

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

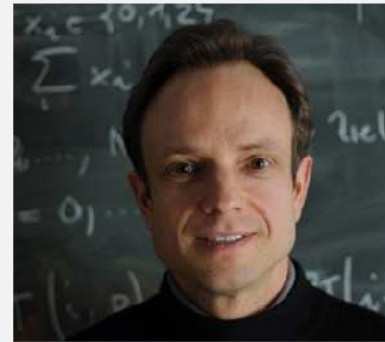
Course Video

Section 11.2: Introduction: Polygonal Approximation Methods

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: difference quotients

Dependencies. Section 5.2.3.3, Section 11.1.1 and Section 11.1.3

Duration: minutes

XI. Numerical Integration – Single-Step Methods

11.2 Introduction: Polygonal Approximation Methods

$$\text{IVP: } \dot{y} = f(t, y), \quad y(t_0) = y_0 \in D \subset \mathbb{R}^N$$

$\uparrow$ given only in *procedural form*
 (function object)

```
Eigen::VectorXd operator () (double t, const Eigen::VectorXd &y)
const;
```

Goal: Approximation of $y(t)$ on a temporal mesh/
time grid

$$\mathcal{M} := \{t_0 < t_1 < t_2 < \dots < t_{M-1} < t_M := T\} \subset [t_0, T], \quad (11.2.0.3)$$

$\uparrow$ final time

Autonomous ODE: $t_0 = 0$

② 11.2.1. Explicit Euler Method

Replace $\dot{y}$ with diff. quot. anchored on M

$$\left[\begin{array}{l} \dot{y} = f(t, y) \\ \Rightarrow \frac{y(t_{k+1}) - y(t_k)}{h_k} \approx f(t_k, y(t_k)) \end{array} \right.$$

$$h_k := t_{k+1} - t_k$$

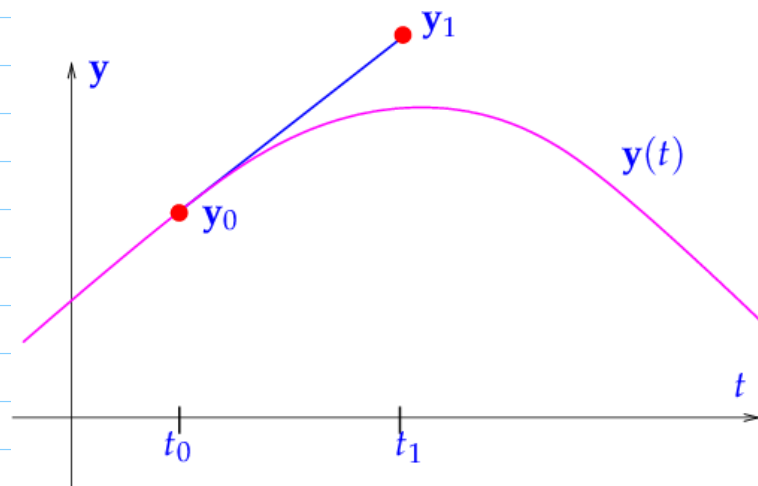
$$y_k \approx y(t_k) \Rightarrow y_{k+1} - y_k = h_k f(t_k, y_k)$$

▷ Generates a sequence $(y_k)_{k=1}^M$, $y_k \approx y(t_k)$

$$y_0 \xrightarrow{\quad} y_1 \xrightarrow{\quad} y_2 \xrightarrow{\quad} \dots \xrightarrow{\quad} y_M$$

⌞ by f-evaluations: "explicit"

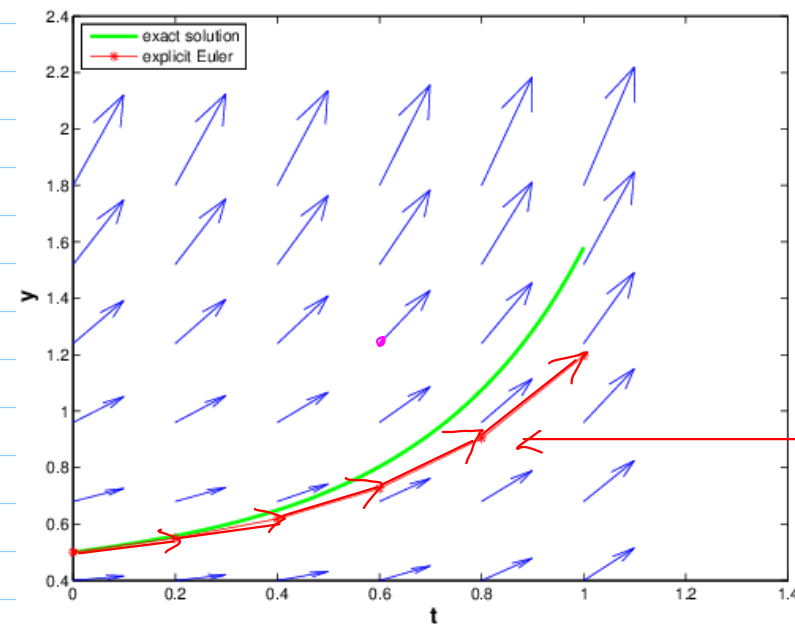
Ex. 11.2.1.3: (Visualization of explicit Euler method)



$$N = 1:$$

$$\dot{y} = f(t, y)$$

↑
slope of tangent to graph of $t \mapsto y(t)$



Ricatti ODE

$$\dot{y} = t^2 + y^2$$

→ $\hat{=}$ tangent field

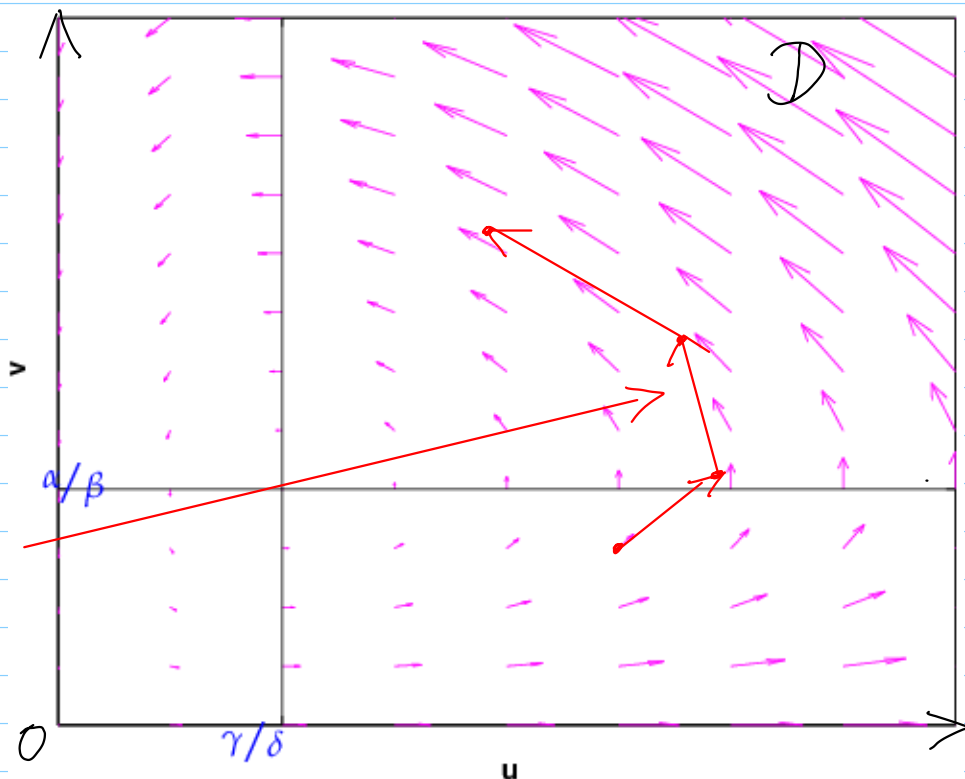
Euler polygon

Lotka-Volterra ODE:

$$D =]0, \infty[^2 \subset \mathbb{R}^2 \quad (N=2)$$

→ $\hat{=}$ velocity vector field

Euler polygon
(in state space)



③

11.2.2. Implicit Euler method

Difference quotient
or ~~all~~

$$[h_k := t_{k+1} - t_k]$$

$$\begin{aligned} \dot{y} &= f(t, y) \\ \Rightarrow \frac{y(t_{k+1}) - y(t_k)}{h_k} &\approx f(t_{k+1}, y(t_{k+1})) \end{aligned}$$

Now: backward
difference quotient

$$y_{k+1} - y_k = h_k f(t_{k+1}, y_{k+1})$$

▷ Need to solve (system of) equations to compute y_{k+1} from y_k "implicit"

$$y_{k+1} = y_k + h_k f(t_{k+1}, y_{k+1}), \quad k = 0, \dots, M-1,$$

(11.2.2.2)

with local timestep (stepsize) $h_k := t_{k+1} - t_k$.

(11.2.2.2) = implicit Euler method

Rem 11.2.2.3 : Feasibility* of implicit Euler timestepping

* (11.2.2.2) has a solution y_{k+1}

Intuition : $h_k = 0 \Rightarrow$ "solution" : $y_{k+1} = y_k$

"Continuity argument" : For $h_k \approx 0$ a solution should still exist.

$$y_{k+1} = y_k + h_k f(t_{k+1}, y_{k+1}) \Leftrightarrow G(h, y_{k+1}) = 0 \text{ with } G(h, z) := z - h f(t_{k+1}, z) - y_k.$$

$$G(0, y_k) = 0$$

$$\frac{dG}{dz}(h, z) = I - h D_y f(t_{k+1}, z) \Rightarrow \frac{dG}{dz}(0, z) = I \leftarrow \text{invertible}$$

▷ (Implicit function theorem)

Existence & uniqueness of solutions of (11.2.2.2) for "sufficiently small" $|h_k|$

11.2.3 Implicit midpoint method

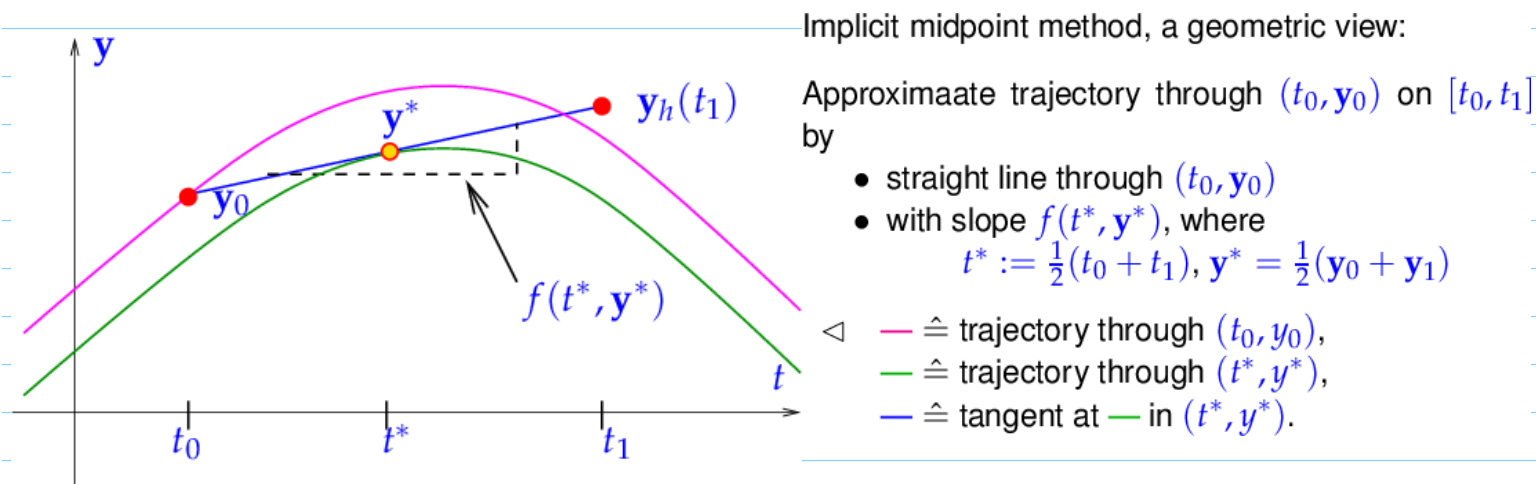
Again: replace $\dot{y}$ w/ diff. quot. on $\mathcal{M}$

$$\dot{y} = f(t, y)$$

$$\Downarrow$$

$$\frac{y_{k+1} - y_k}{h_k} \approx f\left(\frac{1}{2}(t_k + t_{k+1}), \frac{1}{2}(y_k + y_{k+1})\right)$$

$$y_{k+1} : y_{k+1} = y_k + h_k f\left(\frac{1}{2}(t_k + t_{k+1}), \frac{1}{2}(y_k + y_{k+1})\right)$$



▷ Please tackle review questions 11.2.3.4

6

7