

ETH Lectures 401-0663-00L Numerical Methods for Computer Science
401-2673-00L Numerical Methods for CSE

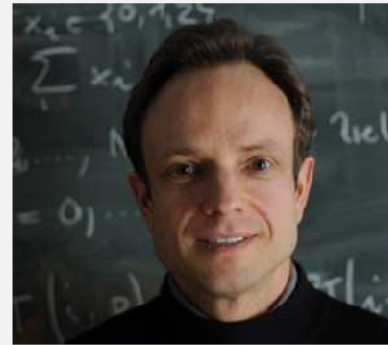
Course Video

Section 12.4: Semi-Implicit Runge-Kutta Methods

Prof. R. Hiptmair, SAM, ETH Zurich

Date: November 15, 2022

(C) Seminar für Angewandte Mathematik, ETH Zürich



Tablet notes may have been corrected and/or extended after the recording of the video!

Dependencies. Section 12.3

Duration: minutes

XII. Single-Step Methods for Stiff Initial-Value Problems

12.4 Semi-Implicit Runge-Kutta Methods

Recall: IRK-SSM for $\dot{y} = f(t, y)$

Definition 12.3.3.1. General Runge-Kutta single step method (cf. Def. 11.4.0.11)

For $b_i, a_{ij} \in \mathbb{R}$, $c_i := \sum_{j=1}^s a_{ij}$, $i, j = 1, \dots, s$, $s \in \mathbb{N}$, a single step of size $h > 0$ of an **s-stage Runge-Kutta single step method** (RK-SSM) for the IVP (11.1.3.2) is defined by

$$k_i := f(t_0 + c_i h, y_0 + h \sum_{j=1}^s a_{ij} k_j), \quad i = 1, \dots, s, \quad y_1 := y_0 + h \sum_{i=1}^s b_i k_i.$$

As before, the vectors $k_i \in \mathbb{R}^N$ are called **increments**.

→ $N_s \times N_s$ non-linear system of equations



Expensive to solve
(e.g. by simplified Newton method)
↳ Rem. 12.3.3.9

(2)



Idea: Use only a fixed *small* number of Newton steps to solve for the $\mathbf{k}_i, i = 1, \dots, s$.

Extreme case: use only a single Newton step! Let's try.

Ex. 12.4.0.1 (Semi-implicit Euler single-step method)

▷ Above idea applied to implicit Euler method

ODE : $\dot{y} = f(y)$

Impl. Euler : $y_{k+1} : y_{k+1} = y_k + h f(y_{k+1})$
(timestep h)

$$F(y_{k+1}) = y_{k+1} - y_k - h f(y_{k+1}) = 0$$

Single Newton step, initial guess y_k :

$$\begin{aligned} y_{k+1} &= y_k - DF(y_k)^{-1} F(y_k) \\ &= y_k + (I - h DF(y_k))^{-1} h f(y_k) \end{aligned}$$

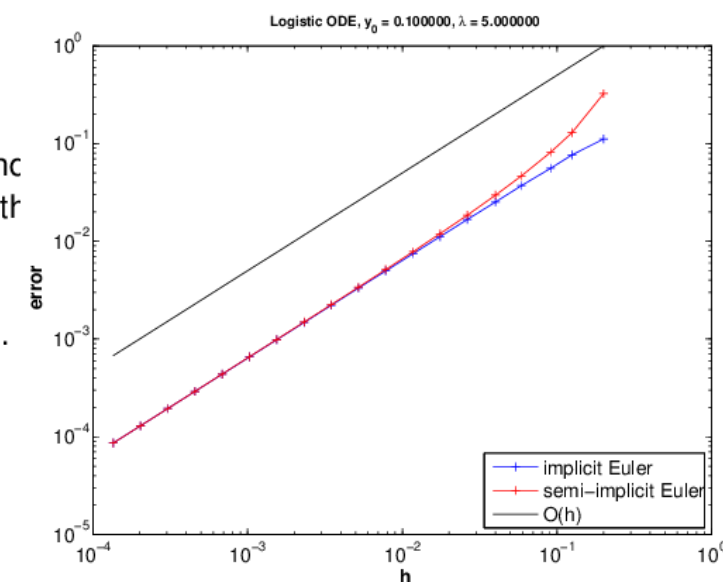
≡ semi-implicit Euler method

Exp. 12.4.0.3 (Empiric convergence of semi-implicit Euler method)

IVP : $\dot{y} = \lambda y(1-y), y(0) = 0.1, \lambda = 5$

♦ We run the implicit Euler method (11.2.2.2) and the semi-implicit Euler method (12.4.0.2) with uniform timestep $h = 1/n$,
 $n \in \{5, 8, 11, 17, 25, 38, 57, 85, 128, 192, 288, 432, 649, 973, 1460, 2189, 3284, 4926, 7389\}$.

♦ Measured error $\text{err} = \max_{j=1, \dots, n} |y_j - y(t_j)|$

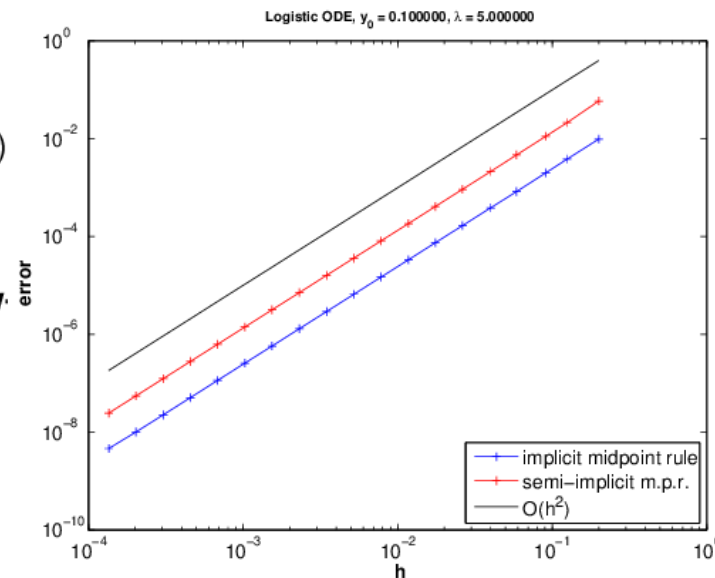


▷ Alg. conv. with rate = 1

Sem. impl. Euler : Same order & accuracy as implicit Euler

③ Exp. 12.4.0.4 (Empirical convergence of semi-implicit midpoint method)

IVP: $\dot{y} = \lambda y(1-y)$, $y(0) = 0.1$, $\lambda = 5$



Now, implicit midpoint method (11.2.3.3) uniform timesteps $h = 1/n$ as above

& approximate computation of y_{k+1} by 1 Newton step, initial guess y_k

Measured error $\text{err} = \max_{j=1, \dots, n} |y_j - y(t_j)|$

Preserves order = 2 of the fully implicit midpoint method!

Try: Use linearized increment equations for implicit RK-SSM

$$k_i := f(y_0 + h \sum_{j=1}^s a_{ij} k_j), \quad i = 1, \dots, s$$

$$k_i = f(y_0) + h Df(y_0) \left(\sum_{j=1}^s a_{ij} k_j \right), \quad i = 1, \dots, s. \quad (12.4.0.5)$$

[Drop $O(h^2)$ terms]

▷ No change for (affine-) linear ODEs
▷ Stability functions not affected by linearization

Exp. 12.4.0.6 (Naïve semi-implicit Radau method)

IVP: $\dot{y} = \lambda y(1-y)$, $y(0) = 0.1$, $\lambda = 5$

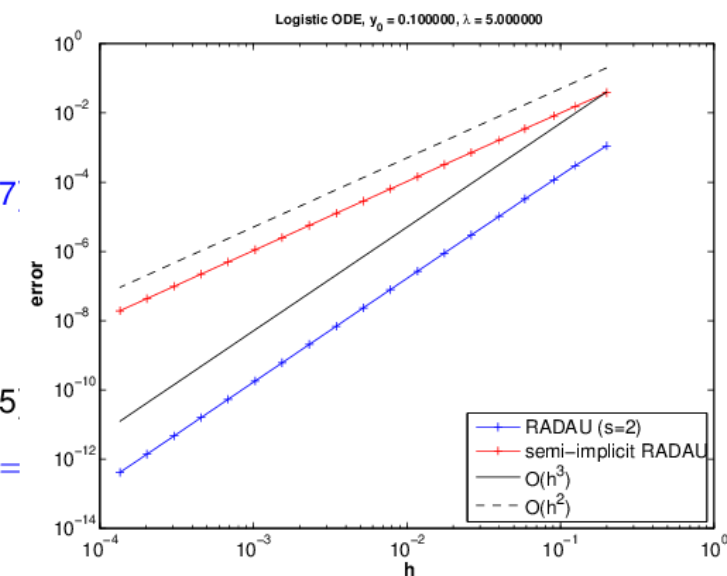
2-stage Radau RK-SSM, Butcher scheme

$$\begin{array}{c|cc} \frac{1}{3} & \frac{5}{12} & -\frac{1}{12} \\ \hline 1 & \frac{3}{4} & \frac{1}{4} \end{array}, \quad (12.4.0.7)$$

order = 3, see Ex. 12.3.4.21.

Increments from linearized equations (12.4.0.5)

We monitor the error through $\text{err} = \max_{j=1, \dots, n} |y_j - y(t_j)|$



▷ Loss of order due to linearization

§ 12.4.0.8 (Rosenbrock-Wanner methods)

General formula for ODE $\dot{y} = f(y)$

$$(\mathbf{I} - h a_{ii} \mathbf{J}) \mathbf{k}_i = \mathbf{f}(y_0 + h \sum_{j=1}^{i-1} (a_{ij} + d_{ij}) \mathbf{k}_j) - h \mathbf{J} \sum_{j=1}^{i-1} d_{ij} \mathbf{k}_j, \quad \mathbf{J} = \mathbf{D} f(y_0), \quad (12.4.0.9)$$

$$y_1 := y_0 + h \sum_{j=1}^s b_j \mathbf{k}_j.$$

Next: Determine coefficients

- to achieve higher order ($\rightarrow$ solve order conditions)
- to ensure A-/L-stability

$\downarrow$
 $\sim$ solve $N \times N$ LSE in each step

$\triangleright$ Please try to answer Review Questions 12.4.0.10 without aids

7