

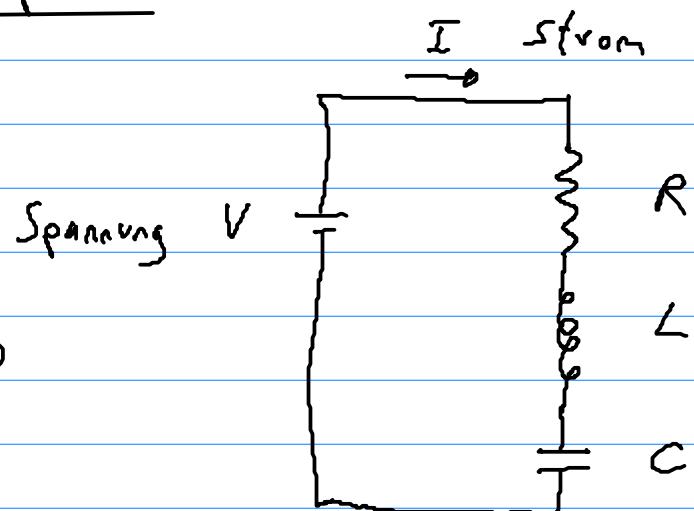
II. Explizite Einschrittverfahren

- Ziele:
- Anfangswertproblem (gew. Diff.-Gl. + Anfangsbedingungen)
 - Lösbarkeit?
 - explizite Einschrittverfahren (Euler, Runge-Kutta)
 - Genauigkeit (Diskretisierungsfehler)

Wozu: Viele Anwendungen...

II.A Motivation und Beispiele

(1) Schwingkreis



$$\ddot{I} + \frac{R}{L} \dot{I} + \frac{1}{LC} I = 0$$

$\left(\frac{\partial^2 I}{\partial t^2} \quad \frac{\partial I}{\partial t} \right)$

(Skalare) gewöhnliche Differentialgleichung
(DGL) 2. Ordnung für Strom $I(t)$

Oft: ODE für Ordinary Differential Equation

(2) Molekular-Dynamik (MD)

$$m_i \cdot \tilde{x}_i = -\tilde{D}_i \cup (\tilde{x}_1, \dots, \tilde{x}_N) , \quad i=1, \dots, N$$

wobei N ... Anzahl Atome

$\vec{x}_i \dots$ Position des i-ten Atom

U ... Potential (hängt von allen Atomen ab!)

System gewöhnlicher DGLen 2. Ordnung für die Positionen der Atome $\vec{x}(t)$ (Im Prinzip einfach nur Newton's Bewegungsgleichungen).

(3) Maxwell-Gleichungen

$$\vec{D} \times \vec{H} = \vec{G} + \frac{\partial \vec{D}}{\partial t}$$

magnetische Feldstärke Stromdichte elektrische Flussdichte

$$\vec{\nabla} \times \vec{E} = - \frac{\partial \vec{B}}{\partial t} - \text{magnetische Flussdichte}$$

'elektrische Feldstärke'

$\vec{D} \cdot \vec{D}$ = s - elektrische Ladungsdichte

$$\vec{D} \cdot \vec{B} = 0$$

System von partiellen DGL

„weil partielle Ableitungen auftauchen!“

Durch diskretisieren der Orts-Ableitungen
 (z.B. mit finiten Differenzen/Volumen/Elementen)
 erhält man ein (grosses) System von
 gewöhnlichen DGLen 1. Ordnung
 \star diese Vorlesung

oft: PDE für Partial Differential Equation

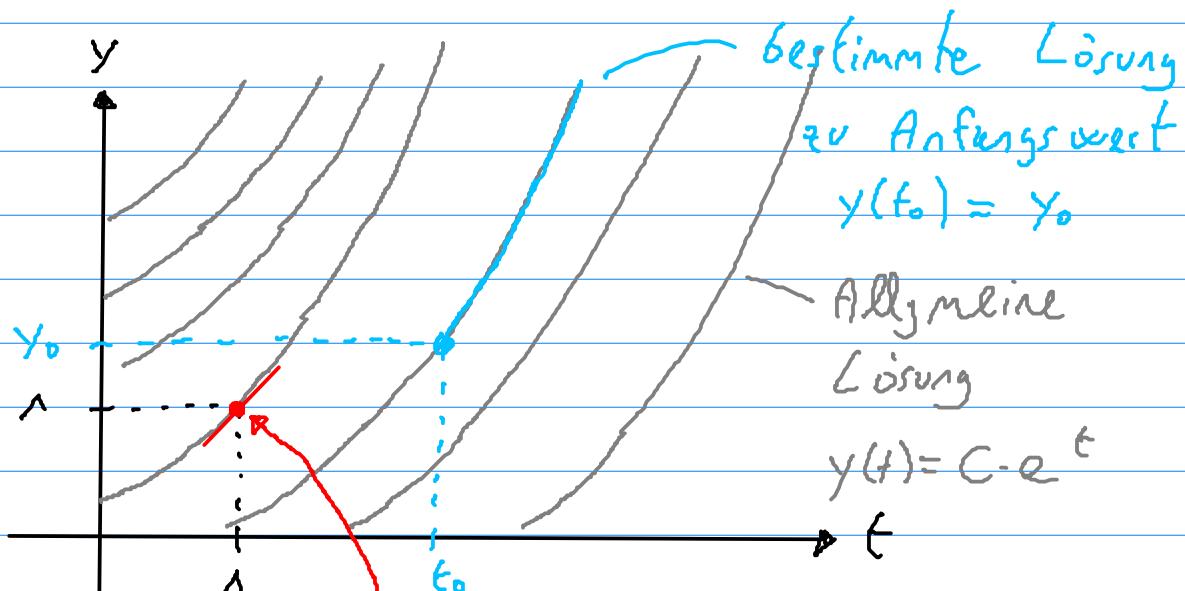
Zur Illustration betrachten wir eine
 (die?!?) einfache skalare DGL

$$\dot{y}(t) = y'(t) = f(t, y(t))$$

sog. rechte Seite der DGL

(skalare)

Eine ^v DGL lässt sich graphisch mit einem
 Richtungsfeld/Vektorfeld darstellen:

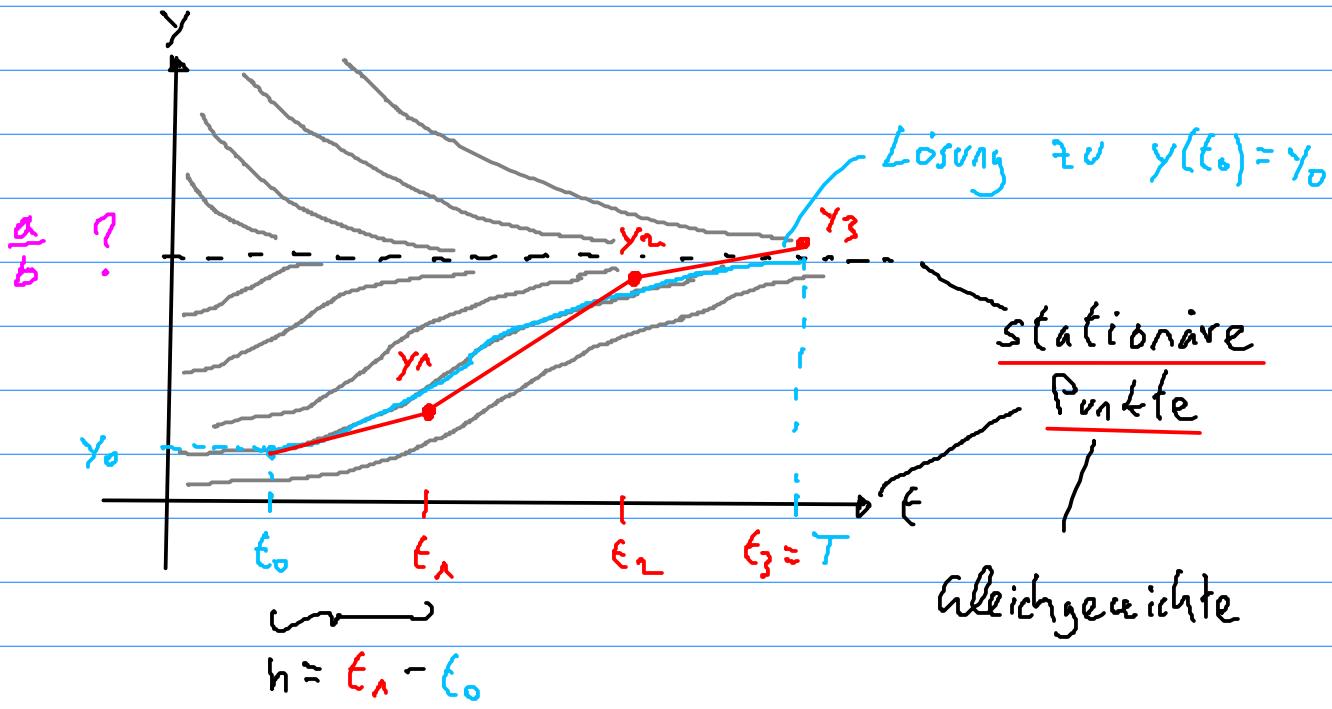


Steigung von $y(t)$ in $(t=1, y=1)$ ist 1

Eine etwas "interessanter" (skalare) DGL:

$$\dot{y}(t) = a \cdot y(t) - b \cdot y^2(t) = (a - b \cdot y(t)) y(t)$$

Graphisch:



Einfache Approximation der Lösung

① Diskretisiere das Zeitintervall $I = [t_0, T]$

$$t_k = t_0 + k \cdot h, \quad k = 0, 1, \dots, N$$

$$\text{mit } h = \frac{T - t_0}{N}$$

② Setze Anfangswert $y_0 = y(t_0)$

③ Berechne für $k = 0, 1, \dots, N-1$

$$y_{k+1} = y_k + h \cdot f(t_k, y_k)$$

Dies ist das Euler Verfahren

oder expliziter Euler

oder Vorwärts Euler

(Euler Forward)

I.A. stellen sich nun folgende Fragen:

(i) Macht es Sinn (approx.) Lösungen zu suchen? (Bedingungen an die rechte Seite $f(t, y(t))$ für Existenz und Eindeutigkeit)

(ii) DGLen höherer Ordnung? (z.B. (1) & (2) oben)

(iii) Konvergiert das Euler Verfahren gegen die exakten Lösungen für $h \rightarrow 0$? Wie schnell?

(iv) Gibt's "bessere" Methoden?

II.2 Grundbegriffe

Def.: Ein skalares Anfangswertproblem (AWP) erster Ordnung:

Finde eine Funktion $y(t)$ einer Variablen (z.B. die Zeit) mit

$$\dot{y}(t) = f(t, y(t)) \quad (\text{gew. DGL 1. Ordnung})$$

auf dem Intervall $I = [t_0, T]$ und

$$y(t_0) = y_0 \quad (\text{Anfangswert (AW)})$$

oder Anfangsbedingung

Oft hat man nicht nur eine DGL, sondern ein ganzes System

Def.: Ein (allgemeines) AWP erster Ordnung:

Finde die Funktionen $y_1(t), \dots, y_n(t)$ einer Variablen (z.B. Zeit) mit

$$\dot{y}_1(t) = f_1(t, y_1(t), \dots, y_n(t))$$

⋮

$$\dot{y}_n(t) = f_n(t, y_1(t), \dots, y_n(t))$$

$\left(\begin{array}{l} n \text{ gew.} \\ \text{DGLen} \\ \text{erster} \\ \text{Ordnung} \end{array} \right)$

auf den Intervall $I = [t_0, T]$ und

$$y_1(t_0) = y_{1,0}$$

:

(n AWs)

$$y_n(t_0) = y_{n,0}$$

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

$$\vec{y}(t_0) = \vec{y}_0$$

wo bei $\vec{y}(t) = \begin{pmatrix} y_1(t) \\ \vdots \\ y_n(t) \end{pmatrix}, \vec{y}_0 = \begin{pmatrix} y_{1,0} \\ \vdots \\ y_{n,0} \end{pmatrix}$

und $\vec{f}(t, \vec{y}) = \begin{pmatrix} f_1(t, y_1(t), \dots, y_n(t)) \\ \vdots \\ f_n(t, y_1(t), \dots, y_n(t)) \end{pmatrix}$

Bsp.: (4) Lineares DGL System

$$\dot{\vec{y}}(t) = A \vec{y}(t), \quad \vec{y} \in \mathbb{R}^n, \quad A \in \mathbb{R}^{n \times n}$$

$$\vec{y}(t_0) = \vec{y}_0 \quad \text{nxn Matrix}$$

Lösung: $\vec{y}(t) = e^{\underbrace{A(t-t_0)}_{\text{Matrix Exp. Fkt.}}} \vec{y}_0$

$$= \left\{ \sum_{i=0}^{\infty} \frac{1}{i!} A^i t^i \right\} \vec{y}_0$$

Def.: Ein (skalares) AWP n-ter Ordnung:

finde die Funktion $y(t)$ einer Variablen (z.B. Zeit) mit

$$y^{(n)}(t) = f(t, y(t), \dot{y}(t), \ddot{y}(t), \dots, y^{(n-1)}(t))$$

(skalare gew. DGL n-ter Ordnung)

auf dem Intervall $I = [t_0, T]$ und

$$y(t_0) = y_0$$

$$\dot{y}(t_0) = \dot{y}_0$$

$$\ddot{y}(t_0) = \ddot{y}_0$$

⋮

$$y^{(n-1)}(t_0) = y^{(n-1)}_0$$

(n AWs für y und die $(n-1)$ -ten Ableitungen)

Bsp.: (S) Newton's Bewegungsgleichungen

$$\text{Masse } m \cdot \ddot{x} = F \quad (\text{ged. DGL 2-ter Ordnung})$$

Beschleunigung Kraft

mit

$$x(t_0) = x_0 \quad (\text{Anfangs-Position})$$

$$\dot{x}(t_0) = \dot{x}_0 (= v_0) \quad (\text{Anfangs-Geschv.})$$

Im Folgenden werden wir nur numerische Methoden für AWP erster Ordnung betrachten. Jedes AWP n -ter Ordnung lässt sich in ein System AWP erster Ordnung umschreiben:

Reduktion auf ein System erster Ordnung

Gegeben eine gek. DGL n -ter Ordnung

$$y^{(n)}(t) = f(t, \dot{y}, \ddot{y}, \dots, y^{(n-1)})$$

Wir definieren

$$z_0(t) = y(t)$$

$$z_1(t) = \dot{y}(t) = \dot{z}_0(t)$$

$$z_2(t) = \ddot{y}(t) = \dot{z}_1(t)$$

⋮

$$z_{n-1}(t) = y^{(n-1)}(t) = \dot{z}_{n-2}(t)$$

Beachte: $\dot{z}_{n-1}(t) = ?$

Dann können wir

$$y^{(n)}(t) = f(t, y(t), \dot{y}(t), \ddot{y}(t), \dots, y^{(n-1)}(t))$$

↓ ↓ ↓ ↓ ↓
 $\left\{ \begin{array}{l} \dot{z}_{n-1}(t) = f(t, z_0(t), z_1(t), z_2(t), \dots, z_{n-1}(t)) \end{array} \right.$
gek. DGL erster Ordnung!

Obige Definitionen sind auch $(n-1)$ gek. DGL erster Ordnung. Insgesamt haben wir also folgendes System von n gek. DGL erster Ordnung

$$\dot{\vec{z}}(t) = \vec{g}(t, \vec{z}(t))$$

wobei

$$\vec{z}(t) = \begin{pmatrix} z_0(t) \\ z_1(t) \\ \vdots \\ z_{n-2}(t) \\ z_{n-1}(t) \end{pmatrix}$$

und

$$\vec{g}(t, \vec{z}(t)) = \begin{pmatrix} z_n(t) \\ z_2(t) \\ \vdots \\ z_{n-1}(t) \\ f(t, z_0(t), z_1(t), \dots, z_{n-1}(t)) \end{pmatrix}$$

Für die AVE ergibt sich

$$\vec{z}(t_0) = \vec{z}_0 = \begin{pmatrix} y_0 \\ \dot{y}_0 \\ \ddot{y}_0 \\ \vdots \\ y_0^{(n-n)} \end{pmatrix}$$

Bsp.: (6) $\ddot{y}(t) = f(t, y(t), \dot{y}(t))$

$$y(t_0) = y_0$$

$$\dot{y}(t_0) = \dot{y}_0$$

Reduktion auf System erster Ordnung:

$$z_0(t) = y(t)$$

$$z_1(t) = \dot{y}(t) = \dot{z}_0(t)$$

$$z_2(t) = \ddot{y}(t) = \dot{z}_1(t) = f(t, z_0(t), z_1(t))$$

Somit

$$\dot{\vec{z}} = \vec{g}(t, \vec{z}(t))$$

wo bei

$$\vec{z} = \begin{pmatrix} z_0(t) \\ z_1(t) \end{pmatrix}, \vec{g}(t, \vec{z}(t)) = \begin{pmatrix} z_1 \\ f(t, z_0(t), z_1(t)) \end{pmatrix}$$

und AVE

$$\vec{z}(t_0) = \vec{z}_0 = \begin{pmatrix} y_0 \\ \dot{y}_0 \end{pmatrix}$$

Genausgleich kann man ein System von n gew. DGLen n -ter Ordnung auf ein System von $m \cdot n$ gew. DGLen erster Ordnung reduzieren.

Def.: Eine gew. DGL heisst autonom, falls die rechte Seite die Form $\vec{f} = \vec{f}(\vec{y}(t))$ hat (anstatt $\vec{f} = \vec{f}(t, \vec{y}(t))$).

Autonomisieren von DGLen

Betrachte folgende gew. DGL

$$\dot{\vec{y}} = \vec{f}(t, \vec{y}(t)), \quad \vec{y} \in \mathbb{R}^n$$

Durch einführen der neuen Variablen

$$\vec{z}(t) = \begin{pmatrix} \vec{y}(t) \\ f \end{pmatrix} = \begin{pmatrix} \vec{z}_1 \\ \vdots \\ \vec{z}_{n+1} \end{pmatrix}$$

und der rechten Seite

$$\vec{g}(\vec{z}(t)) = \begin{pmatrix} \vec{f}(t, \vec{y}(t)) \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} \vec{f}(\vec{z}_{n+1}, \vec{z}_1) \\ \vdots \\ 1 \end{pmatrix}$$

erhält man eine autonome gew. DGL

$$\dot{\vec{z}} = \vec{g}(\vec{z})$$

Bsp.: (+) $\dot{y}(t) = y(t)^2 + t^2$

$$\vec{z}(t) = \begin{pmatrix} z_1(t) \\ z_2(t) \end{pmatrix} = \begin{pmatrix} y(t) \\ t \end{pmatrix}$$

$$\vec{g}(\vec{z}(t)) = \begin{pmatrix} y(t)^2 + t^2 \\ 1 \end{pmatrix} = \begin{pmatrix} z_1(t)^2 + z_2(t)^2 \\ 1 \end{pmatrix}$$

aus $\dot{\vec{z}}(t) = \vec{g}(\vec{z}(t))$ autonom ✓.

Bem.: Allgemein bezeichnet man auch DGLen höherer Ordnung als autonom, wenn die rechte Seite nicht explizit von der Zeit abhängt.

D.h.

$$y^{(n)}(t) = f(y(t), \dot{y}(t), \dots, y^{(n-1)}(t))$$

(anstatt $f = f(t, y(t), \dot{y}(t), \dots, y^{(n-1)}(t))$)

II.2.1 Existenz, Eindeutigkeit und Stabilität

Im folgenden wollen wir uns ein paar theoretische Fragen bez. AWPen anschauen:

(i) Was muss erfüllt sein, dass es überhaupt Lösungen gibt? (Existenz)

(ii) Gibt's mehrere Lösungen? Unter welcher Bedingung gibt es nur eine?
(Eindeutigkeit)

(iii) Wie hängt die Lösung vom AW ab? (Stabilität)

Betrachten wir folgender (allgemeines) AWP

$$\dot{\vec{y}}(t) = \vec{f}(t, \vec{y}(t)) \quad (\text{System von DGLen})$$

$$\vec{y}(t_0) = \vec{y}_0 \quad (\text{AW})$$

wobei

$$\vec{y} : I = [t_0, T] \subset \mathbb{R} \rightarrow D \subset \mathbb{R}^n$$

(Zeit) Intervall

'zustands- oder
Phasenraum'

$$\vec{f} : I \times D \rightarrow \mathbb{R}^n$$

Um (i) und (ii) von oben sicherzustellen, benötigen wir einen etwas stärkeren Begriff der Stetigkeit

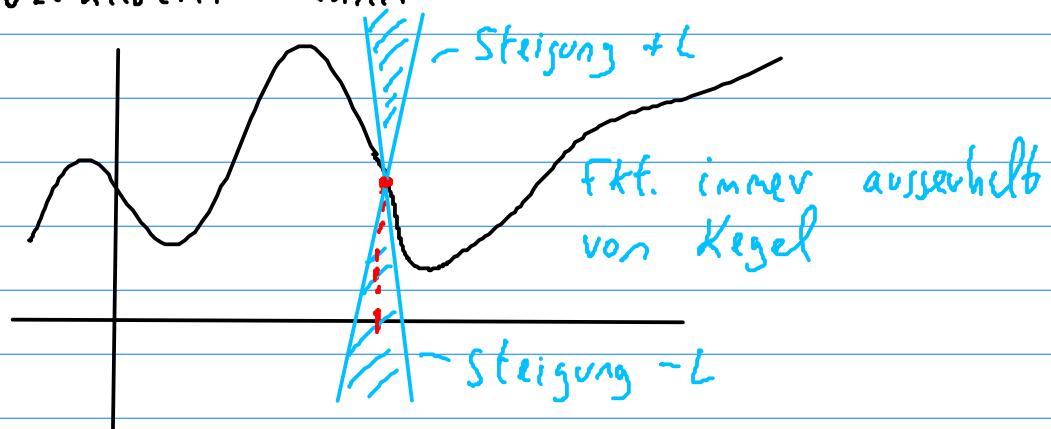
Def.: Eine Funktion $\vec{f}: I \times D \rightarrow \mathbb{R}^n$ ist Lipschitz-stetig in (Variabell) \vec{y} mit der Lipschitz-Konstanten $L > 0$, wenn für alle $t \in I$ und $\vec{y}, \vec{z} \in D$ gilt

$$\|\vec{f}(t, \vec{y}) - \vec{f}(t, \vec{z})\| \leq L \|\vec{y} - \vec{z}\|$$

Hier ist $\|\cdot\|$ eine Norm.

(λ -, euklidische, max-Norm
...)

Bem.: (i) Lipschitz-Stetigkeit beschränkt wie stark eine Funktion um einen Punkt sich verändern kann



(ii) Stetige diff'bare Funktionen sind Lipschitz-stetig: Setze $L = \max_{y \in [a,b]} |f'(y)|$

(iii) Auch nicht diff'bare Funktionen können Lipschitz-stetig sein

$$\text{z.B.: } f(y) = |y|$$

(iv) Manchmal auch Lipschitz-Bedingung genannt
 $\{y \in \mathbb{R} \mid y \geq 0\}$ D.h. nicht stetig diff'bar

Frage: Ist $f(y) = \sqrt{y}$, $y \in \mathbb{R}^+$, überall Lipschitz-stetig? Nein, da $f'(y) = \frac{1}{2\sqrt{y}} \xrightarrow{y \rightarrow 0} \infty$

23.03.20

Der folgende Satz stellt nun (i) und (ii) sicher

Satz II.1: (Picard-Lindelöf)

Sei \tilde{f} stetig in (t, \tilde{y}) und Lipschitz-stetig in \tilde{y} auf $[t_0, t_0 + \delta] \times D$, mit $\delta > 0$ und D eine Umgebung vom AW \tilde{y}_0 .

Dann existiert eine eindeutige Lösung $\tilde{y}(t)$ des AWP

$$\tilde{y}(t) = \tilde{f}(t, \tilde{y}(t)), \quad \tilde{y}(t_0) = \tilde{y}_0$$

für zumindest eine kurze Zeit $[t_0, t_0 + \varepsilon]$, $\varepsilon > 0$.

$$\underline{\text{Bsp.: (8)}} \quad \dot{y}(t) = 2\sqrt{|y(t)|}$$

$$y(0) = 0$$

$$\text{Lösungen: } y(t) = 0$$

$$y(t) = t \cdot |t|$$

Grund: $f(t, y) = 2\sqrt{|y|}$ stetig, aber
nicht Lipschitz stetig in $y=0$!

$$(9) \quad \text{Bsp. (8) mit } y(1) = 1$$

$$\rightsquigarrow y(t) = t \cdot |t| \text{ eindeutige Lösung!}$$

$$(10) \quad \dot{y}(t) = y(t)^2$$

$$y(0) = y_0 > 0$$

$$\text{Lösung: } y(t) = \frac{y_0}{1 - y_0 \cdot t}$$

Aber nur für $0 \leq t < \frac{1}{y_0}$

(... "zumindest eine kurze Zeit" ...)

Grund: Lipschitz-Konstante L unbeschränkt

Der folgende Satz gibt Aufschluss über die eingehende Frage (iii) bzgl. der Abhängigkeit der Lösung von den AOV:

Satz II.2: { Die Funktion $\vec{f}(t, \vec{y})$ sei v Lipschitz-stetig
stetig in (t, \vec{y}) und
mit Lipschitz-Konstanten L in einer
Umgebung der AOVs \vec{y}_0, \vec{z}_0 und
seien $\vec{y}(t), \vec{z}(t)$ die Lösungen des
jeweiligen AOVPs.

Dann gilt für $t \in [t_0, t_0 + \varepsilon]$

$$\|\vec{y}(t) - \vec{z}(t)\| \leq e^{L(t-t_0)} \|\vec{y}_0 - \vec{z}_0\|$$

Bem.: Satz II.2 stellt sicher, dass die Lösungen stetig vom AOV abhängen.

Aber Lösungen können exponentiell in der Zeit auseinander driften

Bsp.: (11) $\dot{y}(t) = \lambda y(t), y(t_0) = y_0 \rightsquigarrow y(t) = y_0 e^{\lambda(t-t_0)}$

$\dot{z}(t) = \lambda z(t), z(t_0) = z_0 \rightsquigarrow z(t) = z_0 \cdot e^{\lambda(t-t_0)}$

$$\Rightarrow y(t) - z(t) = e^{\lambda(t-t_0)} (y_0 - z_0)$$

↑
Vgl. Satz II.2

II.3 Runge-Kutta Verfahren

Betrachten wir das (skalare) AWP

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

$$y(t_0) = y_0$$

Die Lösung können wir formal durch Integration erhalten

$$y(t) = y_0 + \int_{t_0}^t f(\tau, y(\tau)) d\tau$$

Integral-Gleichung... Nicht unbedingt "besser" wie mit der gew. DGL...

Idee: Verwende Quadratur für das Integral der rechten Seite f

Konkret, versuchen wir die Lösung ausgehend von t_0, y_0 in einem Punkt t_1, y_1 , wobei $t_1 = t_0 + h$, mittels Quadratur zu approx.:

$$y(t_1) = y_0 + \int_{t_0}^{t_1} f(z, y(z)) dz$$

$$= y_0 + h \cdot \int_0^1 f(t_0 + hz, y(t_0 + hz)) dz$$

... Grenzen verschoben und skaliert ...

$$\approx y_0 + h \cdot \sum_{i=1}^5 w_i \cdot f(t_0 + h \cdot c_i, y(t_0 + h \cdot c_i))$$

Quadratur Gewichte / Knoten

Problem: $y(t_0 + h \cdot c_i)$ immer noch unbekannt!

→ Müssen auch approx. werden ...

Versuchen wir es mit der \overline{NR} :

$$y(t_1) \approx y_0 + h \cdot f\left(t_0 + \frac{h}{2}, y(t_0 + \frac{h}{2})\right)$$

Wie könnte man eine Approx. von $y(t_0 + \frac{h}{2})$ berechnen? \rightsquigarrow z.B. mit dem expliziten Euler-Verfahren!

$$y(t_0 + \frac{h}{2}) \approx y_0 + \frac{h}{2} f(t_0, y_0) = y_{1h}$$

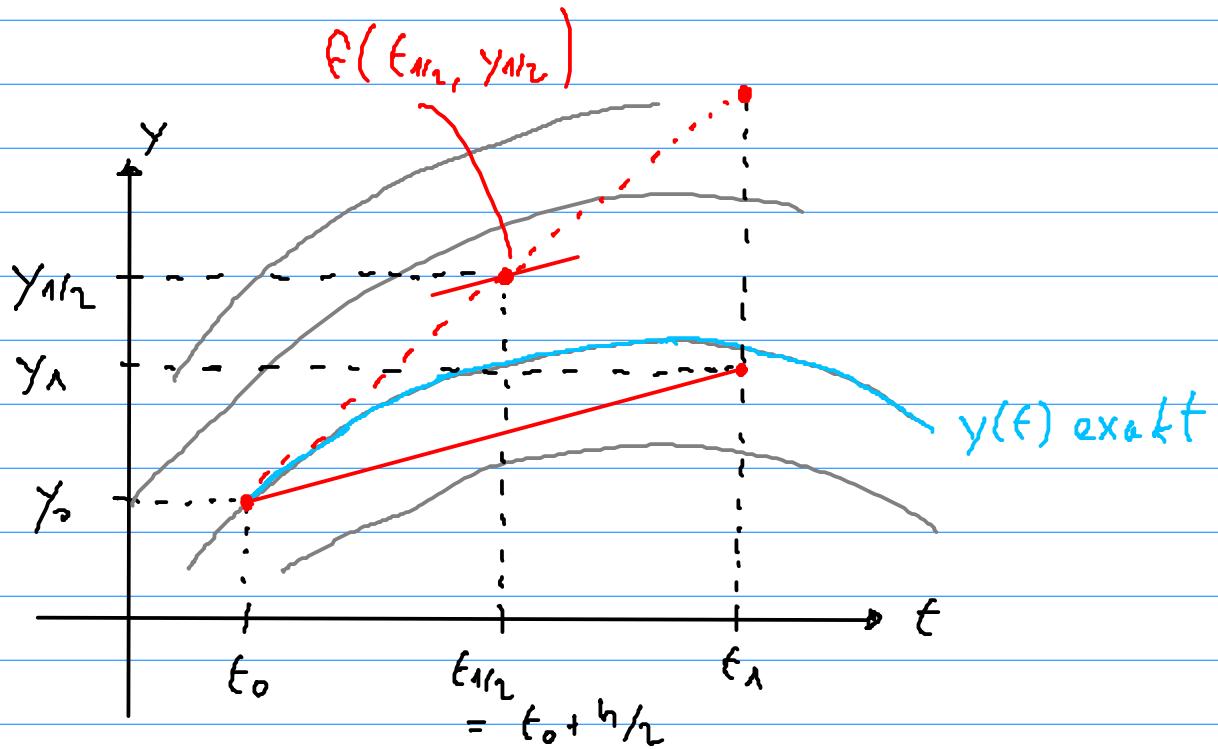
\uparrow

$$y(t_0) = y_0 \text{ AW}$$

Setzen wir dies oben ein erhalten wir

$$y(t_1) \approx y_0 + h \cdot f\left(t_0 + \frac{h}{2}, y_{1h}\right) = y_1$$

Graphisch:



Zusammenfassend erhalten wir folgende Methode für Allgemeine:

$$k_1 = f(t_j, y_j)$$

$$k_2 = f(t_j + h/2, y_j + h/2 \cdot k_1)$$

$$y_{j+1} = y_j + h \cdot k_2$$

für $j = 0, 1, \dots, N-1$. Dieses Verfahren ist bekannt als die verbesserte Polygonzug-Methode von Euler.

Oder auch: modifiziertes Euler Verfahren
explizite Mittelpunkts-Methode

Eine andere Möglichkeit wäre als QR
die TR zu verwenden

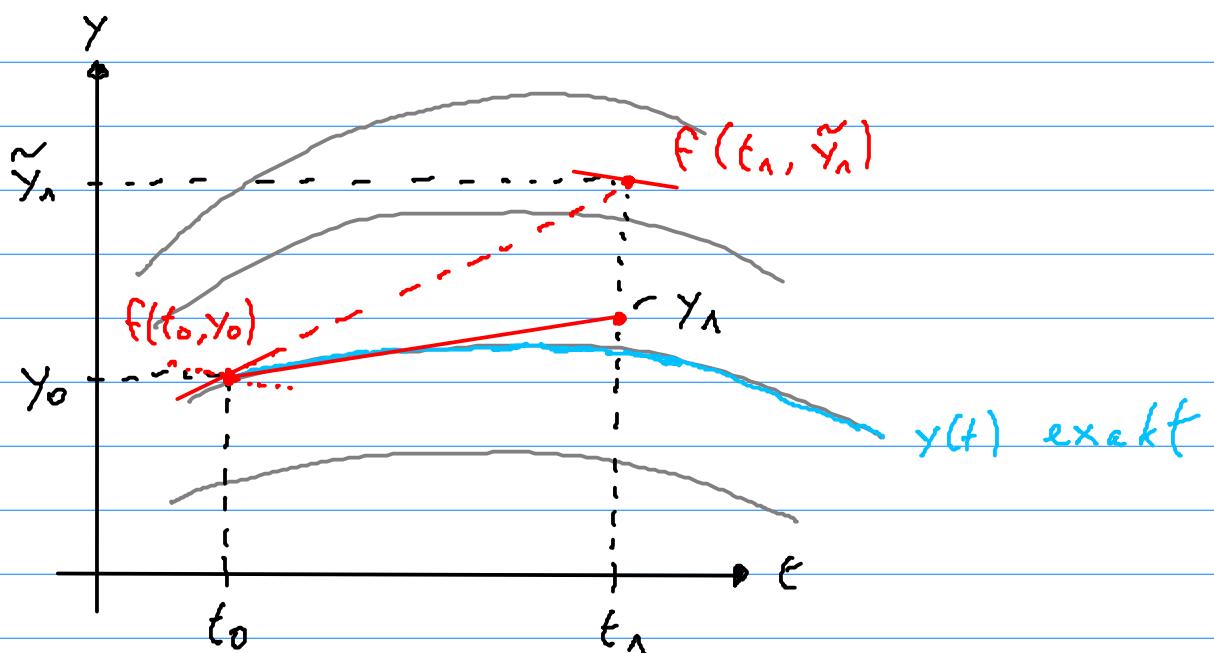
$$y(t_1) \approx y_0 + \frac{h}{2} \left(f(t_0, y_0) + f(t_1, y(t_1)) \right)$$

?

Wie vorhin, approx. wir $y(t_1)$ mit dem
Euler-Verfahren

$$y(t_1) \approx y_0 + h \cdot f(t_0, y_0) = \tilde{y}_1$$

Graphisch:



Wir erhalten folgendes Verfahren

$$k_1 = f(t_j, y_j)$$

$$k_2 = f(t_j + h, y_j + h \cdot k_1)$$

$$y_{j+1} = y_j + \frac{h}{2} (k_1 + k_2)$$

für $j = 0, 1, \dots, N-1$. Dies ist das Verfahren von Heun.

Oder auch explizite Trapez-Methode.

Alle Verfahren die wir bis jetzt kennengelernt haben sind Teil einer grossen Familie von Einschrittverfahren (ESV):

Runge-Kutta Verfahren

Ein s-stufiges Runge-Kutta (Einschritt-)Verfahren (RK-ESV) ist definiert durch

$$y_{j+1} = y_j + h \cdot \sum_{i=1}^s b_i \cdot k_i$$

wobei die Stufen/Steigungen

$$k_i = f(t_j + c_i \cdot h, y_j + h \cdot \sum_{\ell=1}^s a_{i\ell} \cdot k_\ell)$$

sind. Weiter nennt man

s ... Anzahl Stufen

c_i ... Knoten

b_i ... Gewichte

$a_{i\ell}$... Runge-Kutta Matrix/
Koeffizienten

RK Verfahren schreibt man am besten
in einem sog. Butcher-Tableau (BT)

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline b_1 & b_1 & b_2 & \cdots & b_s \end{array} = \begin{array}{c|c} \vec{c} & A \\ \hline \vec{b}^T & q \end{array}$$

transponiert

Bsp.: (12) BT der verbesserten Polygonzug-Methode von Euler

$$\begin{array}{c|cc} \vec{c} & 0 & 0 \\ \hline 0 & 0 & 0 \\ 1/2 & 1/2 & 0 \end{array}$$

2 Stufen { \vec{c} } \vec{A} die Nullen der RK-Matrix A schreibt

$s=2$ \vec{B}^T men (oft) nicht

$$k_1 = f\left(\epsilon_j + c_1 \cdot h, y_j + h \cdot (a_{11} \cdot k_1 + a_{12} \cdot k_2)\right)$$

$$= f(\epsilon_j, y_j)$$

$$k_2 = f\left(\epsilon_j + \frac{1}{2}h, y_j + h \cdot (a_{21} \cdot k_1 + a_{22} \cdot k_2)\right)$$

$$= f\left(\epsilon_j + \frac{h}{2}, y_j + \frac{h}{2} \cdot k_1\right)$$

$$\begin{aligned} y_{j+1} &= y_j + h \cdot (b_1 \cdot k_1 + b_2 \cdot k_2) \\ &= y_j + h \cdot k_2 \end{aligned}$$

(13) BT der Methode von Heun

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \end{array}$$

(14) BT des Euler-Verfahrens

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

Obwohl wir bisher nur skalare AWPe betrachtet haben, kann man die RK-Verfahren sehr einfach auf allgemeine AWPe (also Systeme) verallgemeinern:

$$\begin{array}{ccc} y & \rightsquigarrow & \vec{y} \\ f & \rightsquigarrow & \vec{f} \end{array}$$

Daneben ist die Def. der RK-Verfahren sehr allgemein. Sie umfasst auch sog. implizite Verfahren

Bsp.: (AS) Implizite Mittelpunkts-Methode

$$\begin{array}{c|c} h/2 & h/2 \\ \hline & h \end{array}$$

$$\underline{k_n} = f(t_j + \frac{h}{2}, y_j + \frac{h}{2} \cdot \underline{k_n})$$

$$y_{j+1} = y_j + h \cdot k_n$$

implizite Def. von k_n

d.h. man muss eventuell ein i.A.
nicht-lineares Gleichungssystem lösen!

Implizite Methoden benötigt man bei sog. steifen Problemen Kap. IV.

Explizite RK-Verfahren haben ein BT der Form

$$\begin{array}{c|cccc} c_1 & 0 & 0 & \cdots & 0 \\ c_2 & a_{2,1} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ c_s & a_{s,1} & a_{s,2} & \cdots & a_{s,s-1} & 0 \\ \hline b_1 & b_1 & b_2 & & b_{s-1} & b_s \end{array}$$

D.h. A ist eine Untere-Dreiecksmatrix mit Nullen auf der Diagonalen.

Das wohl bekannteste RK-Verfahren ist das sog. klassische RK-Verfahren

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & 1/6 & 2/6 & 2/6 & 1/6 \end{array} \quad \text{Graphisch } \rightarrow \text{Slides}$$

Oft wird es auch nur als DAS RK-Verfahren oder RK4 bezeichnet.

4 Stufen

II.4 Fehlerbetrachtungen für ESV

Hier wollen wir die Genauigkeit von ESV untersuchen. Dazu betrachten wir ein allgemeines ESV der Form

$$y_{j+1} = y_j + h \cdot \phi(t_j, y_j, h)$$

wobei ϕ die sog. Verfahrens- oder Integrationsfunktion ist.

Bem.: (i) Bei RK-ESV ist

$$\phi(t_j, y_j, h) = \sum_{i=1}^s b_i \cdot k_i$$

(ii) Bei einem expliziten ESV kann man ϕ einfach durch einsetzen berechnen.

Bei einem impliziten ESV muss man um ϕ zu berechnen i.A. nicht-lineare Gleichungssysteme lösen.

Im folgenden betrachten wir das (skalare)

AIVP

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

$$y(t_0) = y_0 \quad (\text{AV})$$

auf dem Intervall $[t_0, T]$.

Anfangs- $\overset{\wedge}{\text{End-}}\text{-Zeit}$

Mit $y(t)$ bezeichnen wir die exakte Lösung.

Wir approx. die Lösung durch diskretisieren des Zeit-Intervalls in N Teil-Intervalle

$$t_j = t_0 + j \cdot h$$

mit der/dem Schrittweite/Zeitschritt

$$h = \frac{T - t_0}{N}$$

Die approx. Lösung zur Zeit t_j bezeichnen wir mit y_j .

Also

$$y(t_j) \approx y_j$$

Def.: Der globale Diskretisierungsfehler (GDF) zur Zeit t_j ist definiert durch

$$\epsilon_j = y(t_j) - \underline{y}_j$$

exakte approx. Lösung zur
Zeit t_j

D.h. ϵ_j ist der Fehler (die Differenz zwischen der exakten und approx. Lösung) nach j Schritten.

Def.: Ein ESV heisst konvergent von der Ordnung p (oder hat Konvergenz-Ordnung (KO) p), wenn gilt

$$\epsilon = \max_{j=0, \dots, N} \underbrace{|y(t_j) - \underline{y}_j|}_{\epsilon_j} = O(h^p)$$

für h klein genug.

Bsp.: (16) Empirische KO der bisher
getroffenen RK-ESV
 ↳ Slides

$$\text{Euler} \sim O(h^1)$$

$$\text{verb. Euler} \sim O(h^2)$$

$$\text{Heun} \sim O(h^2)$$

$$\text{RK4} \sim O(h^4)$$

Um die KO einer ESV theoretisch zu untersuchen benötigen wir:

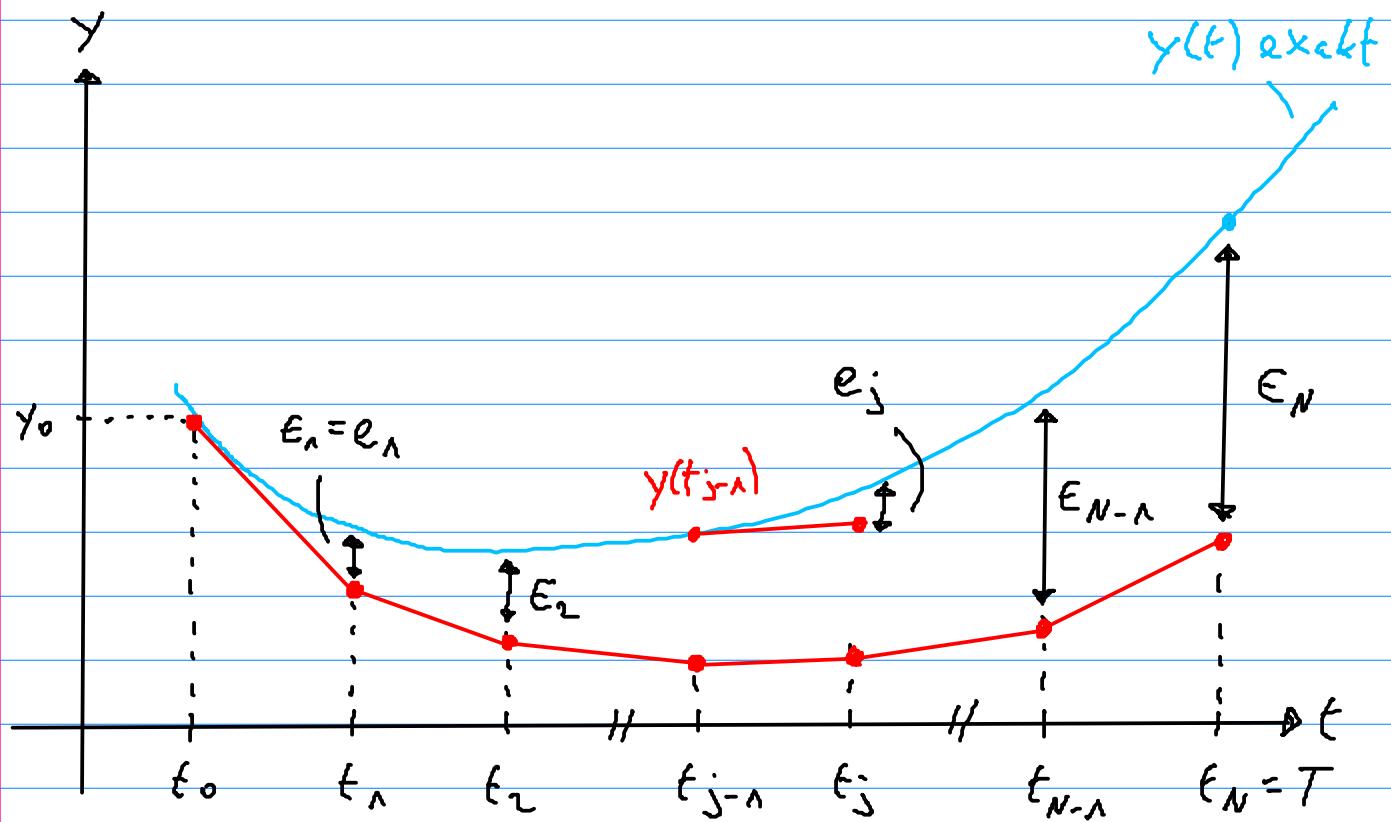
Def.: Der lokale Diskretisierungsfehler (LDF) zur Zeit t_j ist definiert durch

$$e_j = y(t_j) - \left(y(t_{j-1}) + h \cdot \phi(t_{j-1}, y(t_{j-1}), h) \right)$$

↖ ↗
exakte Lösung bei t_{j-1}

D.h. e_j ist der Fehler bei t_j nach einem Schritt ausgehend von der exakten Lösung bei t_{j-1} .

Graphisch:



Def.: Der Konsistenzfehler (KF) zur Zeit t_j ist definiert durch

$$\begin{aligned} \tau_j &= \frac{e_j}{h} \\ &= \frac{y(t_j) - y(t_{j-1})}{h} - \phi(t_{j-1}, y(t_{j-1}), h) \end{aligned}$$

Def.: Ein ESV heisst konsistent von der Ordnung p (oder hat Konsistenzordnung p), wenn gilt

$$\tau = \max_{j=0, \dots, N} |\tau_j| = \mathcal{O}(h^p)$$

für h klein genug.

Ein ESV heisst konsistent falls $p \geq 1$.

LDF, KF und Konsistenzordnung sind lokale Größen. Diese kann man (relativ) einfach mittels Taylor-Entwicklungen bestimmen:

$$\begin{aligned}\tau_{j+N} &= \frac{\overbrace{y(t_j+h)} - y(t_j)}{h} - \phi(t_j, y(t_j), h) \\ &= \frac{y(t_j) + h \cdot \dot{y}(t_j) + \frac{h^2}{2} \ddot{y}(t_j) + \dots - y(t_j)}{h} \\ &\quad - \left(\phi(t_j, y(t_j), 0) + h \cdot \phi'(t_j, y(t_j), 0) + \frac{h^2}{2} \phi''(t_j, y(t_j), 0) + \dots \right) \\ &= \dot{y}(t_j) - \phi(t_j, y(t_j), 0) \\ &\quad + \frac{h}{2} \left(\ddot{y}(t_j) - 2 \cdot \phi'(t_j, y(t_j), 0) \right) \\ &\quad + \frac{h^2}{6} \left(\ddot{y}(t_j) - 3 \cdot \phi''(t_j, y(t_j), 0) \right) \\ &\quad + \dots + \mathcal{O}(h^p) + \dots\end{aligned}$$

Berechnen wir zunächst die Ableitungen der Lösung

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

$$\ddot{y}(t) = \frac{d}{dt} f(t, y(t))$$

$$\text{Ketten-Regel!} = \frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t)) \cdot \dot{y}(t)$$

" $f(t, y(t))$ DGL!"

$$= \frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t)) \cdot f(t, y(t))$$

$$\ddot{y}(t) = \frac{d}{dt} \left(\frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t)) \cdot f(t, y(t)) \right)$$

$$= \frac{\partial^2 f}{\partial t^2}(t, y(t)) + \frac{\partial^2 f}{\partial y \partial t}(t, y(t)) \cdot \dot{y}(t) = f(t, y(t))$$

$$\frac{\partial^2 f}{\partial t^2} = \frac{\partial^2 f}{\partial y \partial t} + \left(\frac{\partial^2 f}{\partial t \partial y}(t, y(t)) + \frac{\partial^2 f}{\partial y^2}(t, y(t)) \cdot \dot{y}(t) \right) \cdot f(t, y(t))$$

$$+ \frac{\partial f}{\partial y}(t, y(t)) \cdot \left(\frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t)) \cdot \dot{y}(t) \right)$$

" $f(t, y(t))$

$$= \frac{\partial^2 f}{\partial t^2}(t, y(t)) + 2 \cdot \frac{\partial^2 f}{\partial t \partial y}(t, y(t)) \cdot f(t, y(t))$$

$$+ \frac{\partial^2 f}{\partial y^2}(t, y(t)) \cdot f(t, y(t))^2 + \frac{\partial f}{\partial t}(t, y(t)) \cdot \frac{\partial f}{\partial y}(t, y(t))$$

$$+ \left(\frac{\partial f}{\partial y}(t, y(t)) \right)^2 \cdot f(t, y(t))$$

$$y^{(k)}(t) = \left(\frac{\partial}{\partial t} + f(t, y(t)) \frac{\partial}{\partial y} \right)^{k-1} f(t, y(t))$$

Zur Berechnung der Entwicklung der Verfahrens-Funktion benötigt man die zweidimensionale Taylor-Reihe:

$$f(t + \Delta t, y + \Delta y) = f(t, y)$$

$$\begin{array}{l} \text{~\textcolor{red}{h}} \\ \uparrow \quad \uparrow \\ + \frac{\partial f}{\partial t}(t, y) \cdot \Delta t + \frac{\partial f}{\partial y}(t, y) \cdot \Delta y \end{array}$$

$$+ \frac{1}{2} \frac{\partial^2 f}{\partial t^2}(t, y) \cdot \Delta t^2 + \frac{\partial^2 f}{\partial t \partial y}(t, y) \cdot \Delta t \cdot \Delta y + \frac{1}{2} \frac{\partial^2 f}{\partial y^2}(t, y) \cdot \Delta y^2$$

$$+ \dots$$

Am besten ein paar Beispiele

Bsp.: (1+) Euler-Verfahren

$$\phi(t_j, y(t_j), h) = f(t_j, y(t_j))$$

$$\rightsquigarrow \phi(t_j, y(t_j), 0) = f(t_j, y(t_j)) = \dot{y}(t_j)$$

$$\phi'(t_j, y(t_j), 0) = ? \quad \text{0 ist ja konstant für geg. } t_j, y(t_j)$$

Und damit Konsistenzordnung

$$p = ? \wedge$$

D.h. $\varepsilon = \mathcal{O}(h^?)$ wie gemessen
in Bsp. (1b)!

(18) Verbesserter Euler

$$k_1 = f(t_j, y(t_j))$$

$$k_2 = f(t_j + h/2, y(t_j) + \frac{h}{2} k_1)$$

$$y_{j+1} = y(t_j) + h \cdot k_2$$

Entwickeln der Verfahrens-Funktion

$$\phi(t_j, y(t_j), h) = f(t_j + \frac{h}{2}, y(t_j) + \frac{h}{2} k_1)$$

$$= f(t_j, y(t_j)) \leftarrow \phi(t_j, y(t_j), 0)$$

$$\phi'(t_j, y(t_j), 0) \cdot h \left\{ + \frac{\partial f}{\partial t}(t_j, y(t_j)) \cdot \frac{h}{2} + \frac{\partial f}{\partial y}(t_j, y(t_j)) \cdot \frac{h}{2} k_1 \right.$$

$$\phi''(t_j, y(t_j), 0) \cdot \frac{h^2}{2} \left\{ + \frac{1}{2} \frac{\partial^2 f}{\partial t^2}(t_j, y(t_j)) \cdot \left(\frac{h}{2}\right)^2 + \frac{\partial^2 f}{\partial t \partial y}(t_j, y(t_j)) \cdot \left(\frac{h}{2}\right)^2 \cdot k_1 + \frac{1}{2} \frac{\partial^2 f}{\partial y^2}(t_j, y(t_j)) \cdot \left(\frac{h}{2} k_1\right)^2 \right.$$

+ ...

Durch Vergleich

$$\phi(t_j, y(t_j), 0) = f(t_j, y(t_j)) = \dot{y}(t_j) \quad \checkmark$$

$$\begin{aligned} 2. \phi'(t_j, y(t_j), 0) &= \left(\frac{\partial f}{\partial t}(t_j, y(t_j)) + \frac{\partial f}{\partial y}(t_j, y(t_j)) \cdot f(t_j, y(t_j)) \right) \\ &= \ddot{y}(t_j) \quad \checkmark \end{aligned}$$

$$3. \phi''(t_j, y(t_j), 0) = \dots \quad \times$$

Und damit Konsistenzordnung $p=2$

D.h. $\varepsilon = \mathcal{O}(h^2)$ wie gemessen
in Bsp. (16)!

(13) Allgemeines 3-stufiges explizites RK

c_1			
c_2	a_{21}		
c_3	a_{31}	a_{32}	
	b_1	b_2	b_3

Durch Entwicklung der Verfahrensfunktion
und Vergleich mit der exakten
Entwicklung erhält man folgende
Bedingungen:

$$b_1 + b_2 + b_3 = 1$$

1. Ordnung

$$b_1 c_1 + b_2 c_2 + b_3 c_3 = 1/2$$

2. Ordnung

$$a_{21} b_2 + (a_{31} + a_{32}) b_3 = 1/2$$

$$b_1 c_1^2 + b_2 c_2^2 + b_3 c_3^2 = 1/3$$

$$a_{21} b_2 c_2 + (a_{31} + a_{32}) b_3 c_3 = 1/3$$

$$a_{21}^2 b_2 + (a_{31} + a_{32})^2 b_3 = 1/3$$

$$a_{21} b_2 c_1 + a_{31} b_3 c_1 + a_{32} b_3 c_2 = 1/6$$

$$a_{21} a_{32} b_3 = 1/6$$

8 nicht lineare Gleichungen für 3 Variablen!

Fazit: die lokalen Fehlermasse (LDF, KF, Konsistenzordnung) lassen sich mittels Taylor-Reihenentwicklungen abschätzen

(Dabei wird natürlich immer stillschweigend vorausgesetzt, dass die rechte Seite der DGL genügend glatt ist! $\rightarrow f$)

Aber eigentlich sind wir ja an globalen Fehlergrößen (GDF & KO) interessiert.

Bsp. (16), (17) und (18) deuten bereits an, dass zwischen den lokalen und globalen Größen ein Zusammenhang besteht. Der folgende Satz untermauert dies:

Satz II.3: Falls die rechte Seite der DGL
 $\vec{f}(t, \vec{y})$ und die Verfahrens-funktion
 $\vec{\phi}(t, \vec{y}, t)$ Lipschitz-stetig in \vec{y} sind, dann
gilt für das ESV folgende (globale)
Fehlerabschätzung

$$\begin{aligned} \epsilon &= \max_{j=0, \dots, N} \|\vec{y}(t_j) - \vec{y}_j\| \\ &\leq \left(\|\vec{y}(t_0) - \vec{y}_0\| + \sum_{j=1}^N \|\vec{e}_j\| \right) \cdot e^{\tilde{L}(t_N - t_0)} \end{aligned}$$

AW fehler

fehler in jedem Schritt
summieren sich
schlimmsterfalls

wobei \tilde{L} die Lipschitz-Konstante der
Verfahrens-funktion $\vec{\phi}$ ist.

Obige Aussage impliziert

(für f glatt
genug $\vec{\phi}$) Konsistenz der Ordnung p
 \Rightarrow Konvergenz der Ordnung p

Bem.: (i) Satz II.3 macht eine Aussage über die
Stabilität von ESV, d.h. eine kontrollierte
fehlerfortpflanzung (AW + LDF $\vec{\phi}$)

(ii) \tilde{L} hängt über $\vec{\phi}$ mit der rechten Seite
 \vec{f} zusammen

Betrachten wir den GDF zur Endzeit $T=t_N$:

$$\begin{aligned}
 |\epsilon_N| &= |y(t_N) - y_N| \\
 &= \left| \underbrace{y(t_{N-1})}_{\text{Def.}} + h \cdot \frac{y(t_N) - y(t_{N-1})}{h} \right| + \text{O Trick} \\
 &\quad + \text{O Trick} \\
 &= \left| h \left(\phi(t_{N-1}, y(t_{N-1}), h) - \phi(t_{N-1}, y(t_{N-1}), h) \right) \right. \\
 &\quad \left. - \left(y_{N-1} + h \cdot \phi(t_{N-1}, y_{N-1}, h) \right) \right| \\
 &\quad \text{y}_N
 \end{aligned}$$

$$\begin{aligned}
 \Delta\text{-UG} \\
 &\leq \left| y(t_{N-1}) - y_{N-1} \right| = \epsilon_{N-1} \\
 &\quad + h \cdot \left| \frac{y(t_N) - y(t_{N-1})}{h} - \phi(t_{N-1}, y(t_{N-1}), h) \right| \\
 &\quad \text{=} h \epsilon_N \\
 &\quad \text{=} R_N
 \end{aligned}$$

$$\begin{aligned}
 &+ h \cdot \left| \phi(t_{N-1}, y(t_{N-1}), h) - \phi(t_{N-1}, y_{N-1}, h) \right| \\
 &\leq h \tilde{\lambda} |y(t_{N-1}) - y_{N-1}| = h \tilde{\lambda} |\epsilon_{N-1}| \\
 &\quad \text{Lipschitz-Stetigkeit von } \phi
 \end{aligned}$$

$$\leq (\lambda + h \tilde{\lambda}) |\epsilon_{N-1}| + |\epsilon_N|$$

$$\begin{aligned}
 &\text{Rekursiv} \\
 &\text{einsetzen} \leq (\lambda + h \tilde{\lambda}) \left((\lambda + h \tilde{\lambda}) |\epsilon_{N-2}| + |\epsilon_{N-1}| \right) + |\epsilon_N|
 \end{aligned}$$

⋮

$$\begin{aligned}
 &\leq (\lambda + h \tilde{\lambda})^N |\epsilon_0| + \sum_{j=1}^N (\lambda + h \tilde{\lambda})^{N-j} |\epsilon_j| \\
 &\leq (\lambda + h \tilde{\lambda})^N
 \end{aligned}$$

$$\leq \left(|\epsilon_0| + \sum_{j=n}^N |\epsilon_j| \right) (\underbrace{1 + h\tilde{\epsilon}}_{\leq e^{h\tilde{\epsilon}}})^N$$

$$\leq \left(|\epsilon_0| + \sum_{j=n}^N |\epsilon_j| \right) e^{h\tilde{\epsilon} N}$$

$h = \frac{T-t_0}{N}$

$$= \left(|\epsilon_0| + \sum_{j=n}^N |\epsilon_j| \right) \cdot e^{\tilde{\epsilon}(T-t_0)}$$

$$= \left(|\underset{\text{AW Fehler}}{y(t_0)} - y_0| + \sum_{j=n}^N |\epsilon_j| \right) \cdot e^{\tilde{\epsilon}(T-t_0)}$$

\uparrow Fehler in jedem Schritt

Und damit hätten wir Satz II.3 für den skalaren Fall begründet!