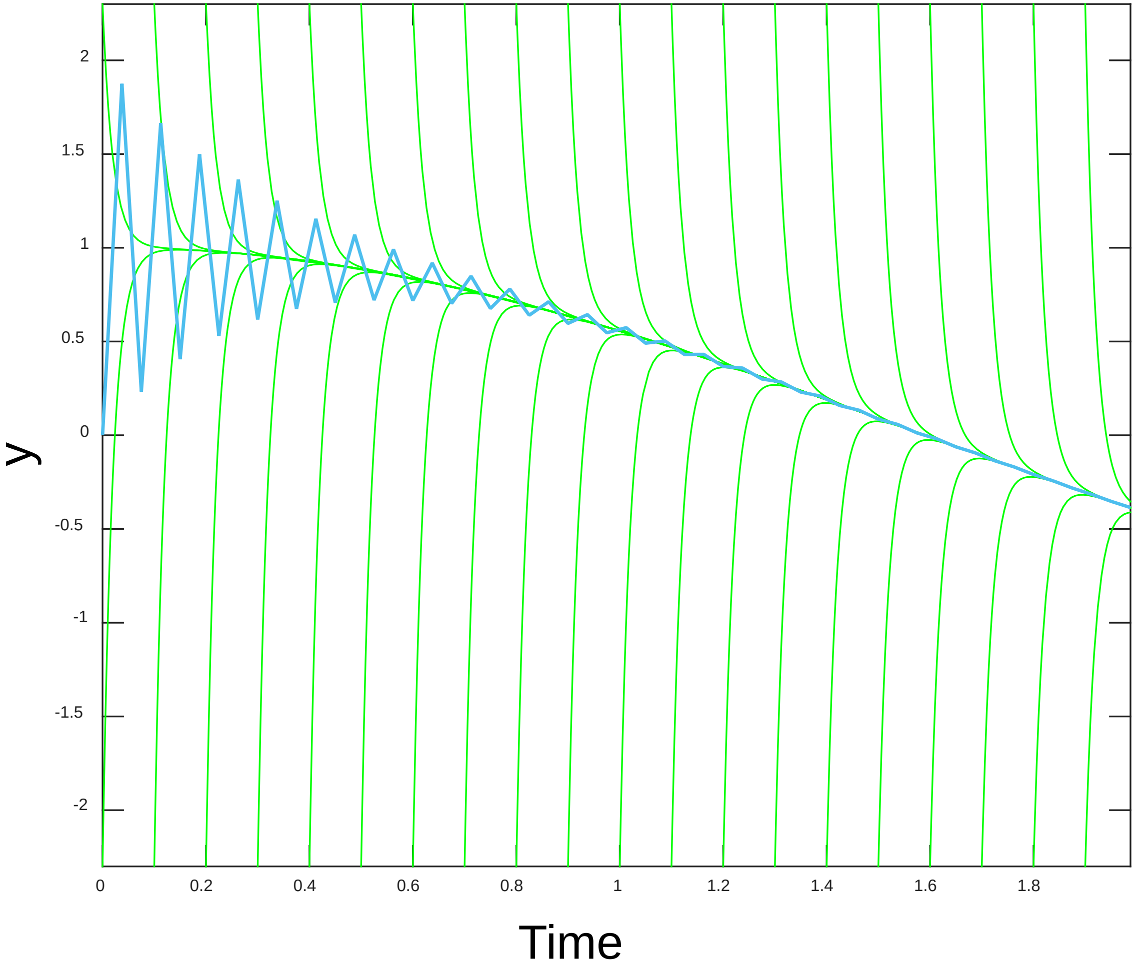
Solution manifold



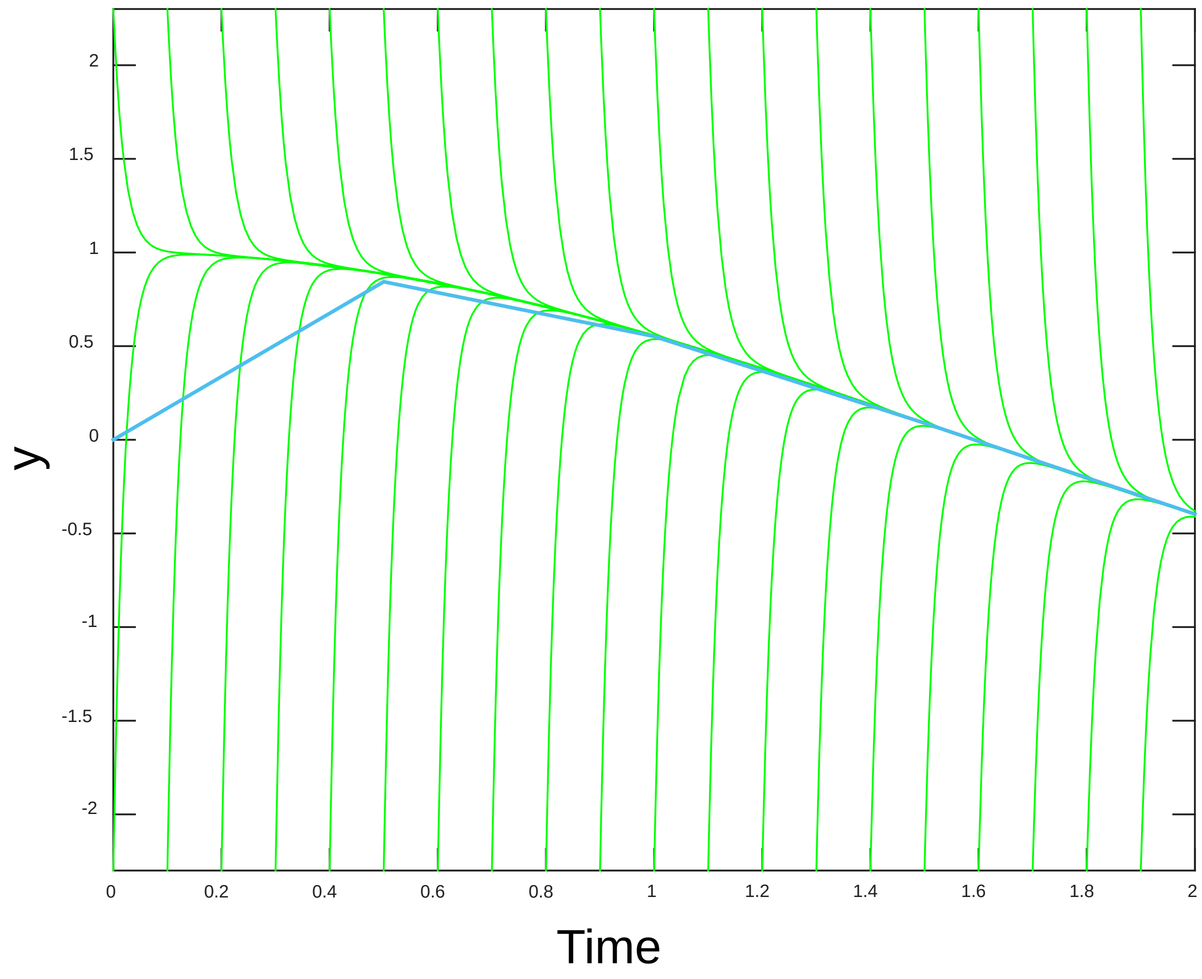
Attempt 1: Euler (explicit), h = 0.0402



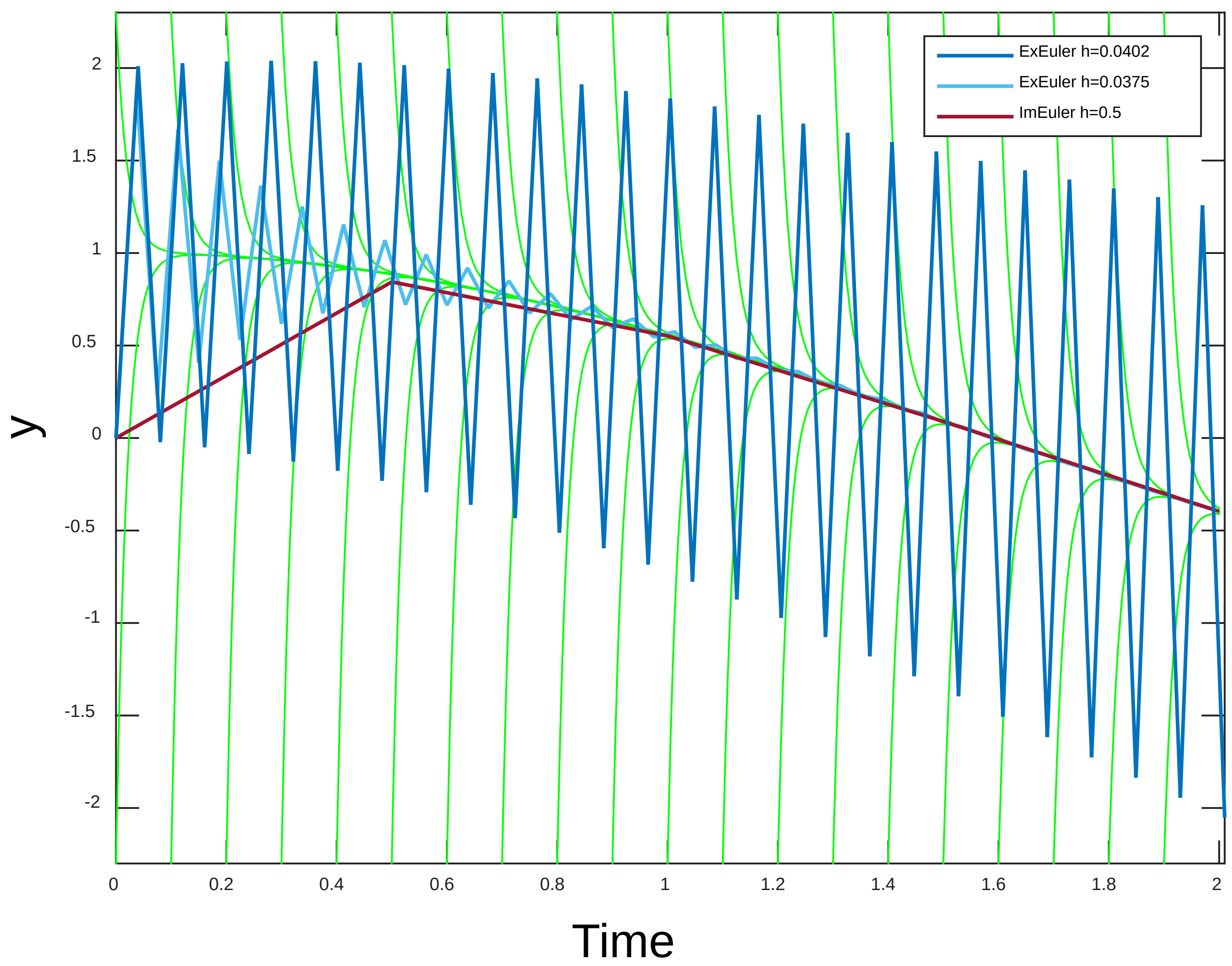
Attempt 2: Euler (explicit), h = 0.0375



Attempt 3: Euler (implicit), h = 0.5



Comparison


ExEuler h=0.0402
ExEuler h=0.0375
ImEuler h=0.5
y
Time
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

Recap: Explicit Runge-Kutta (ERK) methods

- IVP: $\dot{y} = f(y,t), \quad y(0) = y_0$
- One-step methods: $y_{n+1} = y_n + h\phi(y_n, t_n), \quad h = t_{n+1} - t_n$
- 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:

$$
\begin{array}{c|c}
\mathbf{c} & \mathbf{A} \\
\hline
 & \mathbf{b}^\mathsf{T}
\end{array}
\qquad
\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{21} & & & & \\
c_3 & a_{31} & a_{32} & & & \\
\vdots & \vdots & \vdots & \ddots & & \\
c_\sigma & a_{\sigma,1} & a_{\sigma,2} & \cdots & a_{\sigma,\sigma-1} & \\
\hline
 & b_1 & b_2 & \ldots & b_{\sigma-1} & b_\sigma
\end{array}
$$

Implicit Runge-Kutta (IRK) methods

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

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

$0 \leq c_i \leq 1$

$\sum_{i=1}^{\sigma} b_i = 1$

- Butcher array:

$$
\begin{array}{c|c}
\mathbf{c} & \mathbf{A} \\
\hline
 & \mathbf{b}^\mathsf{T}
\end{array}
$$

$$
\begin{array}{c|ccccc}
c_1 & a_{11} & a_{12} & \cdots & a_{1,\sigma-1} & a_{1,\sigma} \\
c_2 & a_{21} & a_{22} & \cdots & a_{2,\sigma-1} & a_{2,\sigma} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
c_\sigma & a_{\sigma,1} & a_{\sigma,2} & \cdots & a_{\sigma,\sigma-1} & a_{\sigma,\sigma} \\
\hline
 & b_1 & b_2 & \ldots & b_{\sigma-1} & b_\sigma
\end{array}
$$

$\sum_{j=1}^{\sigma} a_{ij} = c_i$

(usually)

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 you 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

Implicit Euler method: Order

$$k_1 = f(y_n + hk_1, t_n + h)$$

$$y_{n+1} = y_n + hk_1$$

- Taylor series expansion of k_1:

$$k_1 = f(y_n, t_n) + h\frac{df(y_n, t_n)}{dt} + O(h^2)$$

- Solution:

$$y_{n+1} = y_n + hf(y_n, t_n) + h^2\frac{f(y_n, t_n)}{dt} + O(h^3)$$

- Error: $O(h^2)$

  → method or order 1

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

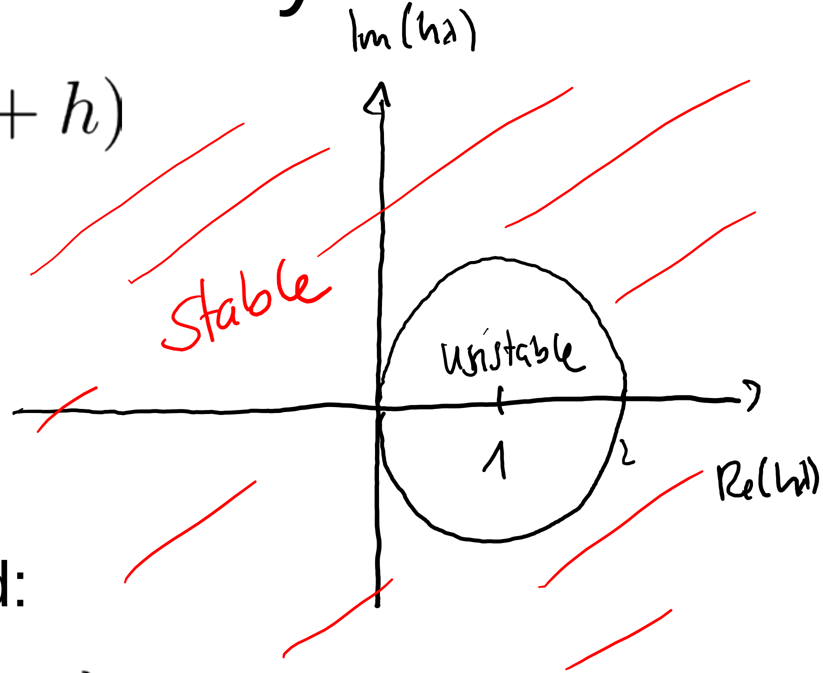
- The method is stable (for test system!) if

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

Implicit Euler method: Stability

$$k_1 = f(y_n + hk_1, t_n + h)$$

$$y_{n+1} = y_n + hk_1$$



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

- Applied to Implicit Euler method:

$$k_1 = \lambda(y_n + hk_1) = \lambda y_{n+1}$$

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

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

$$|R(h\lambda)| \leq 1 \rightarrow |1-h\lambda| \geq 1$$

IRK: Trapezoidal rule

$$
\begin{aligned}
k_1 &= f(y_n, t_n) \\
k_2 &= f\left(y_n + \frac{h}{2}(k_1 + k_2), t_n + h\right) \\
y_{n+1} &= y_n + \frac{h}{2}(k_1 + k_2) \\
&= y_n + \frac{h}{2}\left[f(y_n, t_n) + f(y_{n+1}, t_{n+1})\right]
\end{aligned}
$$

IRK: Trapezoidal rule: Order: 2

$$
\begin{aligned}
k_1 &= f(y_n, t_n) \\
k_2 &= f(y_n + \frac{h}{2}(k_1 + k_2), t_n + h) \\
y_{n+1} &= y_n + \frac{h}{2}(k_1 + k_2) \\
&= y_n + \frac{h}{2}[f(y_n, t_n) + f(y_{n+1}, t_{n+1})] \\
&= y_n + \frac{h}{2}[f(y_n, t_n) + f(y_n, t_n) + h\frac{df(y_n, t_n)}{dt} + O(h^2)] \\
&= y_n + hf(y_n, t_n) + \frac{h^2}{2}\frac{df(y_n, t_n)}{dt} + O(h^3)
\end{aligned}
$$

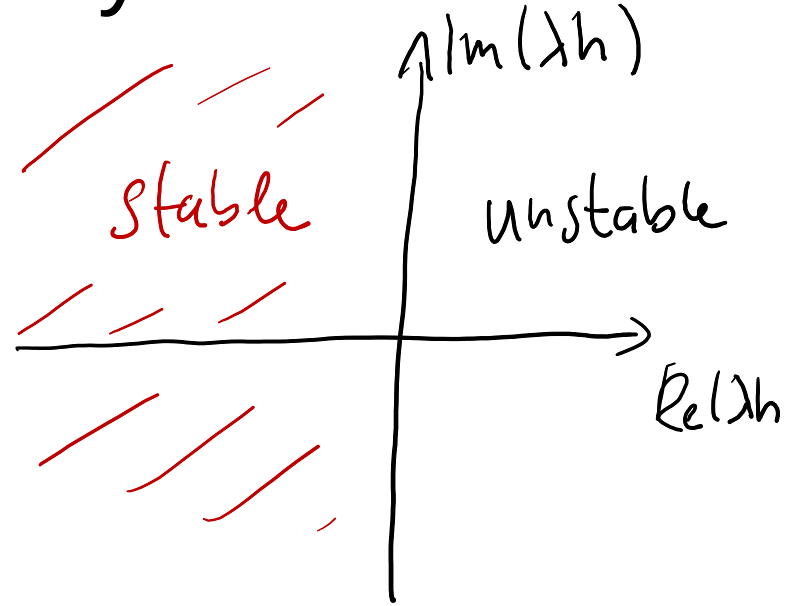
Trapezoidal rule – stability

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

$$k_2 = f(y_n + \frac{h}{2}(k_1 + k_2), t_n + h)$$

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

$$= y_n + \frac{h}{2}[f(y_n, t_n) + f(y_{n+1}, t_{n+1})]$$



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

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

$$y_{n+1} = \underbrace{\frac{1 + \frac{h\lambda}{2}}{1 - \frac{h\lambda}{2}}}_{R(h\lambda)} y_n$$

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

IRK: Implicit midpoint rule

$$
\begin{aligned}
k_1 &= f(y_n + \frac{h}{2}k_1, t_n + \frac{h}{2}) \\
y_{n+1} &= y_n + hk_1 \\
&= y_n + h[f(y_n, t_n) + \frac{h}{2}\frac{df(y_n, t_n)}{dt} + O(h^2)]
\end{aligned}
$$

- Order ?
- Order: 2

Stability of Implicit midpoint rule

$$k_1 = f(y_n + \frac{h}{2}k_1, t_n + \frac{h}{2})$$

$$y_{n+1} = y_n + hk_1$$

trick :

$$y_n + \frac{h}{2}k_1 = \frac{y_n}{2} + \frac{y_n}{2} + \frac{h}{2}k_1$$

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

$$\Rightarrow k_1 = f\left(\frac{y_n + y_{n+1}}{2}, t_n + \frac{h}{2}\right)$$

Stabilty function of IRK

- As for ERK:

$$R(h\lambda) = 1 + h\lambda \mathbf{b}^{\mathsf{T}} \left(\mathbf{I} - h\lambda \mathbf{A}\right)^{-1} \mathbf{1}$$

- Or:

$$R(h\lambda) = \frac{\det \left[\mathbf{I} - h\lambda \left(\mathbf{A} - \mathbf{1}\mathbf{b}^{\mathsf{T}}\right)\right]}{\det \left[\mathbf{I} - h\lambda \mathbf{A}\right]}$$

- $\rightarrow R(h\lambda) = \dfrac{polynominal\ of\ h\lambda\ of\ order\ \leq \sigma}{polynominal\ of\ h\lambda\ of\ order\ \leq \sigma}$

Some implicit Runge-Kutta methods

Implicit midpoint rule

Trapezoidal rule

- Implicit Euler: $\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$

- Gauss (or Gauss-Legendre) methods:

Order 2	Order 4	Order 6
$\begin{array}{c\|c} 1/2 & 1/2 \\ \hline & 1 \end{array}$	$\begin{array}{c\|cc} \frac{1}{2}-\frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4}-\frac{\sqrt{3}}{6} \\ \frac{1}{2}+\frac{\sqrt{3}}{6} & \frac{1}{4}+\frac{\sqrt{3}}{6} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$	$\begin{array}{c\|ccc} \frac{1}{2}-\frac{\sqrt{15}}{10} & \frac{5}{36} & \frac{2}{9}-\frac{\sqrt{15}}{15} & \frac{5}{36}-\frac{\sqrt{15}}{30} \\ \frac{1}{2} & \frac{5}{36}+\frac{\sqrt{15}}{24} & \frac{2}{9} & \frac{5}{36}-\frac{\sqrt{15}}{24} \\ \frac{1}{2}+\frac{\sqrt{15}}{10} & \frac{5}{36}+\frac{\sqrt{15}}{30} & \frac{2}{9}+\frac{\sqrt{15}}{15} & \frac{5}{36} \\ \hline & \frac{5}{18} & \frac{4}{9} & \frac{5}{18} \end{array}$

- Lobatto methods:

	Order 2	Order 4
Lobatto IIIA	$\begin{array}{c\|cc} 0 & 0 & 0 \\ 1 & 1/2 & 1/2 \\ \hline & 1/2 & 1/2 \end{array}$	$\begin{array}{c\|ccc} 0 & 0 & 0 & 0 \\ 1/2 & 5/24 & 1/3 & -1/24 \\ 1 & 1/6 & 2/3 & 1/6 \\ \hline & 1/6 & 2/3 & 1/6 \end{array}$
Lobatto IIIB	$\begin{array}{c\|cc} 0 & 1/2 & 0 \\ 1 & 1/2 & 0 \\ \hline & 1/2 & 1/2 \end{array}$	$\begin{array}{c\|ccc} 0 & 1/6 & -1/6 & 0 \\ 1/2 & 1/6 & 1/3 & 0 \\ 1 & 1/6 & 5/6 & 0 \\ \hline & 1/6 & 2/3 & 1/6 \end{array}$
Lobatto IIIC	$\begin{array}{c\|cc} 0 & 1/2 & -1/2 \\ 1 & 1/2 & 1/2 \\ \hline & 1/2 & 1/2 \end{array}$	$\begin{array}{c\|ccc} 0 & 1/6 & -1/3 & 1/6 \\ 1/2 & 1/6 & 5/12 & -1/12 \\ 1 & 1/6 & 2/3 & 1/6 \\ \hline & 1/6 & 2/3 & 1/6 \end{array}$

Radau methods:

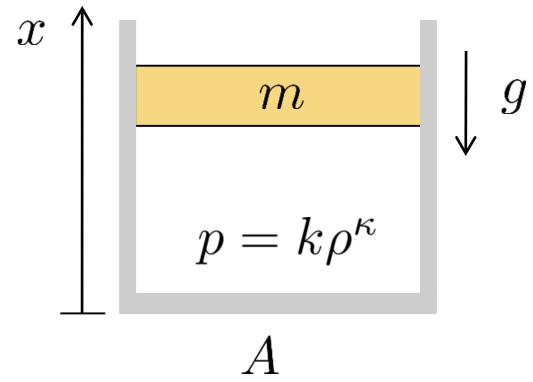
	Order 3	Order 5
Radau IA	$\begin{array}{c\|cc} 0 & 1/4 & -1/4 \\ 2/3 & 1/4 & 5/12 \\ \hline & 1/4 & 3/4 \end{array}$	$\begin{array}{c\|ccc} 0 & \frac{1}{9} & \frac{-1-\sqrt{6}}{18} & \frac{-1+\sqrt{6}}{18} \\ \frac{3}{5}-\frac{\sqrt{6}}{10} & \frac{1}{9} & \frac{11}{45}+\frac{7\sqrt{6}}{360} & \frac{11}{45}-\frac{43\sqrt{6}}{360} \\ \frac{3}{5}+\frac{\sqrt{6}}{10} & \frac{1}{9} & \frac{11}{45}+\frac{43\sqrt{6}}{360} & \frac{11}{45}-\frac{7\sqrt{6}}{360} \\ \hline & \frac{1}{9} & \frac{4}{9}+\frac{\sqrt{6}}{36} & \frac{4}{9}-\frac{\sqrt{6}}{36} \end{array}$
Radau IIA	$\begin{array}{c\|cc} 1/3 & 5/12 & -1/12 \\ 1 & 3/4 & 1/4 \\ \hline & 3/4 & 1/4 \end{array}$	$\begin{array}{c\|ccc} \frac{2}{5}-\frac{\sqrt{6}}{10} & \frac{11}{45}-\frac{7\sqrt{6}}{360} & \frac{37}{225}-\frac{169\sqrt{6}}{1800} & -\frac{2}{225}+\frac{\sqrt{6}}{75} \\ \frac{2}{5}+\frac{\sqrt{6}}{10} & \frac{37}{225}+\frac{169\sqrt{6}}{1800} & \frac{11}{45}+\frac{7\sqrt{6}}{360} & -\frac{2}{225}-\frac{\sqrt{6}}{75} \\ 1 & \frac{4}{9}-\frac{\sqrt{6}}{36} & \frac{4}{9}+\frac{\sqrt{6}}{36} & \frac{1}{9} \\ \hline & \frac{4}{9}-\frac{\sqrt{6}}{36} & \frac{4}{9}+\frac{\sqrt{6}}{36} & \frac{1}{9} \end{array}$

Pneumatic spring example, again (preview)

- Model from Newton's 2nd law:

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

  ”mass-spring-damper with nonlinear spring”



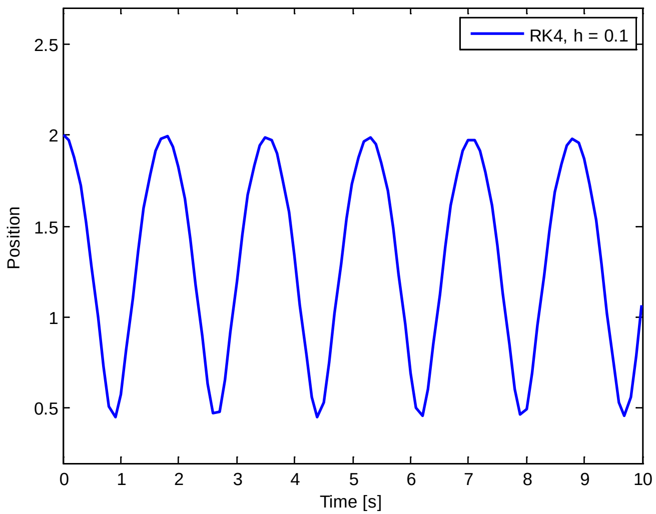
- On state-space form $\dot{y} = f(y, t)$

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

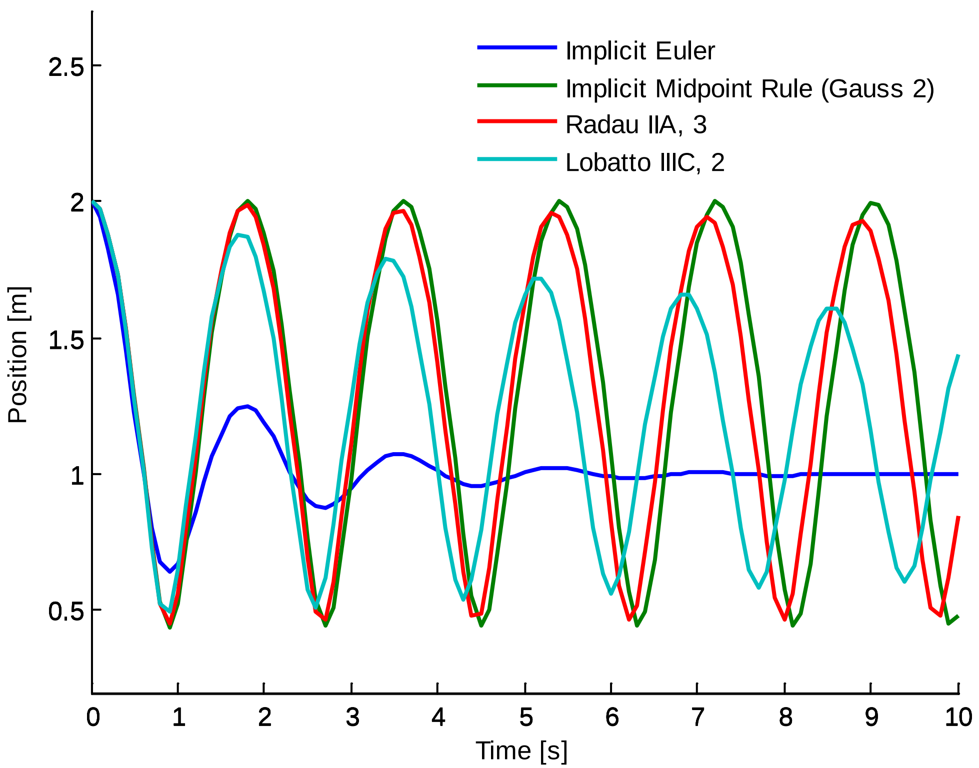
- Linearization about equilibrium:

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

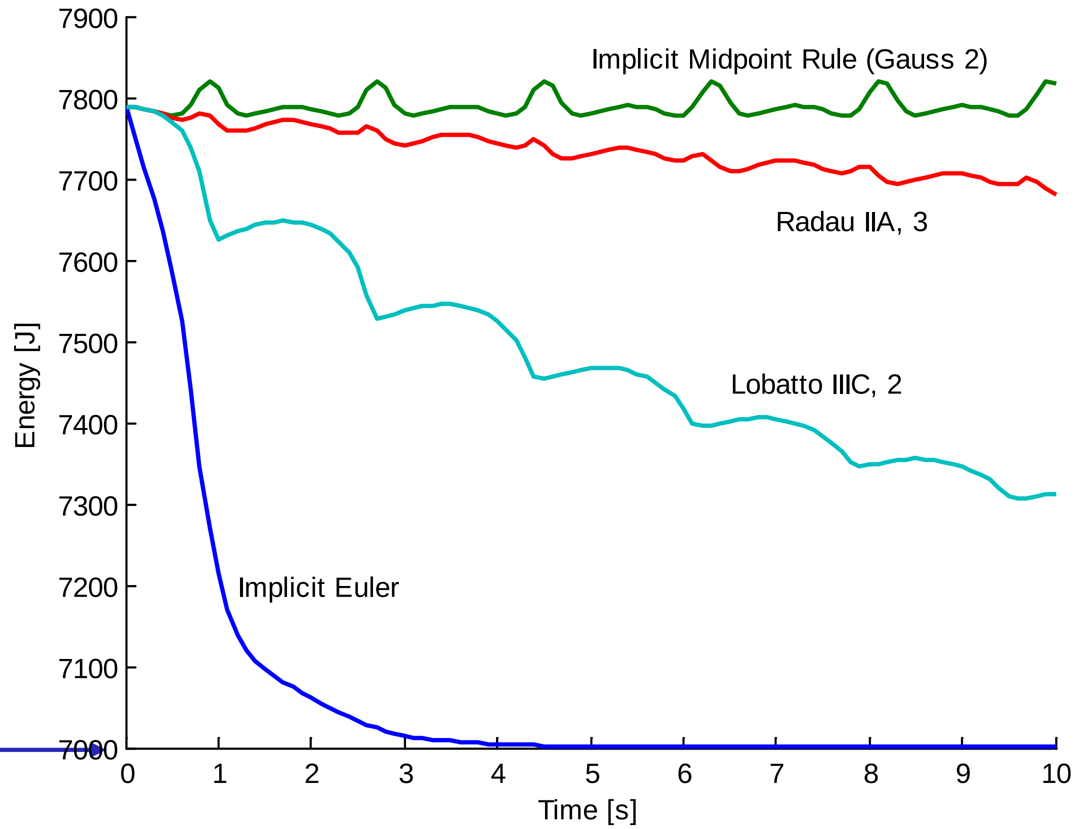
Simulation



h = 0.5 (stability limit for RK4)



Energy



Equilibrium energy

How to solve implicit equations? I

- Define z_i with:

$$k_i = f\left(y_n + h(a_{i1}k_1 + \cdots + a_{i\sigma}k_\sigma)\right) = f(y_n + z_i)$$

$$z_i = ha_{i1}f(y_n + z_1) + \cdots + ha_{i\sigma}f(y_n + z_\sigma)$$

$$\begin{pmatrix} z_1 \\ \vdots \\ z_\sigma \end{pmatrix} = h\begin{pmatrix} a_{11}I_\sigma & \dots & a_{1\sigma}I_\sigma \\ \vdots & \ddots & \vdots \\ a_{\sigma 1}I_\sigma & \dots & a_{\sigma\sigma}I_\sigma \end{pmatrix}\begin{pmatrix} f(y_n + z_1) \\ \vdots \\ f(y_n + z_\sigma) \end{pmatrix}$$

$$z = h(A \otimes I_\sigma)F(z)$$

- Use methods from optimal control to solve:

$$r(z) = z - h(A \otimes I_\sigma)F(z) = 0$$

How to solve implicit equations? II

$$r(z) = z - h(A \otimes I_\sigma)F(z) = 0$$

$$\frac{\partial r}{\partial z} = I - h(A \otimes I_\sigma) \begin{bmatrix} \frac{\partial f(y_n + z_n)}{\partial z_n} & & 0 \\ & \ddots & \\ 0 & & \frac{\partial f(y_{n}, z_\sigma)}{\partial z_\sigma} \end{bmatrix}$$

$$\approx I - h(A \otimes J)$$

$$\frac{\partial f(y_n)}{\partial y} = J$$

How to solve implicit equations? III

- If z is found:

$$y_{n+1} = y_n h b_1 f(y_n, z_1) + \cdots + h b_\sigma f(y_n + z_\sigma)$$

$$= y_n + h b^T F(z)$$

$$= y_n + h b^T \frac{1}{h} (A \otimes I_\sigma)^{-1} z$$

evaluate a priori

- Special case: (S)DIRK: (single) diagonal IRK

$$A = \begin{pmatrix} \gamma & 0 & 0 \\ x & \ddots & 0 \\ x & x & \gamma \end{pmatrix}$$

$z_1 = h\gamma f(y_1 + z_1) \to$ solve z_1

$\vdots$

$z_2 = \ldots \to$ solve z_2

σ equation system : dimension $d \times d$

→ Much faster than solving $(\sigma d \times \sigma d)$ equation system

Homework

- Implement in matlab the Euler method for the Curtiss-Hirschfelder example (slide 14)
- Write down the Butcher array of the trapozoidal rule (slide 31) and midpoint rule (slide 34)
  - Check on slide 37
- Find the stability function of the implicit midpoint rule (use hint on slide 35)

- Read 14.6.1 – 14.6.5

Kahoot

- https://play.kahoot.it/#/k/87256f68-7b17-4aa0-9c9c-c30869da5639