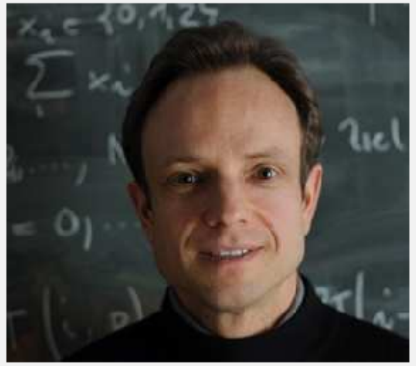
ETH Lecture 401-0674-00L Numerical Methods for ~~Partial~~ Differential Equations

Course Video

Section 6.4: Explicit Runge-Kutta Single-Step Methods (RKSSMs)

Prof. R. Hiptmair, SAM, ETH Zurich

Date: March 27, 2021



Prerequisites.

- Numerical quadrature in 1D

Dependencies. [Lecture → Section 6.3], [Lecture → Section 3.3.5]

Duration: minutes

⚠ Video and accompanying tablet notes may not match completely!

[Corrections and updates may have been made in tablet notes.]

VI. Numerical Integration – Single-Step Methods

6.4. Explicit Runge-Kutta Single-Step Methods

IVP: $\dot{\mathbf{y}}(t) = \mathbf{f}(t,\mathbf{y})$, $\mathbf{f}: I \times D \longrightarrow D \subset \mathbb{R}^N$

$\mathbf{y}(t_0) = \mathbf{y}_0 \in D$

SSM: [autonomous case]

$$\mathbf{y}_{k+1} = \Psi^{t_{k+1}-t_k}\,\mathbf{y}_k$$

[general case]

$$\mathbf{y}_{k+1} = \Psi^{t_{k+1},t_k}\,\mathbf{y}_k$$

Goal: Construction of high-order SSMs

(2) § 6.4.0.1 (Rationale for high-order)

Order $\longleftrightarrow$ rate of asymptotic algebraic cvg.

Order of algebraic convergence of single-step methods

Consider an IVP (6.1.3.2) with solution $t \mapsto \mathbf{y}(t)$ and a single step method defined by the discrete evolution $\mathbf{\Psi}$ ($\rightarrow$ Def. 6.3.1.4). If the *one-step error along the solution trajectory* satisfies ($\mathbf{\Phi}$ is the evolution map associated with the ODE, see Def. 6.1.4.3)

$$\left\|\mathbf{\Psi}^h\mathbf{y}(t) - \mathbf{\Phi}^h\mathbf{y}(t)\right\| \leq Ch^{p+1} \quad \forall h \text{ sufficiently small}, \; t \in [0,T] , \tag{6.3.2.22}$$

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

$$\varepsilon(h) := \max_k \|\mathbf{y}_k - \mathbf{y}(t_k)\| \leq \overline{C}h_{\mathcal{M}}^p ,$$

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

$\rightarrow$ value unknown

[Same considerations as in the context of convergence theory for FEMs for BVPs : Sect. 3.3.5]

Question we can answer :

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

Order-p SSM :

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

Effort : $\#\{f\text{-eval.}\} \sim \#\{\text{steps}\} \sim h^{-1}$

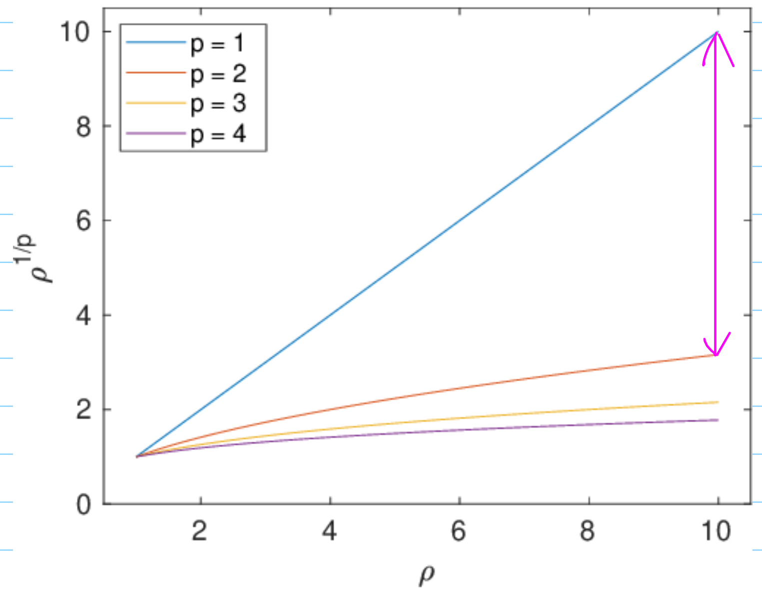
Gain : error reduction by factor $\rho > 1$

$$\frac{\cancel{C}\, h_{old}^p}{\cancel{C}\, h_{new}^p} \overset{(*)}{\approx} \frac{\varepsilon(h_{old})}{\varepsilon(h_{new})} \overset{!}{=} \rho$$

$\blacktriangleright$ $\quad h_{new} = \rho^{-1/p} \cdot h_{old}$

$$\boxed{\text{increase effort by factor } \rho^{1/p} \;\blacktriangleright\; \text{reduce error by factor } \rho > 1}$$

$\uparrow$ the smaller the larger p



§ 6.4.0.2. (Bootstrap construction of explicit SSMs)

IVP: $\begin{aligned}\dot{\mathbf{y}}(t) &= \mathbf{f}(t,\mathbf{y}(t))\,,\\ \mathbf{y}(t_0) &= \mathbf{y}_0\end{aligned}$ $\Rightarrow$ $\mathbf{y}(t_1) = \mathbf{y}_0 + \int_{t_0}^{t_1} \mathbf{f}(\tau,\mathbf{y}(\tau))\,d\tau$

$\Leftrightarrow$ $y(t_1) = y_0 + h\int_0^1 f(t_0+\xi h, y(t_0+\xi h))\,d\xi$

$[h := t_1 - t_0]$

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

$$\Rightarrow \quad y_1 = y_0 + h\sum_{j=1}^{s} w_j f(t_0+c_jh, y(t_0+c_jh))$$

Idea: $y(t_0+c_jh)$ by "bootstrapping":

↳ by simpler/cheaper/lower-order SSM

∇ Error of simpler method multiplied with h

Ex. 6.4.0.4 (Simple Runge-Kutta methods by bootstrapping)

- Quadrature formula: 1D trapezoidal rule

  $s = 2$, $c_1 = 0$, $c_2 = 1$, $w_1 = w_2 = \frac{1}{2}$

- Simple SSM: Explicit Euler

$$y_1 = y_0 + h\sum_{j=1}^{s} w_j f(t_0+c_jh, \underbrace{y(t_0+c_jh)}_{\to\, y_0 + c_jhf(t_0,y_0)})$$

$$\triangleright \quad y_1 = y_0 + h\cdot\tfrac{1}{2}\cdot\left(f(t_0,y_0) + f(t_0+h, y_0+hf(y_0))\right)$$

$\hat{=}$ explicit trapezoidal method

Why explicit?

Rewritten:

$$k_1 = f(t_0,y_0)$$
$$k_2 = f(t_0+h, y_0+hk_1)$$
$$y_1 = y_0 + h\left(\tfrac{1}{2}k_1 + \tfrac{1}{2}k_2\right)$$

(4)

- Quadrature formula: 1D midpoint rule

$$s = 1 \,, \quad c_1 = 1/2 \,, \quad w_1 = 1$$

- Simple SSM: explicit Euler

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

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

$\hat{=}$ explicit midpoint method

Rewritten:

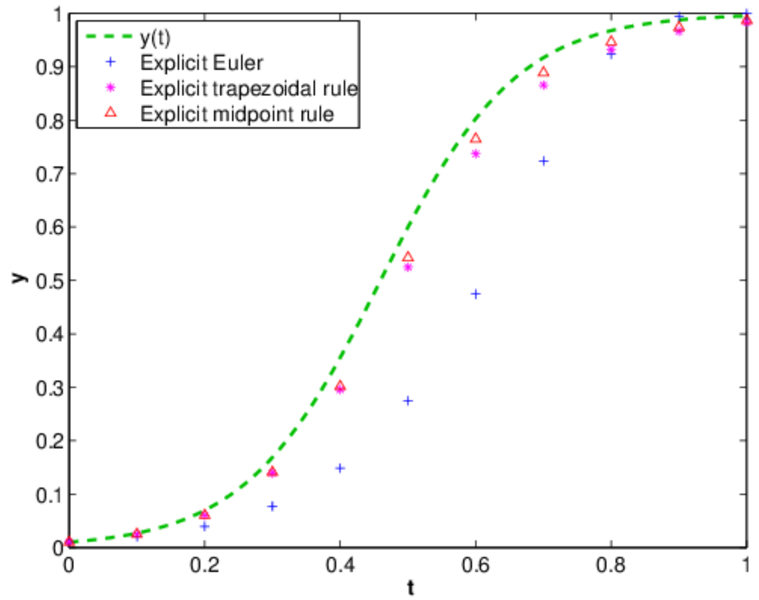
$$\boxed{k_1} = f(t_0, y_0)$$

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

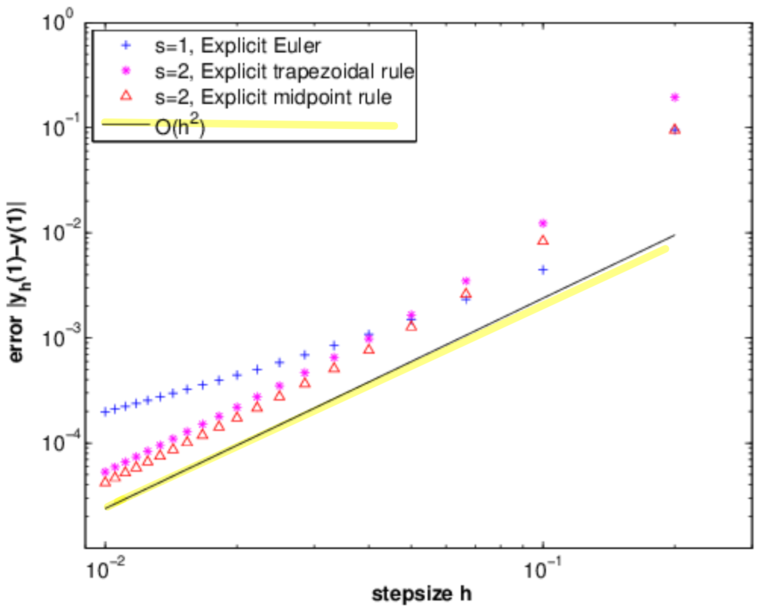
$$y_1 = y_0 + h k_2$$

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.



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



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

EMM/ETM $\to$ emp. alg. cvg. rate $= 2$

(5) Suggests general formula

→ f-eval. enough

Definition 6.4.0.9. Explicit Runge-Kutta 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 $\dot{\mathbf{y}} = \mathbf{f}(t,\mathbf{y})$, $\mathbf{f} : \Omega \to \mathbb{R}^N$, is defined by ($\mathbf{y}_0 \in D$)

$$\mathbf{k}_i := \mathbf{f}\Big(t_0 + c_i h, \mathbf{y}_0 + h\sum_{j=1}^{i-1} a_{ij}\mathbf{k}_j\Big)\,, \quad i = 1,\dots,s \quad , \quad \mathbf{y}_1 := \mathbf{y}_0 + h\sum_{i=1}^{s} b_i \mathbf{k}_i = \Psi^h y_0$$

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

→ Compute $k_1 \to k_2 \to \dots \to k_s$

Notation

Butcher scheme notation for explicit RK-SSM

Shorthand notation for (explicit) Runge-Kutta methods

Butcher scheme ▷

$$\begin{array}{c|c} \mathbf{c} & \mathfrak{A} \\ \hline & \mathbf{b}^T \end{array} := \begin{array}{c|cccc} c_1 & 0 & \cdots & & 0 \\ c_2 & a_{21} & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ c_s & a_{s1} & \cdots & a_{s,s-1} & 0 \\ \hline & b_1 & \cdots & b_{s-1} & b_s \end{array} \,. \tag{6.4.0.11}$$

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

Butcher matrix

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-SSMs

Consistent discrete evolution [for autonomous ODE]

The discrete evolution $\boldsymbol{\Psi}$ defining a single step method according to Def. 6.3.1.4 and (6.3.1.5) for the autonomous ODE $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y})$ must be of the form

$$\boldsymbol{\Psi}^h\mathbf{y} = \mathbf{y} + h\boldsymbol{\psi}(h,\mathbf{y}) \quad \text{with} \quad \begin{array}{l} \boldsymbol{\psi} : I \times D \to \mathbb{R}^N \text{ continuous,} \\ \boldsymbol{\psi}(0,\mathbf{y}) = \mathbf{f}(\mathbf{y})\,. \end{array} \tag{6.3.1.10}$$

For $h = 0$: $k_i = f(y)$

$$\psi(0,y) = \sum_{i=1}^{s} b_i f(y) \overset{!}{=} f(y)$$

Corollary 6.4.0.13. Consistent Runge-Kutta single step methods

A Runge-Kutta single step method according to Def. 6.4.0.9 is consistent (→ Def. 6.3.1.11) with the ODE $\dot{\mathbf{y}} = \mathbf{f}(t,\mathbf{y})$, 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 (6.2.1.4): $\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$ ➢ order = 1

explicit trapezoidal method (6.4.0.6): $\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$ ➢ order = 2

explicit midpoint method (6.4.0.7): $\begin{array}{c|cc} 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \hline & 0 & 1 \end{array}$ ➢ order = 2

Classical 4th-order RK-SSM: $\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6} \end{array}$ ➢ order = 4

$s = 4$

Kutta's 3/8-method: $\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\ \frac{2}{3} & -\frac{1}{3} & 1 & 0 & 0 \\ 1 & 1 & -1 & 1 & 0 \\ \hline & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \end{array}$ ➢ order = 4

$s = 4$

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

Please answer review questions

After you have watched the video take a short break and then try to answer the

review questions 6.4.0.18

Please do not flip pages of the lecture document, nor look at tablet notes.