

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

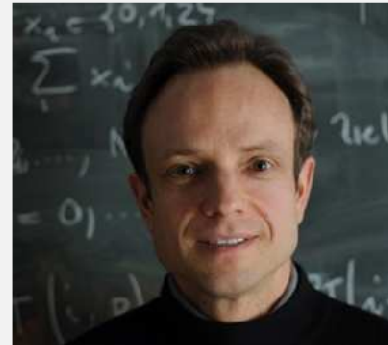
## Course Video

### Section 11.4: Explicit Runge-Kutta Single-Step Methods (RKSSMs)

Prof. R. Hiptmair, SAM, ETH Zurich

Date: November 14, 2022

(C) Seminar für Angewandte Mathematik, ETH Zürich



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

#### Prerequisites.

- Elementary calculus

**Dependencies.** Section 7.2, Section 11.3,

Duration:     minutes

# XI Numerical Integration – Single-Step Methods

## 11.4. Explicit Runge-Kutta Single-Step Methods

$$\text{IVP: } \begin{aligned} \dot{\gamma}(t) &= \underline{f}(t, \gamma) \quad , \quad \underline{f} : I \times \mathcal{D} \longrightarrow \mathcal{D} \subset \mathbb{R}^N \\ \gamma(t_0) &= \gamma_0 \in \mathcal{D} \end{aligned}$$

SSM: [autonomous case]

$$\gamma_{k+1} = \Psi^{t_{k+1}-t_k} \gamma_k$$

[general case]

$$\gamma_{k+1} = \Psi^{t_{k+1}, t_k} \gamma_k$$

goal: Construction of high-order SSMs

## ② § 11.4.0.1 (Rationale for high-order)

Order  $\longleftrightarrow$  rate of asymptotic algebraic conv.

### Order of algebraic convergence of single-step methods

Consider an IVP (11.1.3.2) with solution  $t \mapsto \mathbf{y}(t)$  and a single step method defined by the discrete evolution  $\Psi$  ( $\rightarrow$  Def. 11.3.1.5). If the *one-step error along the solution trajectory* satisfies ( $\Phi$  is the evolution map associated with the ODE, see Def. 11.1.4.3)

$$\|\Psi^h \mathbf{y}(t) - \Phi^h \mathbf{y}(t)\| \leq Ch^{p+1} \quad \forall h \text{ sufficiently small, } t \in [0, T], \quad (11.3.2.22)$$

for **some**  $p \in \mathbb{N}$  and  $C > 0$ , then, usually,

$$\max_k \|\mathbf{y}_k - \mathbf{y}(t_k)\| \leq \bar{C} h_{\mathcal{M}}^p,$$

with  $\bar{C} > 0$  independent of the temporal mesh  $\mathcal{M}$ : The (pointwise) discretization error **converges algebraically** with **order/rate**  $p$ .

unknown in general

[ Same considerations as done in the context of numerical quadrature, cf. Remark 7.4.3.12 ]

Question we can answer:

How much additional effort do we have to invest for a desired gain (in accuracy)

Order- $p$  SSM:

Assume sharpness:  $\epsilon(h) \approx Ch^p$  (\*)

Effort  $\sim$  # steps  $\sim h^{-1} \sim$  # f-evals

Gain  $\sim$  error reduction by factor  $\rho > 1$

$$\frac{\cancel{Ch_0^p}}{\cancel{Ch_n^p}} \stackrel{(*)}{=} \frac{\epsilon(h_0)}{\epsilon(h_n)} \stackrel{!}{=} \rho$$

$$\triangleright h_n = \rho^{-1/p} \cdot h_0$$

increase effort by factor  $\rho^{1/p}$

reduce error by factor  $\rho > 1$

3

# § 11.4.0.4. (Bootstrap construction of explicit SSMs)

IVP:  $\dot{y}(t) = f(t, y(t))$ ,  $y(t_0) = y_0$   $\Rightarrow$   $y(t_1) = y_0 + \int_{t_0}^{t_1} f(\tau, y(\tau)) d\tau$

$\Leftrightarrow y(t_1) = y_0 + h \int_0^1 f(t_0 + h\tilde{s}, y(t_0 + h\tilde{s})) d\tilde{s}$ ,  $h := t_1 - t_0$

Idea: Apply  $s$ -point quadrature rule on  $[0, 1]$ , weights  $w_j$ , nodes  $c_j$

Q.R.  $\Rightarrow y(t_1) \approx y_0 + h \sum_{j=1}^s w_j f(t_0 + c_j h, y(t_0 + c_j h))$

Idea:  $y(t_0 + c_j h)$  by "bootstrapping":  
 $\hookrightarrow$  by a simpler/cheaper SSM (of lower order)

∇ Error of simpler method multiplied with  $h$

## Ex. 11.4.0.6 (Simple Runge-Kutta methods by bootstrapping)

• Quadrature formula: 1D trapezoidal rule

$s = 2$ ,  $c_1 = 0$ ,  $c_2 = 1$ ,  $w_1 = w_2 = 1/2$

• Simple SSM: Explicit Euler

$y(t_1) \approx y_0 + h \sum_{j=1}^s w_j f(t_0 + c_j h, y(t_0 + c_j h))$   
 $\rightarrow y_0 + c_j h f(t_0, y_0)$

$\triangleright y_1 = y_0 + 1/2 h (f(t_0, y_0) + f(t_0 + h, y_0 + h f(t_0, y_0)))$

$\hat{=}$  explicit trapezoidal method

**II** Why explicit?

Rewritten:

$\underline{k}_1 = f(t_0, y_0)$

$\underline{k}_2 = f(t_0 + h, y_0 + h \underline{k}_1)$

$y_1 = y_0 + h (1/2 \underline{k}_1 + \underline{k}_2)$

(4)

- Quadrature formula: 1D midpoint rule  
 $S = 1$ ,  $c_1 = \frac{1}{2}$ ,  $w_1 = 1$

- Simple SSM: explicit Euler

$$y(t_1) \approx y_0 + h \sum_{j=1}^S w_j f(t_0 + c_j h, \underbrace{y(t_0 + c_j h)}_{\rightarrow y_0 + c_j h f(t_0, y_0)})$$

$$\triangleright y_1 = y_0 + h f(t_0 + \frac{1}{2}h, y_0 + \frac{1}{2}h f(t_0, y_0))$$

$\hat{=}$  *explicit midpoint method*  
 $\hookrightarrow$  no solution of eqns. required

Rewritten:

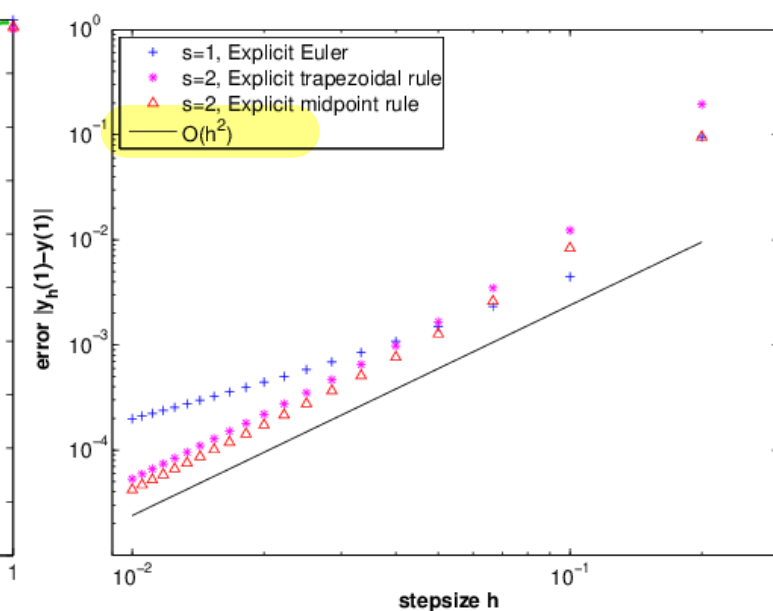
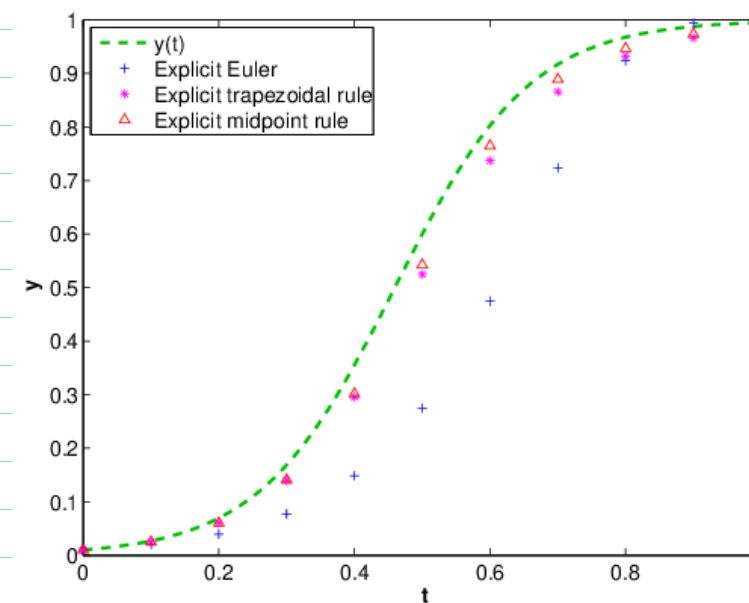
$$k_1 = f(t_0, y_0)$$

$$k_2 = f(t_0 + \frac{1}{2}h, y_0 + \frac{1}{2}h k_1)$$

$$y_1 = y_0 + h k_2$$

## Ex 6.4.0.8 (Convergence of simple RK-SSMs)

- IVP:  $\dot{y} = 10y(1-y)$  (scalar logistic ODE (6.1.2.2)), initial value  $y(0) = 0.01$ , final time  $T = 1$ ,
- Explicit single step methods, uniform timestep  $h$ .



$y_h(j/10)$ ,  $j = 1, \dots, 10$  for explicit RK-methods

Errors at final time  $y_h(1) - y(1)$

$\downarrow$   
 alg. conv. of rate 2

5

Suggests general formula

# Definition 11.4.0.11. Explicit Runge-Kutta single-step method

For  $b_i, a_{ij} \in \mathbb{R}$ ,  $c_i := \sum_{j=1}^{i-1} a_{ij}$ ,  $i, j = 1, \dots, s$ ,  $s \in \mathbb{N}$ , an **s-stage explicit Runge-Kutta single step method** (RK-SSM) for the ODE  $\dot{y} = f(t, y)$ ,  $f: \Omega \rightarrow \mathbb{R}^N$ , is defined by  $(y_0 \in D)$

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

The vectors  $k_i \in \mathbb{R}^N$ ,  $i = 1, \dots, s$ , are called **increments**,  $h > 0$  is the size of the timestep.

→ Compute  $k_0 = f(t_0, y_0) \rightarrow k_1 \rightarrow k_2$

Notation

## Butcher scheme notation for explicit RK-SSM

Shorthand notation for (explicit) Runge-Kutta methods

**Butcher scheme**

(Note that  $\mathcal{A}$  is a strictly lower triangular  $s \times s$ -matrix)

Butcher matrix

$$\begin{array}{c|c} c & \mathcal{A} \\ \hline b^T & \end{array} := \begin{array}{c} c_1 \\ c_2 \\ \vdots \\ c_s \end{array} \begin{array}{c} 0 \quad \dots \quad 0 \\ a_{21} \quad \dots \quad \vdots \\ \vdots \quad \ddots \quad \vdots \\ a_{s1} \quad \dots \quad a_{s,s-1} \quad 0 \end{array} \begin{array}{c} 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{array} \begin{array}{c} b_1 \quad \dots \quad b_{s-1} \quad b_s \end{array} \quad (11.4.0.13)$$



Discrete evolution for s-stage RK-SSM for autonomous ODE  $\dot{y} = f(y)$ ?

$$\Psi^h y = y + h \sum_{i=1}^s b_i k_i$$

Consistency of RK-SSM

## Consistent discrete evolution [for autonomous ODE]

The discrete evolution  $\Psi$  defining a single step method according to Def. 11.3.1.5 and (11.3.1.6) for the autonomous ODE  $\dot{y} = f(y)$  must be of the form

$$\Psi^h y = y + h \psi(h, y) \quad \text{with} \quad \begin{array}{l} \psi: I \times D \rightarrow \mathbb{R}^N \text{ continuous,} \\ \psi(0, y) = f(y). \end{array} \quad (11.3.1.11)$$

$$h = 0 : \quad k_i = f(y) \\ \Psi^h y = y + h \underbrace{\sum_{i=1}^s b_i k_i}_{\psi(h, y) \stackrel{!}{=} f(y) \text{ for } h=0} \quad \Rightarrow \quad \sum_{i=1}^s b_i = 1$$

## Corollary 11.4.0.15. Consistent Runge-Kutta single step methods

A Runge-Kutta single step method according to Def. 11.4.0.11 is **consistent** ( $\rightarrow$  Def. 11.3.1.12) with the ODE  $\dot{y} = f(t, y)$ , if and only if

$$\sum_{i=1}^s b_i = 1.$$

⑥ Ex. 6.4.0.15 (Butcher schemes for some explicit RK-SSMs)

Explicit Euler method (11.2.1.5):  
( $S = 1$ )

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array} \quad \triangleright$$

order = 1

explicit trapezoidal method  
(11.4.0.8):  $S = 2$

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \quad \triangleright$$

order = 2

explicit midpoint method (11.4.0.9):  
 $S = 2$

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \hline & 0 & 1 \end{array} \quad \triangleright$$

order = 2

Classical 4th-order RK-SSM:  
 $S = 4$

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6} \end{array} \quad \triangleright$$

order = 4

Kutta's 3/8-method:  
 $S = 4$

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & 1 & 0 & 0 & 0 \\ \frac{2}{3} & -\frac{1}{3} & 1 & 0 & 0 \\ 1 & 1 & -1 & 1 & 0 \\ \hline & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \end{array} \quad \triangleright$$

order = 4

Practical derivation by order conditions for coefficients  $c_j, b_i, a_{ij}$

⑦

▷ Please answer *Review Questions* 11.4.0.20  
(without peeking at notes)