1 Auflösen einer impliziten Iterationsvorschrift

Ein implizites Zeitschrittverfahren der Form

$$\begin{aligned} t^{\text{neu}} &= t^{\text{alt}} + \Delta t, \\ \mathbf{u}^{\text{neu}} &= \mathbf{u}^{\text{alt}} + \Delta t \cdot \Phi(t^{\text{alt}}, t^{\text{neu}}; \mathbf{u}^{\text{alt}}, \mathbf{u}^{\text{neu}}), \end{aligned}$$

erfordert zur Bestimmung von $\mathbf{u}^{\text{neu}}$ am nächsten Zeitschritt im Allgemeinen die Lösung eines nichtlinearen Gleichungssystems. Zum Beispiel ist der zu berechnende Wert im Falle des impliziten Eulerverfahren die Lösung $\mathbf{x}$ der Gleichung

$$\mathbf{x} = \mathbf{u}^{\text{alt}} + \Delta t \cdot \Phi(t^{\text{alt}}, t^{\text{neu}}; \mathbf{u}^{\text{alt}}, \mathbf{x}) = \mathbf{u}^{\text{alt}} + \Delta t \cdot \mathbf{f}(t^{\text{neu}}, \mathbf{x})$$

und dies ist im Falle einer nichtlinearen rechten Seite $\mathbf{f}$ eine nichtlineare Gleichung. Um eine solche Lösung zu ermitteln bieten sich zwei Vorgehensweisen an.

1.1 Banachscher Fixpunktsatz

Fasst man das nichtlineare Problem als ein Fixpunktproblem auf,

$$\mathbf{h}(\mathbf{x}) := \mathbf{u}^{\text{alt}} + \Delta t \cdot \Phi(t^{\text{alt}}, t^{\text{neu}}; \mathbf{u}^{\text{alt}}, \mathbf{x}) \quad \Rightarrow \quad \mathbf{h}(\mathbf{x}) = \mathbf{x},$$

so kann man den Banachschen Fixpunktsatz verwenden.

Definition 1 (Kontraktion)
Eine Lipschitz-stetige Funktion $\mathbf{h} : \mathbb{R}^d \to \mathbb{R}^d$ mit Lipschitz-Konstante $L < 1$ nennt man eine *Kontraktion*, da die Abstände zwischen zwei Punkten bei Anwendung der Abbildung $\mathbf{h}$ wegen $\|\mathbf{h}(\mathbf{x}) - \mathbf{h}(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$ immer kleiner werden.

Der folgende Satz liefert nun sowohl die Existenz als auch die Eindeutigkeit einer Lösung für Kontraktionen.

Satz 2 (Banachscher Fixpunktsatz)
Jede Kontraktion $\mathbf{h} : \mathbb{R}^d \to \mathbb{R}^d$ besitzt genau einen Fixpunkt, d.h. es gibt ein eindeutiges $\mathbf{x} \in \mathbb{R}^d$ mit $\mathbf{h}(\mathbf{x}) = \mathbf{x}$, und dieser lässt sich ermitteln als Grenzwert der Folge $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots \to \mathbf{x}$ definiert als

$$\begin{aligned} \mathbf{x}^{(0)} &:= \mathbf{x}_0, \quad \text{(beliebig)} \\ \mathbf{x}^{(k)} &:= \mathbf{h}(\mathbf{x}^{(k-1)}), \quad k = 1, 2, 3, \ldots. \end{aligned}$$

Ist die Verfahrensfunktion Φ Lipschitz-stetig, so findet man für $\Delta t L < 1$ wegen

$$\begin{aligned}\|\mathbf{h}(\mathbf{x}) - \mathbf{h}(\mathbf{y})\| &= \|\mathbf{u}^{\text{alt}} + \Delta t \cdot \Phi(t^{\text{alt}}, t^{\text{neu}}; \mathbf{u}^{\text{alt}}, \mathbf{x}) - \mathbf{u}^{\text{alt}} - \Delta t \cdot \Phi(t^{\text{alt}}, t^{\text{neu}}; \mathbf{u}^{\text{alt}}, \mathbf{y})\| \\ &= \Delta t \cdot \|\Phi(t^{\text{alt}}, t^{\text{neu}}; \mathbf{u}^{\text{alt}}, \mathbf{x}) - \Phi(t^{\text{alt}}, t^{\text{neu}}; \mathbf{u}^{\text{alt}}, \mathbf{y})\| \leq \Delta t L \|\mathbf{x} - \mathbf{y}\|\end{aligned}$$

eine Kontraktion und der Banachsche Fixpunktsatz garantiert eine Lösung. In der Praxis konvergiert die zur Gewinnung benötigte Folge jedoch nur sehr langsam.

1.2 Newton-Verfahren

Fasst man das nichtlineare Problem als ein Nullstellenproblem auf,

$$\mathbf{g}(\mathbf{x}) := \mathbf{u}^{\text{alt}} + \Delta t \cdot \Phi(t^{\text{alt}}, t^{\text{neu}}; \mathbf{u}^{\text{alt}}, \mathbf{x}) - \mathbf{x} \quad \Rightarrow \quad \mathbf{g}(\mathbf{x}) = \mathbf{0},$$

so kann man das Verfahren von Newton verwenden. Für die Abbildung $\mathbf{g} : \mathbb{R}^d \to \mathbb{R}^d$ sucht man eine Nullstelle, indem man ausgehend von einer Startschätzung $\mathbf{x}^{(0)}$ die lineare Approximation (sofern möglich) betrachtet,

$$\mathbf{g}(\mathbf{x}) = \mathbf{g}(\mathbf{x}^{(0)}) + \mathbf{J}_{\mathbf{g}}(\mathbf{x}^{(0)}) \, (\mathbf{x} - \mathbf{x}^{(0)}) + o(\|\mathbf{x} - \mathbf{x}^{(0)}\|),$$

mit der Jacobi-Matrix $\mathbf{J}_{\mathbf{g}}(\mathbf{x}^{(0)}) \in \mathbb{R}^{d \times d}$. Vernachlässigt man die Terme höherer Ordnung, so lässt sich die folgende Iterierte sinnvollerweise als $\mathbf{g}(\mathbf{x}^{(1)}) \approx \mathbf{0}$ fordern und man findet

$$\begin{aligned}\mathbf{0} \approx \mathbf{g}(\mathbf{x}^{(1)}) &\approx \mathbf{g}(\mathbf{x}^{(0)}) + \mathbf{J}_{\mathbf{g}}(\mathbf{x}^{(0)}) \, (\mathbf{x}^{(1)} - \mathbf{x}^{(0)}) \\ \Rightarrow \quad \mathbf{x}^{(1)} &= \mathbf{x}^{(0)} - (\mathbf{J}_{\mathbf{g}}(\mathbf{x}^{(0)}))^{-1} \, \mathbf{g}(\mathbf{x}^{(0)}).\end{aligned}$$

Dieses Vorgehen kann man nun wiederholen.

Definition 3 (Newton-Verfahren)
Sei die Abbildung $\mathbf{g} : \mathbb{R}^d \to \mathbb{R}^d$ differenzierbar. Die Newton-Iteration zur Approximation einer Nullstelle $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ lautet mit einer Startschätzung $\mathbf{x}_0 \in \mathbb{R}^d$:

$$\begin{aligned}&\mathbf{x}^{(0)} := \mathbf{x}_0, \\ &\mathbf{x}^{(k)} := \mathbf{x}^{(k-1)} - (\mathbf{J}_{\mathbf{g}}(\mathbf{x}^{(k-1)}))^{-1} \, \mathbf{g}(\mathbf{x}^{(k-1)}), \quad k = 1, 2, 3, \ldots.\end{aligned}$$

In jedem Schritt des Newton-Verfahrens wird folglich die Jacobi-Matrix für der aktuelle Iterierte berechnet und diese Matrix muss invertiert werden.

Gewöhnlich fasst man diesen Schritt als die Berechnung einer Korrektur $\mathbf{c} \in \mathbb{R}^d$ auf, die durch das folgende Vorgehen beschrieben wird:

$$\mathbf{J_g}(\mathbf{x}^{(k-1)})\,\mathbf{c} = -\mathbf{g}(\mathbf{x}^{(k-1)}), \qquad \mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \mathbf{c}, \qquad k = 1, 2, \ldots.$$

Als Startwert des Newton-Verfahrens ist es günstig eine möglichst gut Schätzung zu verwenden. Hier bietet sich der Wert am alten Zeitschritt, $\mathbf{x}_0 := \mathbf{u}^{\text{alt}}$, an, da man davon ausgehen kann, dass sich für kleine Zeitschritt die Werte nicht groß zwischen den Zeitpunkten unterscheiden.
Für die Jacobi-Matrix von $\mathbf{g}$ findet man direkt

$$\begin{aligned}\mathbf{J_g}(\mathbf{x}^{(k-1)}) &= \left.\frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^{(k-1)}} = \left.\frac{\partial\left(\mathbf{u}^{\text{alt}} + \Delta t \cdot \Phi(t^{\text{alt}}, t^{\text{neu}}; \mathbf{u}^{\text{alt}}, \mathbf{x}) - \mathbf{x}\right)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^{(k-1)}} \\ &= \Delta t \cdot \left.\frac{\partial \Phi(t^{\text{alt}}, t^{\text{neu}}; \mathbf{u}^{\text{alt}}, \mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^{(k-1)}} - \mathbb{1}.\end{aligned}$$

Da die Verfahrensfunktion die rechte Seite $\mathbf{f}$ beinhaltet, wird folglich im Allgemeinen die Jacobi-Matrix von $\mathbf{f}$ benötigt.

2 Schrittweitensteuerung

Gegeben sei ein Zeitintervall $[t_0, t_e] \subset \mathbb{R}$ der Länge $T := t_e - t_0$ mit Start- und Endzeitpunkt $t_0, t_e \in \mathbb{R}$. Ein d-dimensionales Anfangswertproblem (AWP) ist die Aufgabe der Bestimmung einer kontinuierlichen Lösung

$$\begin{aligned}\mathbf{u} : [t_0, t_e] &\to \mathbb{R}^d, \\ t &\mapsto \mathbf{u}(t),\end{aligned}$$

für eine rechte Seite $\mathbf{f} : [t_0, t_e] \times \mathbb{R}^d \to \mathbb{R}^d$, so dass gilt

$$\begin{aligned}\frac{\partial}{\partial t}\mathbf{u}(t) &= \mathbf{f}(t, \mathbf{u}(t)), \qquad \text{für alle } t \in [t_0, t_e], \\ \mathbf{u}(t_0) &= \mathbf{u}_0.\end{aligned}$$

Dabei nennt man $\mathbf{u}_0 \in \mathbb{R}^d$ den Startwert. Die geschlossene Lösung des Problems ist gegeben durch

$$\mathbf{u}(t) = \mathbf{u}_0 + \int_{t_0}^{t} \mathbf{f}(s, \mathbf{u}(s))\; ds.$$

2.1 Diskrete Approximation

Eine Strategie zur Bestimmung einer diskreten Approximation besteht darin, das betrachtete Intervall in endlich viele Stücke zu zerlegen und darauf die Lösung durch eine stückweise lineare Funktion anzunähern. Seien dazu n Punkte und Zeitschrittweiten

$$t_0 < t_1 < t_2 < \cdots < t_n = t_e, \quad h_k := t_k - t_{k-1} \ (1 \leq k \leq n), \quad h := \max_{1 \leq k \leq N} h_k,$$

gewählt. Der einfachste Fall ist die Zerlegung in n äquidistante Zeitintervalle, d.h. man wählt eine Zerlegung mit uniformer Zeitschrittweite Δt

$$\Delta t := \frac{T}{n} = \frac{t_e - t_0}{n}$$

und äquidistante Punkte

$$t_k := t_0 + k\Delta t \qquad \text{für alle } k = 1, \ldots, n.$$

In diesem Fall nennt man $h := h_k \ (= \Delta t)$ die uniforme Schrittweite.
Nun sucht man eine diskrete Approximation

$$\mathbf{y}_k \approx \mathbf{u}(t_k) \qquad (0 \leq k \leq n)$$

an den Punkten $t_1, \ldots, t_n$ und berechnet dazu eine Folge von Werten $(\mathbf{y}_k)_{1 \leq k \leq n}$ durch ein numerisches Verfahren der Form

$$\mathbf{y}_k := \mathbf{y}_{k-1} + h_k \Phi(h_k; t_{k-1}, t_k; \mathbf{y}_{k-1}, \mathbf{y}_k), \qquad 1 \leq k \leq n.$$

Ist aus dem Kontext klar, welche Zeitpunkte betrachtet werden, so wird im Folgenden zur kompakteren Notation auch verkürzend

$$\Phi(\mathbf{y}_{k-1}, \mathbf{y}_k) := \Phi(h_k; t_{k-1}, t_k; \mathbf{y}_{k-1}, \mathbf{y}_k)$$

notiert. Prominente Beispiele sind die folgenden Vorschriften:

(i) Expliziter Euler: $\Phi(h_k; t_{k-1}, t_k; \mathbf{y}_{k-1}, \mathbf{y}_k) := \mathbf{f}(t_{k-1}, \mathbf{y}_{k-1})$,

(ii) Impliziter Euler: $\Phi(h_k; t_{k-1}, t_k; \mathbf{y}_{k-1}, \mathbf{y}_k) := \mathbf{f}(t_k, \mathbf{y}_k)$,

(iii) Crank-Nicolson: $\Phi(h_k; t_{k-1}, t_k; \mathbf{y}_{k-1}, \mathbf{y}_k) := \frac{1}{2} \{\mathbf{f}(t_{k-1}, \mathbf{y}_{k-1}) + \mathbf{f}(t_k, \mathbf{y}_k)\}$.

2.2 Konsistenz, globale und lokale Fehler

Definition 4 (Konsistenzfehler)
Der *Konsistenzfehler* (auch *Abschneidefehler*) eines Einschrittverfahrens lautet

$$\tau_k := \tau_k(h_k, t, \mathbf{u}(t), \mathbf{f}) := \frac{\mathbf{u}(t_k) - \mathbf{u}(t_{k-1})}{h_k} - \Phi(h_k; t_{k-1}, t_k; \mathbf{u}(t_{k-1}), \mathbf{u}(t_k)).$$

Der Konsistenzfehler misst folglich, wie gut die diskrete Iterationsvorschrift von der exakten Lösung erfüllt wird. Ein Verfahren hat *Konsistenzordnung* p, falls für den Konsistenzfehler

$$\max_{1 \le k \le n} \|\tau_k\| = \mathcal{O}(h^p)$$

gilt.

Definition 5 (Globaler Fehler)
Der *globale Fehler* eines Einschrittverfahrens am Zeitpunkt t_k ist der Unterschied zwischen exakter und approximativer Lösung

$$\mathbf{e}_k := \mathbf{u}(t_k) - \mathbf{y}_k.$$

Definition 6 (Lokaler Fehler)
Sei $\tilde{\mathbf{y}}_k$ die Lösung nach einem Schritt ausgehend von der exakten Lösung $\mathbf{u}(t_{k-1})$, d.h.

$$\tilde{\mathbf{y}}_k := \mathbf{u}(t_{k-1}) + h_k \Phi(h_k; t_{k-1}, t_k; \mathbf{u}(t_{k-1}), \tilde{\mathbf{y}}_k), \qquad 1 \le k \le n.$$

Der *lokale Fehler* eines Einschrittverfahrens ist

$$\sigma_k := \mathbf{u}(t_k) - \tilde{\mathbf{y}}_k,$$

d.h. der Fehler nach einem Schritt, wenn man mit exakter Lösung startet.

Für den lokalen Fehler gilt (die im Allgemeinen implizite) Beziehung

$$\begin{aligned} \sigma_k &= \mathbf{u}(t_k) - \tilde{\mathbf{y}}_k \\ &= \mathbf{u}(t_k) - \mathbf{u}(t_{k-1}) - h_k \Phi(\mathbf{u}(t_{k-1}), \tilde{\mathbf{y}}_k) \\ &= \mathbf{u}(t_k) - \mathbf{u}(t_{k-1}) - h_k \Phi(\mathbf{u}(t_{k-1}), \mathbf{u}(t_k) - \sigma_k) \end{aligned}$$

Unter der asymptotischen Entwicklung für kleine lokale Fehler

$$\Phi(\mathbf{u}(t_{k-1}), \mathbf{u}(t_k) - \sigma_k) = \Phi(\mathbf{u}(t_{k-1}), \mathbf{u}(t_k)) - \partial_{\mathbf{x}} \Phi(\mathbf{u}(t_{k-1}), \mathbf{x})|_{\mathbf{x}=\mathbf{u}(t_k)} \sigma_k + o(\|\sigma_k\|^2)$$

und der Notation $\Phi' := \partial_{\mathbf{x}}\Phi(\mathbf{u}(t_{k-1}),\mathbf{x})|_{\mathbf{x}=\mathbf{u}(t_k)}$ lässt sich dies umschreiben zu

$$\begin{aligned}
\sigma_k &= \mathbf{u}(t_k) - \mathbf{u}(t_{k-1}) - h_k\Phi(\mathbf{u}(t_{k-1}),\mathbf{u}(t_k)-\sigma_k)\\
&= h_k\left\{\frac{\mathbf{u}(t_k)-\mathbf{u}(t_{k-1})}{h_k} - \Phi(\mathbf{u}(t_{k-1}),\mathbf{u}(t_k)-\sigma_k)\right\}\\
&= h_k\left\{\frac{\mathbf{u}(t_k)-\mathbf{u}(t_{k-1})}{h_k} - \Phi(\mathbf{u}(t_{k-1}),\mathbf{u}(t_k)) + \Phi(\mathbf{u}(t_{k-1}),\mathbf{u}(t_k)) - \Phi(\mathbf{u}(t_{k-1}),\mathbf{u}(t_k)-\sigma_k)\right\}\\
&= h_k\tau_k + h_k\left\{\Phi(\mathbf{u}(t_{k-1}),\mathbf{u}(t_k)) - \Phi(\mathbf{u}(t_{k-1}),\mathbf{u}(t_k)-\sigma_k)\right\}\\
&= h_k\tau_k + \underbrace{h_k\Phi'\sigma_k + o(\|\sigma_k\|^2)}_{=\ 0 \text{ für explizite Verfahren}}\ .
\end{aligned}$$

Folglich gilt der Zusammenhang

$$h_k\tau_k = \begin{cases}\sigma_k, & \text{für explizite Verfahren,}\\ (1-h_k\Phi')\,\sigma_k + o(\|\sigma_k\|^2), & \text{für implizite Verfahren,}\end{cases}$$

d.h. Konsistenzfehler und lokaler Fehler unterscheiden sich bei expliziten Verfahren nur um den Faktor Schrittweite.

2.3 Abschätzungen für den globalen Fehler

Man möchte gerne den globalen Fehler am Zeitpunkt t_k mit dem bereits gemachten (und sich somit fortpflanzenden) Fehler am Zeitpunkt t_{k-1} und dem lokalen Fehler in Zusammenhang setzen. Idealerweise möchte man damit den globalen Fehler durch eine rekursive Formel kontrollieren können.
Dafür notwendig ist die Lipschitz-Stetigkeit der Verfahrensfunktion. Man nimmt also an, dass man diese mit einer Lipschitz-Konstanten $L < \infty$ beschränken kann in der Form

$$\begin{aligned}
\|\Phi(\mathbf{a}_1,\mathbf{x}) - \Phi(\mathbf{a}_2,\mathbf{x})\| &\le L\|\mathbf{a}_1-\mathbf{a}_2\|, \quad \text{für alle } \mathbf{x},\\
\|\Phi(\mathbf{x},\mathbf{b}_1) - \Phi(\mathbf{x},\mathbf{b}_2)\| &\le L\|\mathbf{b}_1-\mathbf{b}_2\|, \quad \text{für alle } \mathbf{x},
\end{aligned}$$

oder zusammengefasst

$$\|\Phi(\mathbf{a}_1,\mathbf{b_1}) - \Phi(\mathbf{a}_2,\mathbf{b_2})\| \le L\left\{\|\mathbf{a}_1-\mathbf{a}_2\| + \|\mathbf{b}_1-\mathbf{b}_2\|\right\}.$$

Nun kann man versuchen den globalen Fehler durch den lokalen zu kontrollieren als

$$\begin{aligned}
\mathbf{e}_k &= \mathbf{u}(t_k) - \mathbf{y}_k = \mathbf{u}(t_k) - \tilde{\mathbf{y}}_k + \tilde{\mathbf{y}}_k - \mathbf{y}_k\\
&= \mathbf{u}(t_k) - \tilde{\mathbf{y}}_k + \mathbf{u}(t_{k-1}) + h_k\Phi(\mathbf{u}(t_{k-1}),\tilde{\mathbf{y}}_k) - \mathbf{y}_{k-1} - h_k\Phi(\mathbf{y}_{k-1},\mathbf{y}_k)\\
&= \underbrace{\mathbf{u}(t_k) - \tilde{\mathbf{y}}_k}_{=\ \sigma_k} + \underbrace{\mathbf{u}(t_{k-1}) - \mathbf{y}_{k-1}}_{=\ \mathbf{e}_{k-1}} + h_k\left\{\Phi(\mathbf{u}(t_{k-1}),\tilde{\mathbf{y}}_k) - \Phi(\mathbf{y}_{k-1},\mathbf{y}_k)\right\}\\
&= \mathbf{e}_{k-1} + \sigma_k + h_k\left\{\Phi(\mathbf{u}(t_{k-1}),\tilde{\mathbf{y}}_k) - \Phi(\mathbf{y}_{k-1},\mathbf{y}_k)\right\}.
\end{aligned}$$

Durch die Lipschitz-Stetigkeit findet man damit

$$\|\mathbf{e}_k\| \leq \|\mathbf{e}_{k-1}\| + \|\sigma_k\| + h_k L \left\{ \|\underbrace{\mathbf{u}(t_{k-1}) - \mathbf{y}_{k-1}}_{\mathbf{e}_{k-1}}\| + \|\underbrace{\tilde{\mathbf{y}}_k - \mathbf{y}_k}_{=\,?}\| \right\}.$$

Dies ist keine praktisch brauchbare Abschätzung, denn der Term $\tilde{\mathbf{y}}_k - \mathbf{y}_k$ lässt sich im Allgemeinen (speziell bei impliziten Verfahren) nicht gut bestimmen. Daher verwendet man besser eine Kontrolle über den Konsistenzfehler gemäß

$$\begin{aligned}
\mathbf{e}_k &= \mathbf{u}(t_k) - \mathbf{y}_k = \mathbf{u}(t_k) - \mathbf{u}(t_{k-1}) + \mathbf{u}(t_{k-1}) - \mathbf{y}_k \\
&= \mathbf{u}(t_k) - \mathbf{u}(t_{k-1}) + \mathbf{u}(t_{k-1}) - \mathbf{y}_{k-1} - h_k \Phi(\mathbf{y}_{k-1}, \mathbf{y}_k) \\
&= \mathbf{u}(t_{k-1}) - \mathbf{y}_{k-1} + h_k \left\{ \frac{\mathbf{u}(t_k) - \mathbf{u}(t_{k-1})}{h_k} - \Phi(\mathbf{y}_{k-1}, \mathbf{y}_k) \right\} \\
&= \mathbf{u}(t_{k-1}) - \mathbf{y}_{k-1} + h_k \left\{ \frac{\mathbf{u}(t_k) - \mathbf{u}(t_{k-1})}{h_k} - \Phi(\mathbf{u}(t_{k-1}), \mathbf{u}(t_k)) \right\} \\
&\qquad + h_k \left\{ \Phi(\mathbf{u}(t_{k-1}), \mathbf{u}(t_k)) - \Phi(\mathbf{y}_{k-1}, \mathbf{y}_k) \right\} \\
&= \mathbf{e}_{k-1} + h_k \tau_k + h_k \left\{ \Phi(\mathbf{u}(t_{k-1}), \mathbf{u}(t_k)) - \Phi(\mathbf{y}_{k-1}, \mathbf{y}_k) \right\}
\end{aligned}$$

und kann mit der Lipschitz-Stetigkeit nun abschätzen

$$\|\mathbf{e}_k\| \leq \|\mathbf{e}_{k-1}\| + h_k \|\tau_k\| + h_k L \left\{ \|\underbrace{\mathbf{u}(t_{k-1}) - \mathbf{y}_{k-1}}_{\mathbf{e}_{k-1}}\| + \|\underbrace{\mathbf{u}(t_k) - \mathbf{y}_k}_{\mathbf{e}_k}\| \right\}$$

und findet folglich die rekursive Beziehung

$$\|\mathbf{e}_k\| \leq \|\mathbf{e}_{k-1}\| + \underbrace{h_k L \|\mathbf{e}_{k-1}\|}_{=\ 0 \text{ für implizit}} + \underbrace{h_k L \|\mathbf{e}_k\|}_{=\ 0 \text{ für explizit}} + h_k \|\tau_k\|,$$

$$\text{bzw.} \quad \|\mathbf{e}_k\| \leq \left(\frac{1 + h_k L}{1 - h_k L} \right) \|\mathbf{e}_{k-1}\| + \left(\frac{h_k}{1 - h_k L} \right) \|\tau_k\|.$$

Für ein explizites Verfahren kann man nun

$$\|\mathbf{e}_k\| \leq \|\mathbf{e}_{k-1}\| + h_k L \|\mathbf{e}_{k-1}\| + h_k \|\tau_k\|$$

sukzessive Anwenden und erhält

$$\|\mathbf{e}_k\| \leq \sum_{i=0}^{k-1} \underbrace{h_{i+1}L}_{=:\ a_i} \|\mathbf{e}_i\| + \underbrace{\|\mathbf{e}_0\| + \sum_{i=1}^{k} h_i \|\tau_i\|}_{=:\ b_k}.$$

Analog liefert ein implizites Verfahren

$$\|\mathbf{e}_k\| \leq \sum_{i=1}^{k} \underbrace{h_i L}_{=:\ a_i} \|\mathbf{e}_i\| + \underbrace{\|\mathbf{e}_0\| + \sum_{i=1}^{k} h_i \|\tau_i\|}_{=:\ b_k}.$$

Die Kontrolle des Fehlers über den Konsistenzfehler lässt sich abschließen mit dem diskreten Lemma von Gronwall.

Lemma 7 (Diskretes Lemma von Gronwall)
Seien $(w_i)_{i\geq 0}, (a_i)_{i\geq 0}$ und $(b_i)_{i\geq 0}$ Folgen nichtnegativer Zahlen und es gelte

$$w_0 \leq b_0, \quad \text{sowie} \quad w_k \leq \sum_{i=0}^{k-1} a_i w_i + b_k,\ k \geq 1.$$

Ist die Folge $(b_i)_{i\geq 0}$ nichtfallend, so gilt

$$w_k \leq \exp\left(\sum_{i=0}^{k-1} a_i\right) \cdot b_k,\ k \geq 1.$$

Gilt zudem $a_i < 1$ $(1 \leq i \leq k)$ so folgt aus

$$w_k \leq \sum_{i=0}^{k} a_i w_i + b_k,\ k \geq 1$$

die Abschätzung

$$w_k \leq \exp\left(2\sum_{i=0}^{k} a_i\right) \cdot b_k,\ k \geq 1.$$

Beweis. Übung. □

Somit findet man die explizite Fehlerabschätzung

$$\|\mathbf{e}_k\| \leq \exp\left(\sum_{i=0}^{k-1} h_{i+1}L\right) \cdot \left\{\|\mathbf{e}_0\| + \sum_{i=1}^{k} h_i\|\tau_i\|\right\} = e^{L(t_k - t_0)} \cdot \left\{\|\mathbf{e}_0\| + \sum_{i=1}^{k} h_i\|\tau_i\|\right\}.$$

und analog für implizite Verfahren unter der Bedingung $h_i L < 1$ $(1 \leq i \leq k)$

$$\|\mathbf{e}_k\| \leq \exp\left(2\sum_{i=1}^{k} h_i L\right) \cdot \left\{\|\mathbf{e}_0\| + \sum_{i=1}^{k} h_i \|\tau_i\|\right\} = e^{2L(t_k - t_0)} \cdot \left\{\|\mathbf{e}_0\| + \sum_{i=1}^{k} h_i \|\tau_i\|\right\}.$$

Damit findet man in beiden Fällen:

Satz 8 (A priori Abschätzung des globalen Fehlers)
Für ein Lipschitz-stetiges Einschrittverfahren (und bei impliziten Verfahren unter der Bedingung $h_k L < 1,\ 1 \leq k \leq n$) gilt für die a priori Abschätzung des globalen Fehlers

$$\max_{1 \leq k \leq n} \|\mathbf{e}_k\| \leq e^{2LT} \cdot \left\{\|\mathbf{e}_0\| + T \max_{1 \leq k \leq n} \|\tau_k\|\right\},$$

d.h. der maximale globale Fehler im Intervall $[t_0, t_n]$ lässt sich bis auf eine Konstante (abhängig von $T := |t_n - t_0|$ und der Lipschitz-Konstanten L) abschätzen durch den Startfehler und den maximalen Konsistenzfehler.

2.4 Schrittweitenkontrolle

Bei der praktischen Durchführung eines diskreten Verfahrens stellt sich nun die Frage, wie die Zeitschritte konkret gewählt werden sollen. Einerseits möchte man sicherlich möglichst große Zeitschritte vornehmen, um die Anzahl der Rechenschritte und damit den Rechenaufwand gering zu halten. Auf der anderen Seite soll der Zeitschritt jedoch so klein sein, dass der gemachte Fehler kontrollierbar und idealerweise unterhalb einer vom Benutzer vorgegebenen Schranke bleibt. An kritischen Stellen, z.B. dort wo wie Lösung stark variiert, erfordert dies gewiss kleine Zeitschritte, jedoch können an unkritischen Stellen, z.B. dort wo wie Lösung sehr glatt und regulär ist, deutlich größere Schritte zur Reduzierung des Rechenaufwands zugelassen werden.
Die direkte Kontrolle des globalen Fehlers wäre dabei am wünschenswertesten, jedoch ist dieser nur schwer zugänglich. Nimmt man exakte Startwerte und rundungsfehlerfreie Computerarithmetik an, so legt die *a priori* Fehlerabschätzung für den globalen Fehler

$$\max_{1 \leq k \leq n} \|\mathbf{e}_k\| \leq KT \max_{1 \leq k \leq n} \|\tau_k\|,$$

nahe, stattdessen den Fehler über eine Kontrolle des Konsistenzfehlers in den einzelnen Zeitschritten zu steuern. Dadurch wird der Fehler zwar nur bis auf eine Konstante $K := e^{2LT}$ kontrolliert, die im Extremfall exponentiell mit dem betrachteten Zeitintervall anwächst, jedoch ist diese Konstante für Probleme aus der Praxis nur von moderater Größenordnung. Besitzt die rechte

Seite $\mathbf{f}$ z.B. neben der Lipschitz-Stetigkeit noch stärkere gutartige Eigenschaften, so kann man sogar oftmals auch mathematisch bessere strengere Abschätzung zeigen. Im Folgenden wird daher $K \approx 1$ angenommen.
Der Konsistenzfehler selbst ist ebenfalls nicht exakt zugänglich, aber man kann geeignete Schätzungen für diesen vornehmen.

2.4.1 Schätzungen für den Konsistenzfehler

Besitzt das numerische Verfahren Konsistenzordnung p, so lässt sich der Konsistenzfehler über das Restglied der Taylorentwicklung abschätzen und man findet

$$\|\tau_k\| \leq \frac{1}{(p+1)!}\ h_k^p \max_{t\in[t_{k-1},t_k]} \|\mathbf{u}^{(p+1)}(t)\|.$$

Aufgrund der Anfangswertaufgabe gilt $\mathbf{u}^{(p+1)}(t) = \mathbf{f}^{(p)}(t, \mathbf{u}(t))$ und damit ließe sich der Konsistenzfehler durch Ableitungen von $\mathbf{f}$ und Abschätzungen von $\mathbf{u}$ beschränken. Allerdings ist dies mit hohem Aufwand verbunden und daher in der Praxis tendenziell nicht anzuraten.
Vielmehr verwendet man einen heuristischen Standpunkt. Der Einfachheit halber wird die folgende Argumentation auf explizite Verfahren beschränkt, sie lässt sich jedoch leicht auf implizite Verfahren verallgemeinern. Unter der Beachtung von

$$h_k\tau_k = \sigma_k = \mathbf{u}(t_k) - \tilde{\mathbf{y}}_k$$

lässt sich der Konsistenzfehler durch den lokalen Fehler, d.h. die Differenz zwischen exakter Lösung $\mathbf{u}(t_k)$ und diskreter Approximation aus vorhergehender exakter Lösung $\tilde{\mathbf{y}}_k$, berechnen.
Nun macht man zunächst zwei Annahmen:

(i) Man nimmt an, dass die Approximation im letzten Schritt so gut war, dass sie quasi-exakt ist, d.h. $\mathbf{y}_{k-1} \approx \mathbf{u}(t_{k-1})$. Dies ist dadurch motiviert, dass man bereits den Fehler in den vorgehenden Berechnungen kontrolliert und somit klein gehalten hat. Damit kann man $\mathbf{y}_k \approx \tilde{\mathbf{y}}_k$ setzen, d.h. man betrachtet formal gesehen den Konsistenzfehler entlang der diskreten Approximation und nicht entlang der exakten Lösung.

(ii) Man nimmt an, dass man neben der Approximation $\mathbf{y}_k$ eine weitere diskrete Lösung $\mathbf{y}_k^{\text{exakt}}$ berechnen kann, die deutlich besser den exakten Wert $\mathbf{u}(t_k)$ annähert. Ist dies der Fall, so kann man für die Berechnung des lokalen Fehlers anstatt der exakten Lösung auch $\mathbf{y}_k^{\text{exakt}} \approx \mathbf{u}(t_k)$ verwenden.

Durch diese beiden Annahmen erhält man einen sogenannten *a posteriori* Fehlerschätzer, denn die Abschätzung von τ_k lässt sich nun nach der Berechnung von zwei Approximation auf die Differenz

$$\sigma_k = h_k \tau_k \approx \mathbf{y}_k^{\text{exakt}} - \mathbf{y}_k$$

zurückführen. Desweiteren ist es hilfreich eine dritte Annahme zu treffen:

(iii) Man nimmt an, dass der Konsistenzfehler τ_k eine lokale Entwicklung

$$\tau_k(h_k) = \mathbf{c} h_k^p + \mathcal{O}(h_k^{p+1})$$

auf dem Intervall $[t_{k-1}, t_k]$ mit einer Konstanten $\mathbf{c} \in \mathbb{R}^d$ unabhängig von h_k erlaubt.

Beachtet man die Abschätzungen der Konsistenzfehler mit Hilfe der Taylor-Entwicklungen, so ist dies oftmals erfüllt.

Für die Wahl einer geeigneten Methode von $\mathbf{y}_k^{\text{exakt}}$ lassen sich zwei Ansätze unterscheiden:

(S1) Verwende dasselbe numerische Verfahren der Ordnung p sowohl für $\mathbf{y}_k$ als auch $\mathbf{y}_k^{\text{exakt}}$. Dazu berechnet man $\mathbf{y}_k$ mit einer Schrittweite h_k, jedoch $\mathbf{y}_k^{\text{exakt}}$ durch zwei Schritte mit Schrittweite $\frac{h_k}{2}$.

(S2) Berechnet man $\mathbf{y}_k$ mit einem numerischen Verfahren der Ordnung p und Zeitschritt h_k, so wählt man zur Berechnung von $\mathbf{y}_k^{\text{exakt}}$ denselben Zeitschritt h_k, jedoch ein Verfahren der Ordnung $p+1$.

Im Fall (S1) berechnet man also zusätzlich zum gesuchten $\mathbf{y}_k^{(h_k)}$

$$\mathbf{y}_k^{(h_k)} = \mathbf{y}_{k-1} + h_k \Phi(\mathbf{y}_{k-1})$$

noch eine verfeinerte Lösung $\mathbf{y}_k^{(h_k/2)}$ durch

$$\mathbf{y}_{k-\frac{1}{2}}^{(h_k/2)} = \mathbf{y}_{k-1} + \frac{h_k}{2}\Phi(\mathbf{y}_{k-1}), \qquad \mathbf{y}_k^{(h_k/2)} = \mathbf{y}_{k-\frac{1}{2}}^{(h_k/2)} + \frac{h_k}{2}\Phi(\mathbf{y}_{k-\frac{1}{2}}^{(h_k/2)}),$$

durch einen Zwischenschritt $\mathbf{y}_{k-\frac{1}{2}}^{(h_k/2)}$ an $t_{k-\frac{1}{2}} := t_{k-1} + \frac{h_k}{2}$ und damit findet man eine erste heuristische Schätzung für $\tau_k(h_k)$ durch

$$\tau_k(h_k) = \frac{\sigma_k}{h_k} \approx \frac{\mathbf{y}_k^{(h_k/2)} - \mathbf{y}_k^{(h_k)}}{h_k}.$$

Eine asymptotisch besser Schätzung gewinnt man, wenn man die Annahmen (i) und (iii) verwendet. Damit findet man

$$\begin{aligned}
\mathbf{y}_k^{(h_k/2)} - \mathbf{y}_k^{(h_k)} &= \mathbf{y}_{k-\frac{1}{2}}^{(h_k/2)} + \frac{h_k}{2}\Phi(\mathbf{y}_{k-\frac{1}{2}}^{(h_k/2)}) - \mathbf{y}_{k-1} - h_k\Phi(\mathbf{y}_{k-1}) \\
&= \mathbf{y}_{k-1} + \frac{h_k}{2}\Phi(\mathbf{y}_{k-1}) + \frac{h_k}{2}\Phi(\mathbf{y}_{k-\frac{1}{2}}^{(h_k/2)}) - \mathbf{y}_{k-1} - h_k\Phi(\mathbf{y}_{k-1}) \\
&= \mathbf{u}(t_k) - \mathbf{u}(t_{k-1}) - h_k\Phi(\mathbf{y}_{k-1}) - \mathbf{u}(t_k) + \mathbf{u}(t_{k-\frac{1}{2}}) + \frac{h_k}{2}\Phi(\mathbf{u}(t_{k-\frac{1}{2}})) \\
&\quad - \mathbf{u}(t_{k-\frac{1}{2}}) + \mathbf{u}(t_{k-1}) + \frac{h_k}{2}\Phi(\mathbf{y}_{k-1}) + \frac{h_k}{2}\Phi(\mathbf{y}_{k-\frac{1}{2}}^{(h_k/2)}) - \frac{h_k}{2}\Phi(\mathbf{u}(t_{k-\frac{1}{2}})) \\
&= h_k\tau_k(h_k) - \frac{h_k}{2}\tau_k(\frac{h_k}{2}) + \frac{h_k}{2}\tau_{k-\frac{1}{2}}(\frac{h_k}{2}) + \frac{h_k}{2}\left\{\Phi(\mathbf{y}_{k-\frac{1}{2}}^{(h_k/2)}) - \Phi(\mathbf{u}(t_{k-\frac{1}{2}}))\right\} \\
&= \mathbf{c}h_k^{p+1} - \mathbf{c}\left(\frac{h_k}{2}\right)^{p+1} - \mathbf{c}\left(\frac{h_k}{2}\right)^{p+1} + \frac{h_k}{2}\Phi'(\mathbf{y}_{k-\frac{1}{2}}^{(h_k/2)})(\mathbf{u}(t_{k-\frac{1}{2}}) - \mathbf{y}_{k-\frac{1}{2}}^{(h_k/2)}) + \mathcal{O}(h^{p+2}) \\
&= \mathbf{c}h_k^{p+1}\left(1 - \frac{1}{2^p}\right) + \mathcal{O}(h^{p+2}) \\
&= h_k\tau_k(h_k)\left(1 - \frac{1}{2^p}\right) + \mathcal{O}(h^{p+2})
\end{aligned}$$

und damit die Darstellung des Konsistenzfehlers

$$\tau_k(h_k) = \frac{\mathbf{y}_k^{(h_k/2)} - \mathbf{y}_k^{(h_k)}}{h_k\left(1 - \frac{1}{2^p}\right)} + \mathcal{O}(h_k^{p+1}). \tag{S1}$$

Dabei wurde die Annahme $\mathbf{y}_{k-1} = \mathbf{u}(t_{k-1})$ dahingehend verwendet, dass gilt

$$\begin{aligned}
\mathbf{u}(t_{k-\frac{1}{2}}) - \mathbf{y}_{k-\frac{1}{2}}^{(h_k/2)} &= \mathbf{u}(t_{k-\frac{1}{2}}) - \mathbf{y}_{k-1} - \frac{h_k}{2}\Phi(\mathbf{y}_{k-1}) \\
&= \frac{h_k}{2}\tau_{k-\frac{1}{2}} + \mathbf{u}(t_{k-1}) - \mathbf{y}_{k-1} + \frac{h_k}{2}\left\{\Phi(\mathbf{u}(t_{k-1})) - \Phi(\mathbf{y}_{k-1})\right\} \\
&= \frac{h_k}{2}\tau_{k-\frac{1}{2}} = \mathbf{c}h_k^{p+1} + \mathcal{O}(h^{p+2}) = \mathcal{O}(h_k^{p+1}).
\end{aligned}$$

Im Fall (S2) berechnet man zwei diskrete Approximationen unterschiedlicher Ordnung

$$\begin{aligned}
\mathbf{y}_k^{(p)} &= \mathbf{y}_{k-1} + h_k\Phi^{(p)}(\mathbf{y}_{k-1}), \\
\mathbf{y}_k^{exakt} := \mathbf{y}_k^{(p+1)} &= \mathbf{y}_{k-1} + h_k\Phi^{(p+1)}(\mathbf{y}_{k-1}),
\end{aligned}$$

und betrachtet die Differenz der beiden Methoden. So erhält man

$$\begin{aligned}
\mathbf{y}_k^{(p+1)} - \mathbf{y}_k^{(p)} &= \mathbf{y}_{k-1} + h_k \Phi^{(p+1)}(\mathbf{y}_{k-1}) - \mathbf{y}_{k-1} - h_k \Phi^{(p)}(\mathbf{y}_{k-1}) \\
&= h_k \Phi^{(p+1)}(\mathbf{y}_{k-1}) - h_k \Phi^{(p)}(\mathbf{y}_{k-1}) = h_k \Phi^{(p+1)}(\mathbf{u}(t_{k-1})) - h_k \Phi^{(p)}(\mathbf{u}(t_{k-1})) \\
&= h_k \tau_k^{(p)}(h_k) - h_k \tau_k^{(p+1)}(h_k) \\
&= h_k \left(\mathbf{c}^{(p)} h_k^p + \mathcal{O}(h_k^{p+1}) - \mathbf{c}^{(p+1)} h_k^{p+1} - \mathcal{O}(h_k^{p+2}) \right) \\
&= \mathbf{c}^{(p)} h_k^{p+1} + \mathcal{O}(h_k^{p+2}) \\
&= h_k \tau_k^{(p)}(h_k) + \mathcal{O}(h_k^{p+2}).
\end{aligned}$$

Man beachte, dass dabei zur praktischen Bestimmung nur die Differenz

$$\mathbf{y}_k^{(p+1)} - \mathbf{y}_k^{(p)} = h_k \left\{ \Phi^{(p+1)}(\mathbf{y}_{k-1}) - \Phi^{(p)}(\mathbf{y}_{k-1}) \right\}$$

berechnet werden muss und dadurch möglich Rundungsfehler vermieden werden können, falls die Lösungen sehr groß gegenüber der Verfahrensfunktion sein sollten. Man findet hier die Schätzung für den Fehler des Verfahrens der Ordnung p als

$$\tau_k(h_k) = \frac{\mathbf{y}_k^{(p+1)} - \mathbf{y}_k^{(p)}}{h_k} + \mathcal{O}(h_k^{p+1}). \tag{S2}$$

Man beachte, dass der Fehler des Fehlerschätzers um eine Potenz schneller gegen Null geht als der Fehler des Verfahrens. Es handelt sich damit um einen asymptotisch korrekten Fehlerschätzer für $h_k \to 0$.

2.4.2 Optimale Schrittweite

Beide Verfahren (S1)-(S2) zur Fehlerschätzung erlauben es, bei Rechnungen mit gewälter Schrittweite h_k den resultierenden Konsistenzfehler durch eine Schätzung $\tilde{\tau}_k$

$$\tau_k(h_k) = \tilde{\tau}_k(h_k) + \mathcal{O}(h_k^{p+1}) := \begin{cases} \frac{\mathbf{y}_k^{(h_k/2)} - \mathbf{y}_k^{(h_k)}}{h_k\left(1 - \frac{1}{2^p}\right)} + \mathcal{O}(h_k^{p+1}), & \text{(S1)} \\ \frac{\mathbf{y}_k^{(p+1)} - \mathbf{y}_k^{(p)}}{h_k} + \mathcal{O}(h_k^{p+1}), & \text{(S2)} \end{cases}$$

zu bestimmen. Nun ist jedoch das zu wählende h_k a priori gar nicht bekannt, sondern soll ja gerade auf Grund der Schätzung ermittelt werden. Da jedoch die Schätzung für beliebiges h_k durchgeführt werden kann, geht man folgendermaßen vor: Man wählt testweise eine Schrittweite h_{test} und berechnet mit

dieser die Größe des resultierenden Konsistenzfehlers $\|\tilde{\tau}_k(h_{\text{test}})\|$. Unter der Annahme (iii) lässt sich dieser Fehler schreiben als

$$\|\tilde{\tau}_k(h_{\text{test}})\| + \mathcal{O}(h_{\text{test}}^{p+1}) = \|\tau_k(h_{\text{test}})\| = \|\mathbf{c}\| h_{\text{test}}^p + \mathcal{O}(h_{\text{test}}^{p+1})$$

und dies erlaubt nicht nur eine Bestimmung der Konstanten $\|\mathbf{c}\|$, sondern liefert auch die zu erwartende Konsistenzfehlergröße an frei wählbarer Schrittweite h durch

$$\|\tau_k(h)\| = \|\mathbf{c}\| h^p + \mathcal{O}(h^{p+1}) = \frac{\|\tilde{\tau}_k(h_{\text{test}})\|}{h_{\text{test}}^p} h^p + \mathcal{O}(h^{p+1}) + \mathcal{O}(h_{\text{test}}^{p+1}).$$

Da der Term $\frac{\|\tilde{\tau}_k(h_{\text{test}})\|}{h_{\text{test}}^p}$ aus der Testrechnung bekannt ist, lässt sich damit der Konsistenzfehler durch geeignete Wahl der Schrittweite h beliebig einstellen.

Sei nun eine Fehlerschranke TOL vorgegeben, unter der man den Fehler der Rechnung halten möchte. Erinnert man sich an die Fehlerakkumulation nach n Schritten

$$\max_{1 \le k \le n} \|\mathbf{e}_k\| \approx \sum_{k=1}^{n} h_k \|\tau_k(h_k)\|,$$

so bieten sich zwei Varianten an:

(V1) Weiß man a priori nicht, welche Zeitspanne simuliert werden soll, so kann man die Schrittweite h_k in jedem Schritt so einzustellen, dass für den lokalen Fehler

$$\|\sigma_k\| = h_k \|\tau_k(h_k)\| \approx TOL$$

gilt. Dann erhält man nach n Schritten einen globalen Fehler von

$$\max_{1 \le k \le n} \|\mathbf{e}_k\| \approx \sum_{k=1}^{n} h_k \|\tau_k(h_k)\| \approx n \cdot TOL.$$

Man findet somit die gewünschte Schrittweite $h_{k,opt}$ als

$$TOL = \frac{\|\tilde{\tau}_k(h_{\text{test}})\|}{h_{\text{test}}^p} h_{k,opt}^{p+1} \quad \Rightarrow \quad h_{k,opt} \approx \left(\frac{TOL}{h_{\text{test}} \|\tilde{\tau}_k(h_{\text{test}})\|} \right)^{\frac{1}{p+1}} h_{\text{test}}.$$

(V2) Weiß man a priori, dass das Zeitintervall der Länge T berechnet werden soll, so kann man die Schrittweiten so einstellen, dass der globale Fehler unter der vorgegebenen Toleranz bleibt, indem man

$$\|\tau_k(h_k)\| \approx \frac{TOL}{T}$$

wählt, denn damit folgt wie gewünscht

$$\max_{1\le k\le n}\|\mathbf{e}_k\| \approx \sum_{k=1}^{n} h_k\|\tau_k(h_k)\| \approx \sum_{k=1}^{n} h_k\frac{TOL}{T} = \frac{TOL}{T}\sum_{k=1}^{n} h_k = TOL.$$

Man findet somit die gewünschte Schrittweite $h_{k,opt}$ als

$$\frac{TOL}{T} = \frac{\|\tilde{\tau}_k(h_{\text{test}})\|}{h_{\text{test}}^p}h_{k,opt}^p \quad \Rightarrow \quad h_{k,opt} \approx \left(\frac{TOL}{T\|\tilde{\tau}_k(h_{\text{test}})\|}\right)^{\frac{1}{p}} h_{\text{test}}.$$

Gemäß Taylor-Entwicklung gilt für die meisten Verfahren

$$\tau_k(h_k) \approx \frac{1}{(p+1)!}\mathbf{u}^{(p+1)}(\xi_k)\, h_k^p \quad \text{mit einem } \xi_k \in [t_{k-1}, t_k]$$

und daher lässt sich die Anzahl der benötigten Schritte abschätzen durch

$$\begin{aligned} n &= \sum_{1\le k\le n}\frac{h_k}{h_k} \approx \sum_{1\le k\le n} h_k\left(\frac{T\ \|\mathbf{u}^{(p+1)}(\xi_k)\|}{TOL\cdot(p+1)!}\right)^{\frac{1}{p}} \\ &= \left(\frac{T}{TOL\cdot(p+1)!}\right)^{\frac{1}{p}} \sum_{1\le k\le n} h_k(\|\mathbf{u}^{(p+1)}(\xi_k)\|)^{1/p} \\ &\approx \left(\frac{T}{TOL\cdot(p+1)!}\right)^{\frac{1}{p}} \int_{t_0}^{t_e}(\|\mathbf{u}^{(p+1)}(t)\|)^{1/p}\ dt, \end{aligned}$$

d.h. je größer die Ableitungen der Funktion, je größer das Zeitintervall oder je kleiner die Toleranz, desto mehr Schritte sind zu erwarten.

2.4.3 Algorithmus zur Schrittweitensteuerung

Nun stehen alle Werkzeuge zur Verfügung, um den Algorithmus für die Schrittweitensteuerung anzugehen. Dazu sei der Startwert $\mathbf{y}_0$ am Startzeitpunkt t_0 gegeben. Zudem gibt man sich für die gesamte Rechnung eine minimale und maximale Schrittweite $h_{\min}$ und $h_{\max}$ vor. Für den ersten Schritt benötigt man zudem eine Starttestschrittweite h_{test} (z.B. $h_{\text{test}} := h_{\max}$).
Nun führt man Zeitschritt für Zeitschritt einen der beiden folgenden Algorithmen durch:

(Voraus.) Sei die diskrete Lösung $\mathbf{y}_{k-1} \approx \mathbf{u}(t_{k-1})$ am Zeitpunkt t_{k-1} berechne, eine Testschrittweite h_{test} gegeben und eine geeignete Schrittweite h_k zur Berechnung der Lösung $\mathbf{y}_k$ am nächsten Zeitpunkt $t_k := t_{k-1} + h_k$ gesucht.

(Algo. 1) Steuerung durch Schrittweitenhalbierung und lokalen Fehler:

(1) Berechne mit der Testschrittweite h_{test} eine Schätzung des Konsistenzfehlers $\tilde{\tau}_k(h_{\text{test}})$ mittels (S1) oder (S2) .

(2) Prüfe, ob $h_{\text{test}}\|\tilde{\tau}_k(h_{\text{test}})\| > TOL$ gilt

Falls **ja**: (Schrittweite zu groß)

- Wiederhole Schritt (1) mit $h_{\text{test}} := \frac{1}{2}h_{\text{test}}$
(bzw. beende falls $h_{\text{test}} < h_{\min}$ wird).

Falls **nein**: (Akzeptiere Schrittweite h_{test})

- Wähle $h_k := h_{\text{test}}$ und verwende die beste bereits berechnete Lösung $\mathbf{y}_k$ (d.h. $\mathbf{y}_k^{(p+1)}$ oder $\mathbf{y}_k^{(h_k/2)}$ aus Schritt (1)).
- Plane den nächsten Schritt: Berechne die optimale Schrittweite $h_{k,opt}$ nach (V1) und setze $h_{\text{test}} := h_{k,opt}$.
(und $h_{\text{test}} := \min(h_{\text{test}}, 2h_k, h_{\max})$, $h_{\text{test}} := \max(h_{\text{test}}, h_{\min})$).

(Algo. 2) Steuerung über $h_{k,opt}$.

(1) Berechne mit der Testschrittweite h_{test} eine Schätzung des Konsistenzfehlers $\tilde{\tau}_k(h_{\text{test}})$ mittels (S1) oder (S2) und ermittle daraus die optimale Schrittweite $h_{k,opt}$ nach Variante (V1) oder (V2).

(2) Prüfe, ob $h_{k,opt} \ll h_{\text{test}}$ gilt (z.B. $h_{k,opt} < \frac{1}{4}h_{\text{test}}$)

Falls **ja**: (Die Schätzung von $\tilde{\tau}_k$ ist zu grob)

- Wiederhole Schritt (1) mit $h_{\text{test}} := h_{k,opt}$
(und $h_{\text{test}} := \max(h_{k,opt}, h_{\max})$, beende falls $h_{\text{k,opt}} < h_{\min}$).

Falls **nein**: (Akzeptiere Schrittweite h_{test})

- Wähle $h_k := h_{\text{test}}$ und verwende die beste bereits berechnete Lösung $\mathbf{y}_k$ (d.h. $\mathbf{y}_k^{(p+1)}$ oder $\mathbf{y}_k^{(h_k/2)}$ aus Schritt (1)).
- Plane den nächsten Schritt mit $h_{\text{test}} := 2h_{\text{test}}$
(und $h_{\text{test}} := \min(h_{\text{test}}, h_{\max})$).

Ist eine Endzeit t_e vorgegeben, so wählt man am Ende die Zeitschritte durch $h_{\text{test}} := \min(h_{\text{test}}, t_e - t_{k-1})$ am besten so, dass dieser Zeitpunkt nicht überschritten wird.
In beiden Verfahren geht man davon aus, dass die Schrittweitenwahl sich nicht abrupt ändert solle. Daher lässt man den nachfolgenden Zeitschritt nicht mehr als das doppelte (evtl. auch drei-, vierfache - je nach Geschmack)

anwachsen. Gelegentlich führt man zur Berechnung der optimalen Schrittweite noch einen Sicherheitsfaktor ein und wählt $h_{\text{test}} := \delta \cdot h_{k,opt}$ mit z.B. $\delta = 0,8$ oder $0,9$.

2.4.4 Eingebettete Runge-Kutta-Methoden

Möchte man die Schrittweitensteuerung mittels Verfahren verschiedener Ordnung durchführen, so verwendet man dazu gerne sogenannte eingebettete Runge-Kutta-Verfahren. Diese zeichnen sich dadurch aus, dass bei der Durchführung viele Teilberechnungen der Methode mit niedriger Ordnung bei der Methode mit hoher Ordnung wiederverwendet werden können.
Ganz allgemein können Runge-Kutta-Verfahren über einen Integrationsprozess motiviert werden. Dazu betrachtet man die Volterra-Integralgleichung

$$\mathbf{u}(t_k) = \mathbf{u}(t_{k-1}) + \int_{t_{k-1}}^{t_k} \mathbf{f}(s, \mathbf{u}(s))ds$$

und approximiert das Integral durch numerische Quadratur an den L Quadraturpunkten

$$t_{k-1,l} := t_{k-1} + c_l h_k, \quad l = 1, \ldots, L,$$

mit mit den Quadraturgewichten $b_1, \ldots, b_L$ und erhält folglich

$$\mathbf{u}(t_k) \approx \mathbf{u}(t_{k-1}) + (t_k - t_{k-1}) \sum_{l=1}^{L} b_l \, \mathbf{f}(t_{k-1,l}, \mathbf{u}(t_{k-1,l})).$$

Dies erhebt man nun zur Iterationsvorschrift und erhält

$$\mathbf{y}_k = \mathbf{y}_{k-1} + h_k \sum_{l=1}^{L} b_l \mathbf{f}(t_{k-1,l}, \underbrace{\mathbf{u}(t_{k-1,l})}_{=\,?}).$$

Dabei ist jedoch zunächst offen, wie der exakte Wert an den Zwischenstellen ermittelt werden soll. Die Idee der Runge-Kutta-Methoden besteht nun darin, diesen Wert $\mathbf{u}(t_{k-1,l})$ anzunähern, indem man wieder Quadratur auf *denselben* Stützstellen verwendet, d.h.

$$\mathbf{y}_k = \mathbf{y}_{k-1} + h_k \sum_{l=1}^{L} b_l \, \mathbf{f}(t_{k-1,l}, \mathbf{y}_{k-1,l}),$$

$$\text{sowie} \quad \mathbf{u}(t_{k-1,l}) \approx \mathbf{y}_{k-1,l} = \mathbf{y}_{k-1} + h_k \sum_{s=1}^{L} a_{ls} \mathbf{f}(t_{k-1,s}, \mathbf{y}_{k-1,s}),$$

wobei man dies oftmals kompakt notiert als die Lösung des Systems von Gleichungen der Form

$$t_{k-1,l} := t_{k-1} + c_l h_k, \quad \mathbf{k}_l := \mathbf{f}(t_{k-1,l}, \mathbf{y}_{k-1,l}), \qquad l = 1, \ldots, L$$
$$\mathbf{y}_{k-1,l} = \mathbf{y}_{k-1} + h_k \sum_{s=1}^{L} a_{ls}\mathbf{k}_s, \qquad \mathbf{y}_k = \mathbf{y}_{k-1} + h_k \sum_{l=1}^{L} b_l\, \mathbf{k}_l.$$

Definition 9 (Runge-Kutta-Verfahren)
Sei die Lösung $\mathbf{y}_{k-1}$ einer Anfangswertaufgabe bekannt und die Lösung $\mathbf{y}_k$ zum Zeitpunkt t_k gesucht. Ein allgemeines L-stufiges *Runge-Kutta-Verfahren* berechnet an den Zwischenzeitpunkten $t_{k-1,l}$ Zwischenwerte $\mathbf{y}_{k-1,l}$ durch Lösen von

$$t_{k-1,l} = t_{k-1} + c_l h_k, \quad l = 1, \ldots, L,$$
$$\mathbf{y}_{k-1,l} = \mathbf{y}_{k-1} + h_k \sum_{s=1}^{L} a_{ls}\mathbf{f}(t_{k-1,s}, \mathbf{y}_{k-1,s}), \quad l = 1, \ldots, L,$$

und kombiniert diese zum neuen Wert gemäß

$$\mathbf{y}_k = \mathbf{y}_{k-1} + h_k \sum_{l=1}^{L} b_l\, \mathbf{f}(t_{k-1,l}, \mathbf{y}_{k-1,l}).$$

Die Koeffizienten des Verfahrens können durch ein Butcher-Tableau dargestellt werden:

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b}^T \end{array} \qquad \text{bzw.} \qquad \begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \ldots & a_{1L} \\ c_2 & a_{21} & a_{22} & \ldots & a_{2L} \\ \vdots & \vdots & & \ddots & \vdots \\ c_L & a_{L1} & a_{L2} & \ldots & a_{LL} \\ \hline & b_1 & b_2 & \ldots & b_L \end{array}.$$

Gilt $a_{ls} = 0, l \geq s$, so ist das Verfahren explizit. Prominente Vertreter sind folgende Verfahren:

- $L = 1$: **Euler**, Ordnung 1: explizit: $\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$, implizit: $\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$
- $L = 2$: **Runge**, Ordnung 2: $\begin{array}{c|cc} 0 & & \\ \frac{1}{2} & \frac{1}{2} & \\ \hline & 0 & 1 \end{array}$

- $L = 2$: **Heun**, Ordnung 2:

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

- $L = 2$: **Crank-Nicolson**, Ordnung 2:

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

- $L = 3$: **Heun**, Ordnung 3:

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{3} & \frac{1}{3} & & \\ \frac{2}{3} & 0 & \frac{2}{3} & \\ \hline & \frac{1}{4} & 0 & \frac{3}{4} \end{array}$$

- $L = 4$: **Runge-Kutta**, Ordnung 4:

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

Im expliziten Fall muss man die rechte Seite $\mathbf{f}$ nur L mal auswerten, denn das System lässt sich sukzessive auflösen.

Ein eingebettetes Runge-Kutta-Verfahren besteht nun aus zwei Runge-Kutta-Verfahren unterschiedlicher Ordnung, so dass der Großteil der Koeffizienten (und damit die nötigen Berechnungen) gleich sind.

Definition 10 (Eingebettetes Runge-Kutta-Verfahren)
Ein eingebettetes Runge-Kutta-Verfahren der Ordnung $p/(p+1)$ sind zwei Verfahren der Ordnung p bzw. $p+1$ deren Koeffizienten sich nur in den $\mathbf{b}^{(p)}$ bzw. $\mathbf{b}^{(p+1)}$ unterscheiden, für die jedoch die Koeffizienten $\mathbf{A}$ und $\mathbf{c}$ übereinstimmen. Die Koeffizienten der Verfahrens können folglich in einem gemeinsamen Butcher-Tableau dargestellt werden:

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & (\mathbf{b}^{(p)})^T \\ \hline & (\mathbf{b}^{(p+1)})^T \end{array} \quad \text{d.h.} \quad \begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \dots & a_{1L} \\ c_2 & a_{21} & a_{22} & \dots & a_{2L} \\ \vdots & \vdots & & \ddots & \vdots \\ c_L & a_{L1} & a_{L2} & \dots & a_{LL} \\ \hline & b_1^{(p)} & b_2^{(p)} & \dots & b_L^{(p)} \\ \hline & b_1^{(p+1)} & b_2^{(p+1)} & \dots & b_L^{(p+1)} \end{array}.$$

Dadurch können alle $\mathbf{k}_i, i = 1, ..., L$ für beide Ordnungen gemeinsam berechnet werden und erst bei der Summierung unterscheiden sich die Verfahren und somit entsteht nur dort doppelter Auswand.
Prominente Vertreter sind folgende Verfahren:

- **Euler/Heun**, Ordnung 1/2:

0		
1	1	
	1	0
	$\frac{1}{2}$	$\frac{1}{2}$

- **Runge-Kutta**, Ordnung 3/4:

0					
$\frac{1}{2}$	$\frac{1}{2}$				
$\frac{1}{2}$	0	$\frac{1}{2}$			
1	0	0	1		
1	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$	0
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$	0
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	0	$\frac{1}{6}$

Für solche eingebettete Verfahren ist die Bestimmung der Schätzung des Konsistenzfehler dann sogar möglich, ohne die Lösungen berechnen zu müssen, denn man findet

$$\begin{aligned} h_k\tilde{\tau}_k &= \mathbf{y}_k^{(p+1)} - \mathbf{y}_k^{(p)} = h_k \left\{ \Phi^{(p+1)}(\mathbf{y}_{k-1}) - \Phi^{(p)}(\mathbf{y}_{k-1}) \right\} \\ &= h_k \left\{ \sum_{l=1}^{L} b_l^{(p)}\ \mathbf{k}_l - \sum_{l=1}^{L} b_l^{(p+1)}\ \mathbf{k}_l \right\} = h_k \sum_{l=1}^{L} (b_l^{(p)} - b_l^{(p+1)})\ \mathbf{k}_l. \end{aligned}$$

Für das Verfahren höherer Ordnung $p+1$ muss man im Allgemeinen mehr Auswertungen von $\mathbf{f}$ machen als beim Verfahren der Ordnung p (dort ist typischer Weise $b_L^{(p)} = 0$, und damit benötigt man nur formal gleich viele Zwischenschritte in beiden Verfahren). Das kann man zum Teil durch den sogenannten Fehlberg-Trick kompensieren. Wählt man nämlich

$$c_L := 1, \quad b_L^{(p)} := 0, \quad a_{Ls} = b_s^{(p)},\ s = 1, \ldots, L-1,$$

so gilt

$$\begin{aligned} \mathbf{k}_L = \mathbf{f}(t_{k-1,L}, \mathbf{y}_{k-1,L}) &= \mathbf{f}(t_{k-1} + c_L h_k, \mathbf{y}_{k-1} + h_k \sum_{s=1}^{L} a_{Ls}\mathbf{k}_s) \\ &= \mathbf{f}(t_{k-1} + h_k, \mathbf{y}_{k-1} + h_k \sum_{s=1}^{L} b_s^{(p)}\mathbf{k}_s) = \mathbf{f}(t_k, \mathbf{y}_k^{(p)}). \end{aligned}$$

Damit ist die Auswertung der letzten Stufe zugleich identisch mit der Auswertung für den ersten Teilschritt im nächsten Zeitschritt und man kann sich diese Auswertung sparen.