Skript zur Vorlesung:
„Numerische Methoden für Differentialgleichungen“

Prof. Dr. P.E. Kloeden
Fachbereich Mathematik

Goethe Universität
Zimmer 101, Robert-Mayer-Straße 10

Telefon: (069) 798 28622 — Sekretariat (069) 798 22422

email: kloeden@math.uni-frankfurt.de

20. Februar 2012

Inhaltsverzeichnis

Kapitel 1

Das Euler-Verfahren

1.1 Gewöhnliche Differentialgleichungen

Literatur Aulbach für Differentialgleichungen und Schwarz: Kap.9, Stummel/Hainer: Kap. 7 für Numerik

Wir beschränken uns hier auf gewöhnlichen Differentialgleichungen erster Ordnung,

$$\frac{dx}{dt} = f(t,x), \quad x \in \mathbb{R}^d, \quad d \geq 1. \tag{1.1}$$

Wegen der Einfachheit werden wir (ohne große Beschränkung der Allgemeinheit) meistens den 1-dimensionalen Fall mit $d = 1$ betrachten.

Eine differenzierbare Funktion

$$x : \ (\alpha, \beta) \to \mathbb{R}^d$$

heißt Lösung der DGL (1.1) auf dem Intervall (α, β), falls

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

für jedes $t \in (\alpha, \beta)$.

Beispiele:

(1) $x(t) = e^{t^2}$ ist eine Lösung von $\frac{dx}{dt} = 2tx$ auf $(-\infty, \infty)$

(2) $x(t) = \frac{1}{1-t}$ ist eine Lösung von $\frac{dx}{dt} = x^2$ auf $(-\infty, 1)$.

Im Allgemeinen besitzt eine DGL erster Ordnung wie (1.1) eine Lösungsschar, die durch d Parameter (tatsächlich, Integrationskonstanten!) beschrieben ist.

Beispiele:

(1) $\frac{dx}{dt} = 2tx$ mit $x_A(t) = Ae^{t^2}$ für alle $t \in \mathbb{R}$ und $A \in \mathbb{R}$

(2) $\frac{dx}{dt} = x^2$ mit $x_A(t) = \frac{1}{A-t}$ für alle $t \in (-\infty, A)$ und $A \in \mathbb{R}$.

1.1.1 Anfangswertaufgabe

Finde eine Lösung $x = x(t)$ der DGL

$$\frac{dx}{dt} = f(t, x),$$

die der Anfangsbedingung

$$x(t_0) = x_0$$

genügt. Diese Aufgabe heißt Anfangswertaufgabe.

Existenz- und Eindeutigkeitssätze geben hinreichende Voraussetzungen für die Existenz einer eindeutigen Lösung einer Anfangswertaufgabe, z.B.

(1) f ist stetig in (t, x); und

(2) f genügt einer Lipschitzbedingung bzgl. x gleichmäßig in t, d.h.

$$|f(t, x) - f(x, y)| \leq L|x - y|.$$

Diese Bedingungen sind ziemlich stark und scliessen viele Anwendungen aus. Andere Möglichkeiten sind dass f stetigdifferenzierbar in (t, x) ist und genügt einer lokalen Lipschitz-Bedingung.

$$|f(t, x) - f(t, y)| \leq L_R|x - y|$$

für $|x|, |y| \leq R$ unabhängig von $t \in [t_0, T]$, aber L_R hängt auch von T ab.

$\Rightarrow$ $\exists$ eine eindeutige Lösung der AWA auf einem Intervall $[t_0, T(t_0, x_0))$.
Aber ein maximales $T(t_0, x_0) < T$ ist möglich.

Zu versichern, dass $T(t_0, x_0) \geq T$ ist, brauchen wir zusätzliche Voraussetzungen, z.B. f genügt einer Linearwachstums-Bedingung

$$|f(t,x)| \leq K(1+|x|), \qquad \forall x \in \mathbb{R}^d, \quad t \in [t_0, T]$$

oder einer Dissipativitäts-Bedingung wie

$$< x, f(t,x) > \quad \leq \quad K - L|x|^2, \qquad \forall x \in \mathbb{R}^d, \quad t \in [t_0, T].$$

Wir werden immer voraussetzen, dass die AWA eine eindeutige Lösung auf dem ganzen gegebenen Intervall $[t_0, T]$ besitzt. Diese Lösung ist meistens nicht explizit bekannt und wir müssen sie numerisch approximieren.

Andere Voraussetzungen versichern globale Existenz, d.h. für alle $t \geq t_0$, z.B. lineare Wachstumsbeschränkung

$$|f(t,x)| \leq K(1+|x|)$$

$\Rightarrow$ die Lösungen können nicht zu schnell oder steil aufsteigen.

Bemerkungen:

Eine Lösung $x = x(t)$ der DGL $\dfrac{dx}{dt} = f(t,x)$ genügt der Integralgleichung

$$x(t) = x_0 + \int_{t_o}^{t} f(s, x(s))ds$$

sowie der Differentialgleichung

$$\frac{d}{dt}x(t) = f(t, x(t)).$$

Hauptsatz der Integralrechnung!

Man beweist den Existenz- und Eindeutigkeitssatz mit dem Fixpunktsatz oder einer Folge sukzessiver Approximationen

$$x_{n+1}(t) = x_0 + \int_{t_0}^{t} f(s, x_n(s))ds, \qquad n = 0, 1, \ldots$$

mit $x_0(t) \equiv x_0$. (Theoretisch gut, in der Praxis fast nutzlos!).

1.2 Das Euler-Verfahren

Wir schreiben $x(t, t_0, x_0)$ für die eindeutige Lösung einer Anfangswertaufgabe

$$\frac{dx}{dt} = f(t, x), \quad x(t_0) = x_0$$

(falls sie existiert!). Eine solche Lösung ist meistens nicht explizit analytisch bekannt. Die Methode sukzessiver Approximationen ist nicht praktisch. Deshalb versuchen wir die Lösung numerisch zu approximieren. Die einfachste numerische Methode heißt Euler-Verfahren

$$x_{n+1} = x_n + h_n f(t_n, x_n), \quad n = 0, 1, \ldots$$

mit Schrittweite $h_n = t_{n+1} - t_n > 0$ für eine Zerlegung

$$t_0 < t_1 < \ldots < t_n < t_{n+1} < \ldots < t_N = T$$

von $[t_0, T]$.

Das Euler-Verfahren ist eine Differenzengleichung erster Ordnung oder 1-Schrittverfahren, d.h. x_{n+1} hängt direkt nur von x_n ab.

1.2.1 Motivierung für das Euler-Verfahren

Betrachte die Lösung $x(t) = x(t, t_0, x_0)$ auf dem Teilintervall $[t_n, t_{n+1}]$.

(1) Differentialgleichungsversion

$$\frac{d}{dt}x(t)|_{t=t_n} = f(t_n, x(t_n))$$

mit

$$\frac{d}{dt}x(t)|_{t=t_n} \simeq \frac{x(t_{n+1}) - x(t_n)}{t_{n+1} - t_n}$$

$$\Rightarrow \quad x(t_{n+1}) \simeq x(t_n) + (t_{n+1} - t_n) f(t_n, x(t_n))$$

(2) Integralgleichungsversion

$$\begin{aligned} x(t_{n+1}) &= x(t_n) + \int_{t_n}^{t_{n+1}} f(s, x(s)) ds \\ &\simeq x(t_n) + \int_{t_n}^{t_{n+1}} \underbrace{f(t_n, x(t_n))}_{\substack{\text{Integrand gefroren} \\ \text{zum Wert mit } s = t_n}} ds \\ &= x(t_n) + (t_{n+1} - t_n) f(t_n, x(t_n)) \end{aligned}$$

1.3 Diskretisierungsfehler

Wir haben nur eine Approximation

$$x_n \simeq x(t_n, t_0, x_0)$$

und wollen den Diskretisierungsfehler (DF) abschätzen. Dafür beschränken wir uns (wegen der Einfachheit) zum Fall mit konstanter Schrittweite

$$h_n = t_{n+1} - t_n \equiv h > 0, \text{ Konstante}, \quad \text{für alle } n.$$

Wir unterscheiden zwischen den globalen und den lokalen Diskretisierungsfehlern.

Der globale Diskretisierungsfehler lautet

$$G_n(h) := |x(t_n, t_0, x_0) - x_n|, \quad n = 0, 1, \ldots, N_h := \frac{T - t_0}{h}$$

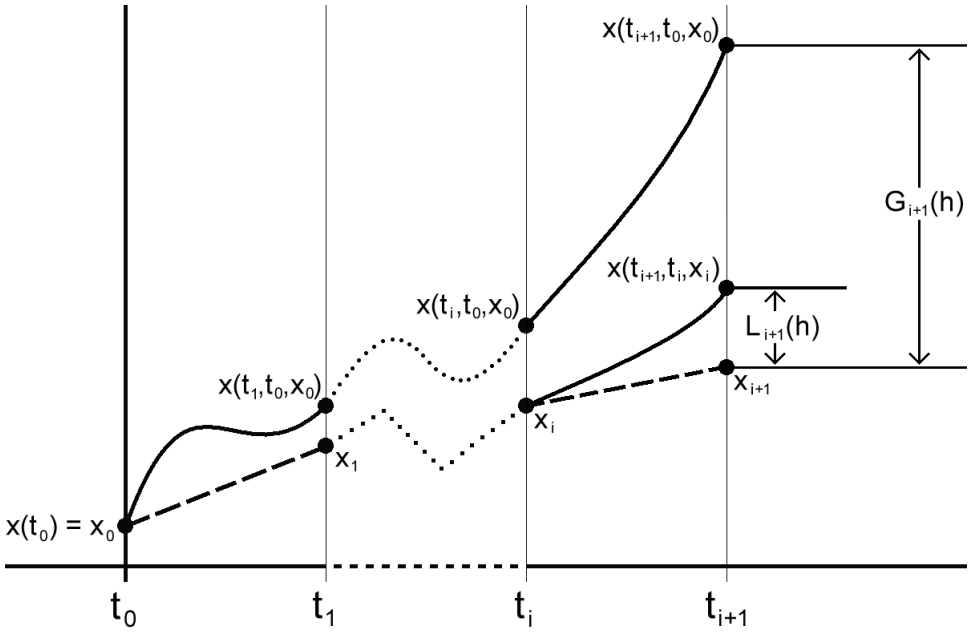
Wir sagen, dass die numerische Approximation konvergiert, falls

$$\lim_{h \to 0} \max_{0 \le n \le N_h} G_n(h) = 0.$$

Der lokale Diskretisierungsfehler lautet

$$L_{n+1}(h) := |x(t_{n+1}, t_n, x_n) - x_{n+1}|, \qquad n = 0, 1, \ldots, N_h - 1$$

Im Allgemeinen ist $L_{n+1}(h) \neq G_{n+1}(h)$, weil die Lösung $x(t, t_n, x_n)$ mit Anfangswert $x(t_n) = x_n$ nicht die gesuchte Lösung $x(t, t_0, x_0)$ ist. Aber der lokale DF ist günstig, weil wir ihn durch eine Taylor-Entwicklung leicht abschätzen können – dann können wir die Abschzätzung benutzen, um den globalen DF abzuschätzen.



Satz: *Sei f stetig differenzierbar. Dann gilt*

$$L_n(h) \leq K_T h^2$$

für $n = 0, 1, \ldots, N_h - 1$, wobei K_T eine geeignete Konstante ist.

Beweis:

Betrachte die Taylor-Entwicklung der Lösung $x(t) = x(t, t_n, x_n)$, d.h.

$$x(t) = x(t_n) + x'(t_n)\ (t - t_n) + \frac{1}{2!} x''(\theta_{n,t}) \cdot (t - t_n)^2$$

mit $\theta_{n,t} \in [t_n, t)$

Insbesondere für $t = t_{n+1}$ gilt

$$\begin{aligned} x(t_{n+1}; t_n, x_n) &= x(t_n) + x'(t_n)\ (t_{n+1} - t_n) + \frac{1}{2} x''(\tilde{\theta}_n)\ (t_{n+1} - t_n)^2 \\ &= \underbrace{x_n + h f(t_n, x_n)}_{x_{n+1}} + \frac{1}{2} x''(\tilde{\theta}_n) h^2 \end{aligned}$$

mit $\tilde{\theta}_n \in [t_n, t_{n+1}]$.

Davon aus haben wir

$$L_{n+1} = |x(t_{n+1}; t_n, x_n) - x_{n+1}| = \frac{1}{2} h^2 \cdot |x''(\tilde{\theta}_n|.$$

Dies sieht nutzlos aus – wie können wir $x''(t)$ kennen, wenn $x(t)$ unbekannt ist? Aber wir können $x''(t)$ abschätzen

$$\begin{aligned} x''(t) = \frac{d}{dt}x'(t) &= \frac{d}{dt}f(t,x(t)) \\ &= \frac{\partial f}{\partial t}(t,x(x)) + \frac{\partial f}{\partial x}(t,x(t))\ \frac{d}{dt}x(t) \\ &= \frac{\partial f}{\partial t}(t,x(t)) + \frac{\partial f}{\partial x}(t,x(t))\ f(t,x(t)) \end{aligned}$$

Die Lösung $x(t)$ ist unbekannt, aber wegen Stetigkeit bleibt sie beschränkt für $t \in [t_0, T]$. Daher existiert eine kompakte Teilmenge $B \subset \mathbb{R}^d$ mit $x(t) \in B$ für jedes $t \in [t_0, T]$. Die Funktionen f, $\frac{\partial f}{\partial t}$, $\frac{\partial f}{\partial x}$ sind stetig und deswegen gleichmäßig beschränkt für $(t,x) \in [t_0, T] \times B$. Daher haben wir die folgende Abschätzung

$$\begin{aligned} \left|x''(\tilde{\theta}_n\right| &\leq \max_{t_0 \leq t \leq T} |x''(t)| \\ &\leq \max_{(t,x)\in[t_0,T]\times B} \left\{ \left|\frac{\partial f}{\partial t}(t,x)\right| + \left|\frac{\partial f}{\partial x}(t,x)\right| \cdot |f(t,x)| \right\} \\ &=: K_T < \infty \end{aligned}$$

Davon aus folgt die gesuchte Abschätzung

$$L_n(n) \leq \frac{1}{2} K_T h^2 \qquad \text{oder} \quad L_n(h) \sim 0(h^2).$$

Bemerkung: K_T ist meistens unbekannt, aber wir arbeiten hier nur theoretisch – die Ordnung ist wichtig.

Satz: *Sei f stetig differenzierbar. Dann existiert eine Konstante $C_T > 0$ mit*

$$G_n(h) \leq C_T h$$

für $n = 0, 1, \ldots, N_h$.

D.h. das Euler–Verfahren hat einen globalen Diskretisierungsfehler erster Ordnung.

Beweis: Sei B eine kompakte Teilmenge von $\mathbb{R}^d$, die die unten erscheinenden Lösungen enthält. Wir können dann gleichmäßige Abschätzungen auf

$[t_0, T] \times B$ benutzen. Insbesondere genügt f einer Lipschitzbedingung

$$|f(t,x) - f(t,y)| \leq L|x-y|$$

gleichmäßig in $t \in [t_0, T]$ für $x, y \in B$ [Mittelwertsatz für Ableitungen, hier für $\frac{\partial f}{\partial x}(t,x)$, und Stetigkeit.]

Wir werden eine Differenzenungleichung für die sukzessiven G_n herleiten. Mit $x(t) = x(t, t_0, x_0)$ haben wir

$$\begin{aligned}
G_{n+1}(h) &= |x(t_{n+1}) - x_{n+1}| \\
&\leq |x(t_{n+1}) - x(t_n) - hf(t_n, x(t_n))| \qquad \text{lokaler DF} \\
&\quad + |x(t_n) - hf(t_n, x(t_n)) - x_{n+1}| \qquad \underbrace{x_{n+1} = x_n + hf(t_n, x_n)}_{\text{Euler Verfahren}} \\
&\leq L_{n+1}(h) + \underbrace{|x(t_n) - x_n|}_{G_n} + \underbrace{h\,|f(t_n, x(t_n)) - f(t_n, x_n)|}_{\text{Lipschitz: } \leq hL|x(t_n) - x_n|} \\
&\leq \frac{1}{2} K_T\, h^2 + (1 + Lh) G_n(h)
\end{aligned}$$

d.h.

$$G_{n+1}(h) \leq (1 + Lh) G_n(h) + \frac{1}{2} K_T h^2.$$

Definiere jetzt $\alpha = 1 + Lh$ und $\beta = \frac{1}{2} K_T h^2 > 0$.

$$\Rightarrow \quad G_{n+1}(h) \leq \alpha G_n(h) + \beta.$$

Durch Induktion gilt

$$G_n(h) \leq \alpha^n \underbrace{G_0(h)}_{= 0 \text{ hier}} + \beta(1 + \alpha + \ldots + \alpha^{n-1})$$

d.h.

$$\begin{aligned}
G_n(h) &\leq \left(1 + \alpha + \ldots + \alpha^{n-1}\right) \beta \\
&= \frac{\alpha^n - 1}{\alpha - 1}\, \beta \\
&= \frac{(1 + Lh)^n - 1}{1 + Lh - 1} \cdot \frac{1}{2} K_T h^2
\end{aligned}$$

$$\leq (1+Lh)^n \cdot \frac{K_T}{2L} \cdot h^1 \qquad \text{hier verlieren wir eine Potenz von } h$$

$$\leq e^{nhL} \frac{K_T}{2L} \cdot h$$

$$\leq e^{L(T-t_0)} \cdot \frac{K_T}{2L} \cdot h$$

d.h.

$$G_n(h) \leq C_T h \qquad \text{mit } C_T = e^{L(T-t_0)} \frac{K_T}{2L}.$$

Kapitel 2

1-Schrittverfahren höherer Ordnung

Wir betrachten eine Anfangswertaufgabe (AWA)

$$\frac{dx}{dt} = f(t,x), \qquad x(t_0) = x_0$$

mit Lösung $x(t) = x(t, t_0, x_0)$.

Das Euler-Verfahren mit konstanter Schrittweite $t_{n+1} - t_n \equiv h > 0$ lautet

$$x_{n+1} = x_n + hf(t_n, x_n), \qquad n = 0, 1, \ldots, \ .$$

Es ist ein explizites 1-Schrittverfahren mit globalem Diskretisierungsfehler erster Ordnung:

$$G_n(h) = |x(t_n, t_0, x_0) - x_n| \sim 0(h) \ .$$

Wir wollen jetzt 1-Schrittverfahren höherer Ordnung herleiten. Wie im Euler-Fall können wir entweder die Differentialversion oder die Integralversion der AWA betrachten.

2.1 Taylor-Verfahren

Betrachte die Taylor-Entwicklung der Lösung $x(t) = x(t, t_0, x_0)$ der AWA auf dem Teilintervall $[t_n t_{n+1}]$:

$$x_(t_{n+1}) = x(t_n) + x'(t_n)\ h + \ldots + \frac{1}{p!}x^{(p)}(t_n)\ h^p + \frac{1}{(p+1)!}x^{(p+1)}(\theta_n)\ h^{p+1}$$

mit Zwischenwert $\theta_n \in [t_n, t_{n+1}]$ in dem (letzten) Restterm.

Definiere den Differentialoperator D durch

$$DG(t,x) := \frac{\partial g}{\partial t}(t,x) + f(t,x)\frac{\partial g}{\partial x}(t,x)$$

d.h. $Dg(t,x(t))$ ist die totale Ableitung von $g(t,x(t))$ bzgl. einer Lösung $x(t)$ der DGL

$$x'(t) = \frac{d}{dt}x(t) = f(t,x(t))$$

d.h., wegen der Kettenregel haben wir

$$\frac{d}{dt}g(t,x(t)) = \frac{\partial g}{\partial t}(t,x(t)) + \frac{\partial g}{\partial x}(t,x(t))\ x'(t) = Dg(t,x(t)).$$

Für eine solche Lösung gilt

$$\begin{aligned} x'(t) &= f(t,x(t)) \\ x''(t) &= \frac{d}{dt}x'(t) = \frac{d}{dt}f(t,x(t)) = Df(t,x(t)) \\ x'''(t) &= \frac{d}{dt}x''(t) = \frac{d}{dt}Df(t,x(t)) = D^2f(t,x(t)) \ , \end{aligned}$$

und im Allgemeinen (falls f glatt genug ist)

$$x^{(j)}(t) = D^{j-1}f(t,x(t)), \qquad j = 1,2,\dots \ .$$

z.B.

$$\begin{aligned} Df &= \frac{\partial f}{\partial t} + f\frac{\partial f}{\partial x} \\ D^2f &= D[Df] = \frac{\partial}{\partial t}[Df] + f\frac{\partial}{\partial x}[Df] \\ &= \frac{\partial}{\partial t}\left\{\frac{\partial f}{\partial t} + f\frac{\partial f}{\partial x}\right\} + f\frac{\partial}{\partial x}\left\{\frac{\partial f}{\partial t} + f\frac{\partial f}{\partial x}\right\} \\ &= \frac{\partial^2 f}{\partial t^2} + \frac{\partial f}{\partial t}\ \frac{\partial f}{\partial x} + f\frac{\partial^2 f}{\partial t \partial x} + f\frac{\partial^2 f}{\partial x \partial t} \\ &\qquad + f\left(\frac{\partial f}{\partial x}\right)^2 + f^2\frac{\partial^2 f}{\partial x^2} \\ D^3f &= D[D^2f] = \frac{\partial}{\partial t}\{D^2f\} + f\frac{\partial}{\partial x}\{D^2f\} \end{aligned}$$

Ein Job für MAPLE !

Die obige Taylor-Entwicklung lautet

$$x(t_{n+1}) = x(t_n) + \sum_{j=1}^{p} \frac{1}{j!} D^{j-1} f(t_n, x(t_n))\ h^j + \frac{1}{(p+1)!} D^p f(\theta_n, x(\theta_n))\ h^{p+1}$$

Davon erhalten wir das Taylor-Verfahren p-ter Ordnung

$$\boxed{x_{n+1} = x_n + \sum_{j=1}^{p} \frac{h^j}{j!} D^{j-1} f(t_n, x_n)}$$

Beispiel $p = 1 \Rightarrow\ x_{n+1} = x_n + hf(t_n, x_n)$ Euler-Verfahren!

Hier lautet der lokale Diskretisierungsfehler

$$\begin{aligned} L_{n+1} &:= |x(t_{n+1}, t_n, x_n) - x_{n+1}| \\ &= \frac{h^{p+1}}{(p+1)!} |D^p f(\theta_n, x(\theta_n; t_n, x_n))| \\ &\sim 0(h^p) \end{aligned}$$

(Im Prinzip können wir $D^p f(t, x)$ abschätzen.)

Wie im Euler-Fall verlieren wir dann eine Potenz zwischen dem lokalen und globalen DF (siehe Satz später) und erhalten

$$G_n(h) \sim O(h^p)$$

d.h. das Taylor-Verfahren p-ter Ordnung ist von p-ter Ordnung!

ABER die höheren Koeffizienten $D^{j-1} f(t, x)$ eines Taylor-Verfahrens sind sehr kompliziert. Deswegen sind solche Verfahren fast nie in der Praxis benutzt, aber sie sind theoretisch sehr nützlich.

2.2 1-Schrittverfahren höherer Ordnung ohne Ableitungen

Wir betrachten jetzt die Integralgleichungsdarstellung der Lösung der AWA auf einem Teilintervall $[t_n, t_{n+1}]$, d.h.

$$x(t_{n+1}) = x(t_n) + \int_{t_n}^{t_{n+1}} f(s, x(s))\, ds$$

und wir werden dann versuchen, das Integral hier zu approximieren.

Rechteckregel

$$x(x_{n+1}) \simeq x(t_n) + \int_{t_n}^{t_{n+1}} f(t_n, x(t_n))ds, \qquad (s \equiv t_n \text{ in dem Integrand hier})$$

d.h. $\quad x(t_{n+1}) \simeq x(t_n) + h_n f(t_n, x(t_n))$

$\Rightarrow$ Euler-Verfahren $\quad x_{n+1} = x_n + h_n f(t_n, x_n)$

oder wir können den anderen Randpunkt wählen

$$x(t_{n+1}) \simeq x(t_n) + \int_{t_n}^{t_{n+1}} f(t_{n+1}, x(t_{n+1}))ds, \qquad (s \equiv t_{n+1} \text{ in dem Integrand hier})$$

d.h.

$$x(t_{n+1}) \simeq x(t_n) + h_n f(t_{n+1}, x(t_{n+1}))$$

$\Rightarrow$ Das implizite Euler-Verfahren

$$x_{n+1} = x_n + h_n f(t_{n+1}, x_{n+1})$$

Dies sieht kompliziert aus: für jedes n müssen wir eine algebraische Gleichung lösen, z.B. mit der Newton-Methode. Warum? Es gibt Vorteile, z.B. numerisch stabiler. (Siehe nächste Vorlesung!).

Trapezregel

$$\begin{aligned} x(t_{n+1}) &= x(t_n) + \int_{t_n}^{t_{n+1}} f(s, x(s))\, ds \\ &\simeq x(t_n) + \frac{1}{2} h_n \left[f(t_n, x(t_n)) + f(t_{n+1}, x(t_{n+1})) \right] \end{aligned}$$

$\Rightarrow$ Trapez-Verfahren

$$x_{n+1} = x_n + \frac{1}{2}h_n[f(t_n, x_n) + f(t_{n+1}, x_{n+1})],$$

das auch ein implizites 1-Schrittverfahren ist.

Approximieren wir das „x_{n+1}“ an der rechten Seite durch das „x_{n+1}“ des entsprechenden Euler-Verfahrens, d.h. durch $x_{n+1} = x_n + h_n f(t_n, x_n)$, dann erhalten wir das Heun-Verfahren,

$$x_{n+1} = x_n + \frac{1}{2}h_n[f(t_n, x_n) + f(t_{n+1}, x_n + h_n f(t_n, x_n))]$$

d.h. ein explizites 1-Schrittverfahren.

Satz: *Das Heun-Verfahren hat lokalen Diskretisierungsfehler dritter Ordnung, falls f zweimal stetig differenzierbar ist.*

Beweis: Schreibe h statt h_n. Dann haben wir

$$x_{n+1} = x_n + \frac{1}{2}hf(t_n, x_n) + \underbrace{\frac{1}{2}f(t_n + h, x_n + hf(t_n, x_n))}_{\text{Taylor-Entwicklung um } (t_n, x_n)}$$

d.h.

$$\begin{aligned} x_{n+1} &= x_n + \frac{1}{2}hf(t_n, x_n) \\ &\quad + \frac{1}{2}h\Big\{ f(t_n, x_n) + \frac{\partial f}{\partial t}(t_n, x_n)h + \frac{\partial f}{\partial x}(t_n, x_n)hf(t_n, x_n) \\ &\qquad + \frac{1}{2!}\,\frac{\partial^2}{\partial t^2}\,(t_n, x_n)h^2 + \frac{\partial^2}{\partial t \partial x}\,(t_n, x_n)h\; hf(t_n, x_n) \\ &\qquad + \frac{1}{2!}\,\frac{\partial^2 f}{\partial x^2}\,(t_n, x_n)h^2 f(t_n, x_n)^2 + 0(h^3)\Big\} \\ &= \underbrace{x_n + hf(t_n, x_n) + \frac{1}{2}h^2 Df(t_n, x_n)}_{x_{n+1} \text{ für das Taylor-Verfahren mit } p=2} + 0(h^3) \end{aligned}$$

d.h. $x_{n+1}^{(\text{Heun})} = x_{n+1}^{(\text{Taylor})} + 0(h^3)$ mit demselben x_n

$\Rightarrow$ $L_{n+1}(h) = \left| x(t_{n+1}, t_n, x_n) - x_{n+1}^{(\text{Heun})} \right|$

$$\leq \underbrace{\left|x(t_{n+1},t_n,x_n)-x_{n+1}^{(\text{Taylor})}\right|}_{\text{Lokaler DF des Taylor-Verfahrens}} + \left|x_{n+1}^{(\text{Taylor})}-x_{n+1}^{(\text{Heun})}\right|$$

$$\sim \quad 0(h^3)+0(h^3)=0(h^3)$$

d.h. das Heun-Verfahren hat 2-te Ordnung – wir verlieren eine Potenz in dem globalen DF; siehe den folgenden Satz.

Das Heun-Verfahren ist ein einfaches Beispiel aus der Familie der Runge-Kutta-Verfahren.

2.3 Allgemeine 1-Schrittverfahren

Ein 1-Schrittverfahren hat die allgemeine Form

$$\boxed{x_{n+1}=x_n+h_n\Phi(h_n,t_n,x_n,x_{n+1})}$$

z.B.

(1) explizites Euler-Verfahren

$$\Phi(h,t,x,y)=f(t,x)$$

(2) implizites Euler-Verfahren

$$\Phi(h,t,x,y)=f(t+h,y)$$

(3) Trapez-Verfahren

$$\Phi(h,t,x,y)=\frac{1}{2}[f(t,x)+f(t+h,y)]$$

(4) Heun-Verfahren

$$\Phi(h,t,x,y)=\frac{1}{2}[f(t,x)+f(t+h,x+hf(t,x))]$$

(5) Taylor-Verfahren p-ter Ordnung

$$\Phi(h,t,x,y)=\sum_{j=1}^{p}\frac{h^{j-1}}{j!}\,D^{j-1}f(t,x)$$

In expliziten Fällen schreiben wir

$$\boxed{x_{n+1} = x_n + h_n \Phi(h_n, t_n, x_n)}$$

d.h. ohne x_{n+1} oder y innerhalb Φ.

Satz: *Betrachte ein explizites* 1*-Schrittverfahren*

$$x_{n+1} = x_n + h_n \Phi(h_n, t_n, x_n)$$

mit lokalem Diskretisierungsfehler $(p+1)$*-ter Ordnung, wobei* $\Phi(h,t,x)$ *einer Lipschitzbedingung bzgl.* x *gleichmäßig in* (h,t) *genügt.*

Dann hat der globale Diskretisierungsfehler p*-te Ordnung.*

Beweis: Schreibe h statt h_n. Dann gilt

$$\begin{aligned}
G_{n+1}(h) &= |x(t_{n+1}) - x_{n+1}| \\
&\leq |x(t_{n+1}) - x(t_n) - h\Phi(h, t_n, x(t_n))| \qquad \text{lokaler DF} \\
&\quad + \Big| x(t_n) + h\Phi(h, t_n, x(t_n)) - \underbrace{x_n - h\Phi(h, t_n, x_n)}_{x_{n+1}} \Big| \\
&\leq L_{n+1}(h) + |x(t_n) - x_n| + \underbrace{h\,|\Phi(h, t_n, x(t_n) - \Phi(h, t_n, x_n)|}_{\text{Lipschitz: } \leq hL|x(t_n) - x_n|} \\
&\leq L_{n+1}(h) + (1 + Lh)\underbrace{|x(t_n) - x_n|}_{G_n(h)} \\
&\leq K_T\, h^{p+1} + (1 + Lh) G_n(h)
\end{aligned}$$

d.h.

$$\boxed{G_{n+1}(h) \leq (1 + Lh) G_n(h) + K_T\, h^{p+1}}$$

Dann wie im Euler-Fall gilt es

$$\begin{aligned}
G_n(h) &\leq \frac{(1 + Lh)^n - 1}{(1 + Lh) - 1} K_T h^{p+1} \\
&\leq (1 + Lh)^n \frac{K_T}{L} h^{\overbrace{p + 1 - 1}^{=p}} \\
&\leq e^{L(T - t_0)} \frac{K_T}{L} h^p
\end{aligned}$$

d.h. $G_n(h) \sim 0(h^p)$

Bemerkungen:

(1) Das entsprechende Ergebnis ist auch im impliziten Fall gültig – dann soll $\Phi(h,t,x,y)$ einer Lipschitzbedingung bzgl. (x,y) gleichmäßig in (h,t) genügen.

(2) Wir müssen zeigen, dass Φ einer solchen Liptschitzbedingung genügt – diese folgt von der Glattheit von f, z.B. f soll p-mal stetig differenzierbar sein.

2.4 Konsistenz

Durch den Begriff Konsistenz können wir schnell bestätigen, ob ein 1-Schrittverfahren

$$x_{n+1} = x_n + h\Phi(h, t_n, x_n, x_{n+1})$$

konvergiert oder nicht konvergiert. Deshalb können wir vermeiden, den Diskretisierungsfehler abzuschätzen – eine mühsame und aufwendige Aufgabe!

Ein 1-Schrittverfahren heißt konsistent, falls

$$\lim_{h \downarrow 0} \frac{L(h)}{h} = 0,$$

wobei $L(h)$ der lokale Diskretisierungsfehler ist, oder äquivalent, falls

$$\lim_{h \downarrow 0} \Phi(h, t, x(t), x(t+h)) = f(t, x(t))$$

für jede Lösung der DGL, d.h. falls

$$\Phi(0, t, x, x) \equiv f(t, x)$$

ist (für Φ stetig bzgl. allen Variablen).

Der Grund dafür ist, dass wir eine Potenz der Ordnung zwischen dem lokalen und dem globalen DF verlieren. Auch gilt es

$$\begin{aligned} \frac{L(h)}{h} &= \frac{1}{h}\{x(t+h) - x(t) - h\Phi(h, t, x(t), x(t+h))\} \\ &= \left\{\frac{x(t+h) - x(t)}{h} - f(t, x(t))\right\} + \{f(t, x(t)) - \Phi(h, t, x(t), x(t+h))\} \end{aligned}$$

$\Rightarrow$ die Bedingungen oben sind äquivalent.
Die letzte Bedingung

$$\Phi(0,t,x,x) \equiv f(t,x)$$

ist ganz einfach zu prüfen.

„Satz“ *Sei Φ Lipschitz. Dann ist Konsistenz $\Leftrightarrow$ Konvergenz*

Der Beweis in die $\Rightarrow$ Richtung folgt von der Definition usw, in die $\Leftarrow$ Richtung ist komplizierter, siehe die Lehrbücher.

Beispiele:

1) Heun-Verfahren

$$\begin{aligned}\Phi(h,t,x,y) &= \frac{1}{2}f(t,x) + \frac{1}{2}f(t+h, x+hf(t,x)) \\ &\to \frac{1}{2}f(t,x) + \frac{1}{2}f(t,x) = f(t,x)\end{aligned}$$

für $h \to 0$, $\forall\, x$, d.h. $\Phi(0,t,x,x) = f(t,x)$

2) implizites Euler-Verfahren

$$\Phi(h,t,x,y) = f(t+h,y) \to f(t,x)$$

für $h \to 0$ und $y \to x$.

Zunächst sollen wir Konsistenz bestätigen, und nur dann wird es wertvoll sein, die Ordnung des lokalen (globalen) DFs abzuschätzen.

2.5 Rundungsfehler in 1-Schrittverfahren

Literatur Schwarz: Kap. 9.3.1, Stummel/Hainer: Kap. 11.4

Betrachte ein 1-Schrittverfahren p-ter Ordnung

$$x_{n+1} = x_n + h\Phi(h,t_n,x_n).$$

Statt x_{n+1} berechnen wir $\widetilde{x}_{n+1}$, wobei

$$|\widetilde{x}_{n+1} - x_{n+1}| \le \varepsilon \ll 1$$

Dann haben wir einen numerischen lokalen Diskretisierungsfehler

$$\begin{aligned}\widetilde{L}_{n+1}(h) &= |x(t_{n+1};t_n,x_n) - \widetilde{x}_{n+1}| \\ &\leq \underbrace{|x(t_{n+1};t_n,x_n) - x_{n+1}|}_{\substack{\text{theoretischer} \\ \text{lokaler DF}}} + \underbrace{|x_{n+1} - \widetilde{x}_{n+1}|}_{\text{numerischer Fehler}} \\ &\leq K_T h^{p+1} + \varepsilon\end{aligned}$$

Wir haben auch einen numerischen globalen DF

$$\widetilde{G}(h) = |x(t_n;t_0,x_0) - \widetilde{x}_n|.$$

Dieser genügt der Differenzenungleichung

$$\widetilde{G}_{n+1}(h) \leq (1+Lh)\widetilde{G}_n(h) + \widetilde{L}_{n+1}(h)$$

d.h.

$$\widetilde{G}_{n+1}(h) \leq (1+Lh)\widetilde{G}_n(h) + [K_T h^{p+1} + \varepsilon]$$

Wie vorher folgt es dann, dass

$$\widetilde{G}(h) \leq \frac{(1+Lh)^n - 1}{(1+Lh) - 1}(K_T h^{p+1} + \varepsilon)$$

weil $\widetilde{G}(h) = 0$, d.h. wir verlieren eine Potenz und erhalten

$$\widetilde{G}_n(h) \leq \frac{1}{L} e^{(T-t_0)L}(K_T h^p + \varepsilon/h)$$

d.h., wie numerische Differentiation! Siehe Stummel/Hainer, Seite 265, Fig. 12.

Der wesentliche Fehler $\widetilde{G}_n(h)$ ist oft viel kleiner – die Abschätzung ist eine obere Abschätzung und der (ε/h)-Wert ist der schlimmste Fall.

Aber wir sollen aufpassen!

2.5.1 Numerische Instabilität und implizite Verhfahren

Implizite Verfahren sind oft numerisch stabiler für größere Schrittweiten als explizite Verfahren.

Betrachte die skalare DGL

$$\frac{dx}{dt} = -10^N x$$

wobei $N \gg 1$ ist. Die Lösung mit dem Anfangswert $x(0) = 1$ lautet

$$x(t) = e^{-10^N t}.$$

Sie fällt sehr schnell monoton gegen 0 ab.

Das explizite Euler-Verfahren für diese DGL lautet

$$x_{n+1} = x_n + h_n \left(-10^N h_n\right) x_n = (1 - 10^N h_n) x_n$$

Sei $h_n \equiv h$ und $x_0 = 1$. Dann gilt

$$x_n = \left(1 - 10^N h\right)^n, \qquad n = 0, 1, \ldots.$$

Aber x_n fällt monoton gegen 0 ab, nur für

$$0 < 1 - 10^N h < 1 \qquad (h > 0 \text{ hier})$$

d.h. $0 < h < 10^{-N}$.

ABER Diese Schrittweiten könnten kleiner als das Maschinen-Epsilon sein (Rechner-Arithmetik), wenn N sehr groß ist. Schwierigkeiten ergeben sich, wenn wir versuchen, eine größere Schrittweite zu benutzen, zum Beispiel

(1) $$10^{-N} < h < 2 * 10^{-N} \quad \Leftrightarrow \quad -1 < 1 - 10^N h < 0$$

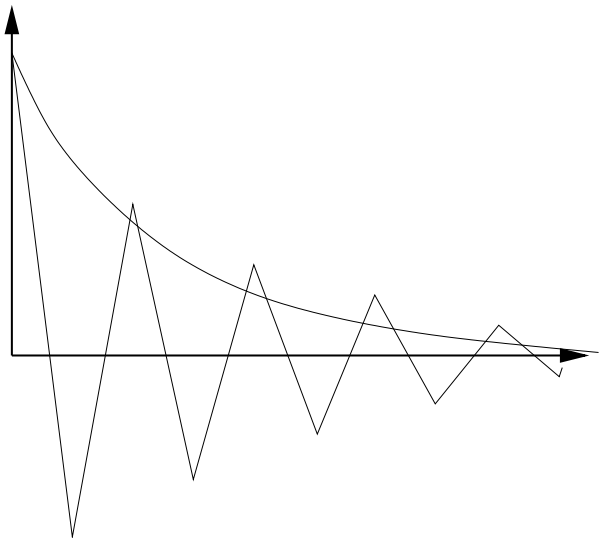
dann $x_n \to 0$ für $n \to 0$ mit Schwankungen wechselndes Vorzeichens

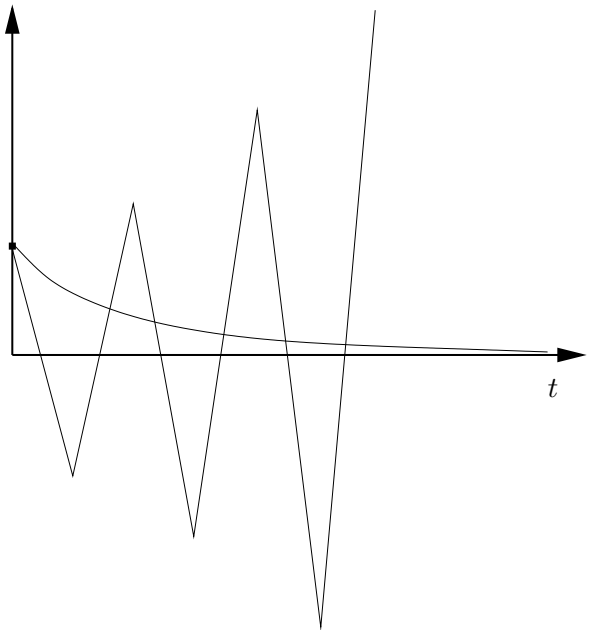
(2) $$2 * 10^{-N} < h \quad \Leftrightarrow \quad 1 - 10^N h < -1$$

dann $|x_n| \to \infty$ für $n \to 0$ mit Schwankungen wechselndes Vorzeichens

Es geht besser für das implizite Euler-Verfahren

$$x_{n+1} = x_n + h_n(-10^N x_{n+1}) \qquad \Rightarrow \quad x_{n+1} = \frac{1}{1 + 10^N h_n}\, x_n$$




t

Für $x_0 = 1$ und $h_n \equiv h$

$$\Rightarrow \quad x_n = \frac{1}{(1+10^N h)^n} \to 0 \qquad \underline{\text{monoton}} \text{ für alle } \quad h > 0$$

Natürlich für Konvergenz müssen wir h klein genug nehmen – aber <u>nicht</u>, um zu versichern, dass das dynamische Verhalten des Verfahrens geeignet ist. Linearisierte DGLen der obigen Art erheben sich oft im Zusammenhang mit Untersuchungen der Fortpflanzung von Abrundungsfehlern. sein!

Kapitel 3

Runge-Kutta-Verfahren

Runge-Kutta-Verfahren gehören zur Klasse der ableitungsfreien Einschrittverfahren, die die Auswertung der Vektorfeldfunktion an mehreren Zwischenzeitstellen des Diskretisierungsteilintervalls benutzen.

Betrachte eine AWA

$$\begin{cases} \dfrac{dx}{dt} &= f(t,x) \\ x(t_0) &= x_0 \end{cases} \qquad x \in \mathbb{R},\ t \in [t_0, T],$$

und eine Unterteilung

$$t_0 < t_1 < \ldots < t_n < \ldots < t_N = T$$

des Intervalls $[t_0, T]$. Schreibe: $h_n = t_{n+1} - t_n > 0$ für die Schrittweite.

Die Lösung $x(t)$ der obigen AWA genügt der Integralgleichung

$$x(t_{n+1}) = x(t_n) + \int_{t_n}^{t_{n+1}} f(t, x(t))\, dt$$

Definiere $g(t) := f(t, x(t))$ für $t \in [t_n, t_{n+1}]$. Wir kennen viele Approximationsformeln für ein Integral wie $\int_{t_n}^{t_{n+1}} g(t)\ dt$, z.B., die Newton-Cotes-Formel und Gauß-Quadratur-Formel:

$$\int_{t_n}^{t_{n+1}} g(t)\, dt \simeq h_n \sum_{j=1}^{s} \alpha_j g(t_n + c_j h_n)$$

mit Auswertungsstellen

$$t_n \le t_n + c_1 h_n < \ldots < t_n + c_j h_n < \ldots < t_n + c_s h_n \le t_{n+1}$$

(deshalb brauchen wir $0 \le c_1 < \ldots < c_j < \ldots < c_s \le 1$)

Mit $\sum_j \alpha_j = 1$ erhalten wir eine Approximation der obigen Integralgleichung:

$$x(t_{n+1}) \simeq x(t_n) + h_n \sum_{j=1}^{s} \alpha_j f\left(t_n + c_j h_n, x(t_n + c_j h_n)\right).$$

Um ein Einschrittverfahren herzuleiten, müssen wir die

$$x(t_n + c_j h_n) = x(t_n) + \int_{t_n}^{t_n + c_j h_n} f(t, x(t))\, dt, \qquad j = 1, \ldots, s$$

ersetzen, durch Formeln, die nur $x(t_n)$ und $x(t_{n+1})$ enthalten.

Beispiele Betrachte die folgenden Integrationsformeln

1) Rechteck-Regel (links), d.h. mit Auswertungsstelle t_n:

$$\begin{aligned} x(t_{n+1}) &= x(t_n) + \int_{t_n}^{t_{n+1}} f(t, x(t))\, dt \\ &\simeq x(t_n) + (t_{n+1} - t_n)\ f(t_n, x(t_n)) \end{aligned}$$

$\Rightarrow$ das explizite Euler-Verfahren

$$\boxed{x_{n+1} = x_n + h_n\ f(t_n, x_n)}$$

2) Rechteck-Regel (mitte), d.h. mit Auswertungsstelle $t_n + \frac{1}{2}\ h_n$:

$$\begin{aligned} x(t_{n+1}) &= x(t_n) + \int_{t_n}^{t_{n+1}} f(t, x(t))\, dt \\ &\approx x(t_n) + (t_{n+1} - t_n)\ f\left(t_n + \frac{1}{2}\ h_n, x\left(t_n + \frac{1}{2}\ h_n\right)\right) \end{aligned}$$

Dann approximieren wir $x\left(t_n + \frac{1}{2}\ h_n\right)$ durch das explizite Euler-Verfahren:

$$x\left(t_n + \frac{1}{2}\ h_n\right) \approx x_n + \frac{1}{2}\ h_n\ f(t_n, x_n)$$

Wir erhalten einen Ausdruck, der nur $x(t_n)$ und $x(t_{n+1})$ enthält:

$$x(t_{n+1}) \approx x_n + h_n \; f\left(t_n + \frac{1}{2}\; h_n, x_n + \frac{1}{2}\; h_n \; f(t_n, x_n)\right)$$

$\Rightarrow$ das verbesserte Euler-Verfahren

$$\boxed{x_{n+1} = x_n + h_n \; f\left(t_n + \tfrac{1}{2}\; h_n, x_n + \tfrac{1}{2}\; h_n \; f(t_n, x_n)\right)}$$

3) Trapez-Regel

$$x(t_{n+1}) \approx x(t_n) + \frac{1}{2}\; (t_{n+1} - t_n)\,\{f(t_n, x(t_n)) + f(t_{n+1}, x(t_{n+1}))\}$$

Jetzt gibt es zwei Möglichkeiten

$\Rightarrow$ das Trapez-Verfahren (implizit!)

$$\boxed{x_{n+1} = x_n + \tfrac{1}{2}\; h_n\,\{f(t_n, x_n) + f(t_{n+1}, x_{n+1})\}}$$

oder wir approximieren das $x(t_{n+1})$ an der rechten Seite durch das explizite Euler-Verfahren

$\Rightarrow$ das Heun-Verfahren (explizit!)

$$\boxed{x_{n+1} = x_n + \tfrac{1}{2}\; h_n\{f(t_n, x_n) + f(t_{n+1}, x_n + h_n f(t_n, x_n))\}}$$

Für größeres s (d.h., die Anzahl der Auswertungsstellen) werden diese summierten Integrationsformel bald sehr kompliziert. Deshalb ist es günstig, die Zwischenauswertungen getrennt zu listen. Wir sagen, dass das Verfahren s Stufen hat, wo $s \geq 1$ die Anzahl von Auswertungsstellen ist.

(1) das explizite Euler-Verfahren

$$s = 1 \qquad \begin{aligned} k_1 &= f(t_n, x_n) \quad \text{nur eine einzige Auswertungstelle !} \\ x_{n+1} &= x_n + h_n k_1 \end{aligned}$$

(2) das verbesserte Euler-Verfahren

$$s = 2 \qquad \begin{cases} k_1 &= f(t_n, x_n) \\ k_2 &= f(t_n + \frac{1}{2}\; h_n,\; x_n + \frac{1}{2}\; h_n k_1) \end{cases}$$

$$x_{n+1} = x_n + h_n k_2$$

(3) das Heun-Verfahren

$$s = 2 \qquad \begin{cases} k_1 &= f(t_n, x_n) \\ k_2 &= f(t_n + h_n, x_n + h_n k_1) \end{cases}$$

$$x_{n+1} = x_n + \frac{1}{2}\, h_n k_1 + \frac{1}{2}\, h_n k_2$$

Wir können implizite Verfahren nach dieser Weise auch umschreiben.

(4) das implizite Euler-Verfahren $\quad x_{n+1} = x_n + h_n\ f(t_{n+1}, x_{n+1})$

Wir brauchen eine ($s = 1$) Auswertungsstelle $t_{n+1} = t_n + h_n$

$$k_1 = f(t_n + h_n, x_n + h_n k_1) \qquad \text{eine implizite Gleichung!}$$

$$x_{n+1} = x_n + h_n k_1$$

(5) das Trapez-Verfahren

$$s = 2 \qquad \begin{cases} k_1 &= f(t_n, x_n) \\ k_2 &= f(t_n + h_n, x_n + \frac{1}{2}\, h_n k_1 + \frac{1}{2}\, h_n k_2) \end{cases}$$

$$x_{n+1} = x_n + \frac{1}{2}\, h_n k_1 + \frac{1}{2}\, h_n k_2$$

3.1 Allgemeine Form eines Runge-Kutta-Verfahrens

Ein allgemeines Runge-Kutta-Verfahren mit s Stufen hat die Form

$$\boxed{\begin{array}{lcl} k_i & = & f\left(t_n + c_i h_n, x_n + h_n\ \sum_{j=1}^{s}\ a_{i,j} k_j\right), \quad i = 1, \ldots, s \\ x_{n+1} & = & x_n + h_n\ \sum_{i=1}^{s}\ b_i k_i \end{array}}$$

wobei $0 \le c_1 < c_2 < \ldots < c_s \le 1$.

Ein solches Verfahren ist eindeutig bestimmt durch die Parameter

$$c = \begin{pmatrix} c_1 \\ \vdots \\ c_s \end{pmatrix}, b = \begin{pmatrix} b_1 \\ \vdots \\ b_s \end{pmatrix}^T, \ A = [a_{i,j}], \quad s \times s - \text{Matrix}$$

d.h. durch das Butcher-Tableau (oder Butcher-Schemata)

$$\begin{array}{c|c} c & A \\ \hline & b \end{array}$$

Beispiele

(1) das explizite Euler-Verfahren

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

(2) das implizite Euler-Verfahren

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

(3) das verbesserte Euler-Verfahren

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

(4) das Heun-Verfahren

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

(5) das Trapez-Verfahren

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

Bemerkungen Ein RK-Verfahren mit Butcher-Tableau

$$\begin{array}{c|c} c & A \\ \hline & b \end{array}$$

ist

(1) explizit $\Leftrightarrow$ $a_{i,j} = 0$ für alle $j \geq i$, $i = 1, \ldots, s$

(2) konsistent $\Leftrightarrow \sum_{i=1}^{s} b_i = 1$

Beweis: Schreibe das Verfahren $x_{n+1} = x_n + h_n \Phi\ (h_n, t_n, x_n)$, wobei

$$\Phi\ (h_n, t_n, x_n) = \sum_{i=1}^{s} b_i k_i \to \sum_{i=1}^{s} b_i f(t_n, x_n)$$

und merke, dass jedes $k_i \to f(t_n, x_n)$ für $h_n \to 0$, d.h.

$$\lim_{h_n \to 0} \Phi(h_n, t_n, x_n) = \left(\sum_{i=1}^{s} b_i \right)\ f(t_n, x_n) \equiv f(t_n, x_n) \Leftrightarrow \sum_{i=1}^{s} b_i = 1$$

3.2 Autonome Differentialgleichungen

Das Vektorfeld f einer autonomen Differentialgleichung hängt nicht von t ab, d.h., die DGL ist der Form

$$\frac{dx}{dt} = f(x)$$

Die Lösungen einer autonomen DGL sind invariant gegen Zeittranslation: ist $x(t)$ eine Lösung, dann ist $z(t) := x(t+c)$ auch eine Lösung für jede Konstante $c \in \mathbb{R}$. Deshalb können wir uns zu der Anfangszeit $t_0 = 0$ beschränken:

$$AWA \qquad \begin{cases} \dfrac{dx}{dt} = f(x), \\ x(0) = x_0. \end{cases}$$

In den entsprechenden numerischen Verfahren fallen die Zeitpunkte t_n weg: z.B., das explizite Euler-Verfahren lautet

$$x_{n+1} = x_n + h_n f(x_n)$$

für eine autonome Differentialgleichung.

3.2.1 Autonomisierung

Wir können jede nichtautonome DGL

$$\frac{dx}{dt} = f(t, x), \quad x \in \mathbb{R}^d, \quad t \in [t_0, T]$$

in eine autonome DGL

$$\frac{dX}{dt} = F(X), \quad X \in \mathbb{R}^{d+1}$$

umschreiben, d.h. der Zustandsraum hat jetzt eine zusätzliche Dimension.

Definiere $\tau = t - t_0$, $X = \begin{pmatrix} x \\ \tau \end{pmatrix}$, $F(X) = \begin{pmatrix} f(\tau + t_0, x) \\ 1 \end{pmatrix}$.

Dann gilt

$$\begin{aligned} \frac{dX}{d\tau} = \frac{d}{d\tau}\begin{pmatrix} x \\ \tau \end{pmatrix} = \begin{pmatrix} \frac{dx}{d\tau} \\ 1 \end{pmatrix} &= \begin{pmatrix} \frac{dx}{dt}\frac{dt}{d\tau} \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} f(t,x)\cdot 1 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} f(\tau + t_0, x) \\ 1 \end{pmatrix} = F(X) \end{aligned}$$

wie erwünscht.

Frage: *Ist ein numerisches Verfahren für eine nichtautonome DGL identisch dem entsprechenden Verfahren für die autonomisierte DGL?*

d.h. ist ein Verfahren invariant gegen Autonomisierung?

JA für das explizite Euler-Verfahren! Von den DGLen und Verfahren

$$x_{n+1} = x_n + h_n f(t_n, x_n), \qquad X_{n+1} = X_n + h_n F(X_n)$$

erhalten wir

$$\begin{aligned} \text{RHS} = X_n + h_n F(X_n) &= \begin{pmatrix} x_n \\ \tau_n \end{pmatrix} + h_n \begin{pmatrix} f(\tau_n + t_0, x_n) \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} x_n + h_n f(\tau_n + t_0, x_n) \\ \tau_n + h_n \end{pmatrix} \\ &= \begin{pmatrix} x_n + h_n f(t_n, x_n) \\ \tau_{n+1} \end{pmatrix} \\ &= \begin{pmatrix} x_{n+1} \\ \tau_{n+1} \end{pmatrix} = X_{n+1} = \text{LHS} \end{aligned}$$

weil $t_n = \tau_n + t_0$ und $h_n = t_{n+1} - t_n = \tau_{n+1} - \tau_n$ sind.

ABER nicht alle numerische Verfahren sind invariant gegen Autonomisierung, z.B. betrachte das Einschrittverfahren

$$x_{n+1} = x_n + h_n \Phi(h_n, t_n, x_n)$$

mit

$$\Phi(h, t, x) = f(t + h, x)$$

Dieses Verfahren ist konsistent:

$$\lim_{h \downarrow 0} \Phi(h, t, x) = \lim_{h \downarrow 0} f(t + h, x) \equiv f(t, x)$$

(falls f stetig ist).

Die nichtautonome und autonome Version der Verfahren lauten:

(1) $x_{n+1} = x_n + h_n f(t_{n+1}, x_n)$, ein "schiefes" Euler-Verfahren

(2) $X_{n+1} = X_n + h_n F(X_n)$, kein t-Komponent hier.

Dann haben wir

$$\begin{aligned}
\text{RHS} = X_n + h_n F(X_n) &= \begin{pmatrix} x_n \\ \tau_n \end{pmatrix} + h_n \begin{pmatrix} f(\tau_n + t_0, x_n) \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} x_n + h_n f(\tau_n + t_0, x_n) \\ \tau_n + h_n \end{pmatrix} \\
&= \begin{pmatrix} x_n + h_n f(t_n, x_n) \\ \tau_{n+1} \end{pmatrix} \\
&\underbrace{\neq}_{\text{im Allgem.}} \begin{pmatrix} x_n + h_n f(t_{n+1}, x_n) \\ \tau_{n+1} \end{pmatrix} = \text{LHS}
\end{aligned}$$

falls $\dfrac{\partial f}{\partial t}(t, x) \not\equiv 0$ (d.h., falls die DGL nichtautonom ist).

Bemerkung Viele Lehrbücher beschränken sich nur zum autonomen Fall oder geben Beweise nur für diesen Fall. Schreibe

$$X_n = \begin{pmatrix} x_n \\ \tau_n \end{pmatrix}, \qquad K_i = \begin{pmatrix} k_i \\ \theta_i \end{pmatrix}.$$

3.2.2 Invarianz gegen Autonomisierung

Ein RK-Verfahren mit Butcher-Tableau

$$\begin{array}{c|c} c & A \\ \hline & b \end{array}$$

ist invariant gegen Autonomisierung (für ein konsistentes Verfahren)

$$\Leftrightarrow \quad c_i = \sum_{j=1}^{s} a_{i,j}, \quad i = 1, \ldots, s$$

Schreibe

$$X_n = \begin{pmatrix} x_n \\ \tau_n \end{pmatrix}, \qquad K_i = \begin{pmatrix} k_i \\ \theta_i \end{pmatrix}.$$

Das obige RK-Verfahren für die autonomisierte DGL $\frac{dX}{d\tau} = F(X)$ lautet

$$\begin{cases} K_i &= F\left(X_n + h_n \ \sum_{j=1}^{s} \ a_{i,j} K_j\right), \qquad i = 1, \ldots, s \\ X_{n+1} &= X_n + h_n \ \sum_{i=1}^{s} \ b_i K_i \end{cases}$$

Wir haben komponentenweise

$$\begin{aligned} K_i = \begin{pmatrix} k_i \\ \theta_n \end{pmatrix} &= F\left(\begin{pmatrix} x_n \\ \tau_n \end{pmatrix} + h_n \sum_{j=1}^{s} a_{i,j} \begin{pmatrix} k_j \\ \theta_j \end{pmatrix}\right) \\ &= F\begin{pmatrix} x_n + h_n \sum_{j=1}^{s} \ a_{i,j} k_j \\ \tau_n + h_n \ \sum_{j=1}^{s} \ a_{i,j} \theta_j \end{pmatrix} \\ &= \begin{pmatrix} f\left(t_0 + \tau_n + h_n \sum_{j=1}^{s} a_{i,j}\theta_j, \ x_n + h_n \sum_{j=1}^{s} a_{i,j} k_j\right) \\ 1 \end{pmatrix} \end{aligned}$$

$$\Rightarrow \begin{cases} k_i &= f\left(t_0 + \tau_n + h_n \sum_{j=1}^{s} a_{i,j}\theta_j, \ x_n + h_n \sum_{j=1}^{s} a_{i,j} k_j\right) \\ \theta_i &= 1 \end{cases}$$

$$\Rightarrow k_i \quad = \quad f\left(t_0 + \tau_n + h_n \sum_{j=1}^{s} a_{i,j}, \ x_n + h_n \sum_{j=1}^{s} a_{i,j} k_j\right)$$

$$= f\left(t_0 + \tau_n + h_n c_i,\ x_n + h_n \sum_{j=1}^{s} a_{i,j} k_j\right)$$

$$= f\left(t_n + c_i h_n,\ x_n + h_n \sum_{j=1}^{s} a_{i,j} k_j\right),$$

weil $c_i = \sum_{j=1}^{s} a_{i,j}$, $i = 1 \ldots, s$.

Zusätzlich gilt

$$X_{n+1} = X_n + h_n \sum_{i=1}^{s} b_i K_i$$

oder komponentenweise

$$\begin{cases} x_{n+1} &= x_n + h_n \sum_{i=1}^{s} b_i k_i \\ \tau_{n+1} &= \tau_n + h_n \underbrace{\sum_{i=1}^{s} b_i}_{=1} = \tau_n + h_n \end{cases},$$

weil das Verfahren konsistent ist.

Das obige RK-Verfahren für die autonomisierte DGL lautet

$$\begin{cases} k_i &= f(t_n + c_i h_n, x_n + h_n \sum_{i=1}^{s} a_{ij} k_j, \qquad i = 1, \ldots, s \\ x_{n+1} &= x_n + h_n \sum_{i=1}^{s} b_i k_i \end{cases},$$

d.h., das entsprechende RK-Verfahren für die ursprüngliche nichtautonome DG.

Beispiele

1) die Euler (explizit, implizit,verbessert)-, Heun- und Trapez-Verfahren sind alle invariant gegen Autonomisierung.

2) Wir haben schon bewiesen, dass das „schiefe“ Euler-Verfahren

$$x_{n+1} = x_n + h_n f(t_{n+1}, x_n)$$

nicht invariant gegen Autonomisierung ist.

Hier ist das Butcher-Tableau für das schiefe Euler-Verfahren:

1	0
	1

(mit $c_1 \neq a_{1,1}$!)

$$\begin{cases} k_1 &= f(t_n + h_n, x_n) \\ x_{n+1} &= x_n + h_n k_i \end{cases}$$

Kapitel 4

Explizite Runge-Kutta-Verfahren

Ein s-stufiges Runge-Kutta-Verfahren mit Butcher-Tableau

$$\begin{array}{c|c} c & A \\ \hline & b \end{array}$$

ist ein explizites Verfahren genau dann, wenn

$$a_{i,j} = 0, \qquad \forall j \geq i, \quad i, j = 1, \ldots, s$$

d.h.

$$A = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 \\ * & 0 & 0 & \ldots & 0 \\ * & * & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ * & * & * & \ldots & 0 \end{bmatrix}$$

d.h. k_j hängt nur von $k_1, \ldots, k_{j-1}$ ab $(j \geq 2)$ und $k_1 = f(t_n + c_1 h_n, x_n)$.

Wir haben auch $c_1 = \sum_{j=1}^{s} a_{1,j} = 0$ für ein autonomisierunginvariantes Verfahren.

4.1 Ordnung und Anzahl von Stufen

Frage: *Wie ist der Zusammenhang zwischen der Anzahl der Stufen s und der Ordnung p eines expliziten RK-Verfahrens?*

Bemerkung Sei $t_n \in [t_0, T]$ und y_n Lösung des numerischen Verfahrens und $y(t_n)$ die exackte Lösung. Wenn für alle $t_n \in [t_0, T]$ gilt

$$\lim_{h \to 0} < \frac{1}{h^p} |y_n - y(t_n)| < \infty$$

dann sagen wir das numerische Verfahren hat Ordnung p.

Die Ordnung eines numerischen Verfahrens ist bezüglich einer Klasse von Differentialgleichungen, z.B. mit p-mal stetig differenzierbaren Vektorfeldfunktionen.

SATZ *Ein explizites RK-Verfahren mit s Stufen hat Ordnung $p \leq s$*

Beweis Betrachte die AWA

$$\frac{dx}{dt} = x, \quad x(0) = x_0,$$

d.h. mit der Vektorfeldfunktion $f(t, x) \equiv x$, die beliebig-mal stetig differenzierbar ist.

Die Lösung lautet $x(t, x_0) = x_0 e^t$.

Insbesondere gilt

$$x(h, x_0) = x_0 \left(1 + h + \ldots + \frac{h^p}{p!}\right) + O(h^{p+1}).$$

Betrachte jetzt ein explizites RK-Verfahren mit Butcher-Tableau

$$\begin{array}{c|c} c & A \\ \hline & b \end{array}$$

Für die Vektorfeldfunktion $f(t, x) \equiv x$ haben wir (nimm $h_n \equiv h$ hier)

$$\begin{aligned}
k_1 &= x_n + h0 = x_n \\
k_2 &= x_n + h a_{2,1} k_1 = x_n (1 + a_{2,1} h) \\
k_3 &= x_n + h a_{3,1} k_1 + h a_{3,2} k_2 \\
&= x_n \left[1 + h a_{3,1} + h a_{3,2} \left(1 + a_{2,1} h\right)\right] \\
&= x_n \left[1 + h \left(a_{3,1} + a_{3,2}\right) + h^2 a_{3,2} a_{2,1}\right]
\end{aligned}$$

und so fort.

$$\Rightarrow \quad k_i = x_n \ \varphi_i(h) \ \text{ mit } \ \varphi_i \in \mathcal{P}_{i-1}$$

d.h. φ_i ist ein Polynom höchstens Grades $i-1$, wo $i = 1, \ldots, s$.

Dann gilt

$$\begin{aligned} x_{n+1} &= x_n + h \sum_{i=1}^{s} b_i k_i \\ &= x_n + h\, x_n \underbrace{\sum_{i=1}^{s} b_i \ \varphi_i(h)}_{\substack{\text{Polynom höchstens} \\ \text{Grades } s-1}} \\ &= x_n \Phi_s^*(h) \Leftarrow \ \text{Polynom höchstens Grades } \ s \end{aligned}$$

Der lokale Diskretisierungsfehler (nimm $n = 0$) lautet

$$\begin{aligned} L_0 = |x(h, x_0) - x_1| &= |x_0 e^h - x_0 \Phi_s^* \ (h)| \\ &= |x_0| \ |e^h - \Phi_s^* \ (h)| \\ &= |x_0| \ \left| \left(1 + h \ldots + \frac{h^p}{p!}\right) - \Phi_s^*(h) + O(h^{p+1}) \right| \end{aligned}$$

Aber $1 + h + \ldots + \dfrac{h^p}{p^!} - \Phi_s^*(h) = O(h^j)$ mit $j = 1 \ + \ \min \{p, s\}$ und $L_0 = O(h^{p+1})$

$$\Rightarrow \quad p \leq s$$

Bemerkung

(i) $p = s = 1$ explizites/implizites Euler-Verfahren

(ii) $p = s = 2$ verbessertes Euler-Verfahren oder Hein-Verfahren.

Frage: *Müssen wir immer $p = s$ haben?* NEIN!

Betrachte das Verfahren

$$\boxed{x_{n+1} = x_n + \tfrac{1}{2}\, h_n\{f(t_n, x_n) + f(t_{n+1}, x_n)\}}$$

d.h. mit

$$\Phi(h,t,x) = \frac{1}{2}\,\{f(t,x) + f(t+h,x)\} \to f(t,x) \;\text{ für }\; h \to 0 \qquad \text{konsistent!}$$

Die Ordnung ist $p = 1$ (das Verfahren ist genau das explizite Euler-Verfahren, falls $f(t,x) \equiv f(x)$ gilt).

Die entsprechende RK-Formulierung hat $s = 2$ Stufen

$$\left\{ \begin{array}{lcl} k_1 &=& f(t_n, x_n) \\ k_2 &=& f(t_n + h_n, x_n) \end{array} \right.$$

$$x_{n+1} \;=\; x_n + \frac{1}{2}\, h_n k_1 + \frac{1}{2}\, h_n k_2$$

mit Butcher-Tableau

0	0	0
1	0	0
	1/2	1/2

Hier gilt $p = 1 < s = 2$.

Ein explizites RK-Verfahren mit s Stufen enthält

$$s + s + \frac{1}{2}\, s(s-1) = \frac{1}{2}\, s(s+3)$$

Parameter, d.h., s von c, s von b und $\frac{1}{2}\, s(s-1)$ von A.

Aber nicht alle Parameter sind frei, z.B. wir haben wegen

Konsistenz $\quad \sum_{i=1}^{s}\, b_i = 1$

Invarianz gegen Autonomisierung $\quad c_i = \sum_{j=1}^{s}\, a_{i,j}$ für $i = 1, \ldots, s$.

Diese Gleichungen bestimmen $1 + s$ Parameter. Deshalb bleiben

$$\frac{1}{2}\, s(s+3) - 1 - s = \frac{1}{2}\, (s-1)\, (s+2)$$

frei.

Beispiele

(1) $\underline{s=1} \quad \Rightarrow \quad \frac{1}{2}(s-1)(s+2) = 0$ keine freien Parameter!

$$\begin{cases} b_1 = 1 \\ c_1 = a_{1,1} = 0 \end{cases} \quad \Leftrightarrow \quad \text{Butcher-Tableau} \quad \begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

d.h. das explizite Euler-Verfahren ist die einzige Möglichkeit hier.

(2) $\underline{s=2} \quad \Rightarrow \quad \frac{1}{2}(s-1)(s+2) = 2$

$$\begin{cases} b_1 = \beta, & b_2 = 1 - b_1 = 1 - \beta \\ c_1 = a_{11} = 0, & c_2 = a_{21} = \alpha \end{cases},$$

wo $0 \leq \alpha \leq 1$,

$$\Leftrightarrow \quad \text{Butcher-Tableau} \quad \begin{array}{c|cc} 0 & 0 & 0 \\ \alpha & \alpha & 0 \\ \hline & \beta & 1-\beta \end{array}$$

$$\Leftrightarrow \quad \begin{cases} k_1 &= f(t_n, x_n) \\ k_2 &= f(t_n + \alpha h,\ x_n + h\alpha k_1) \end{cases}$$

$$x_{n+1} = x_n h\beta k_1 + h(1-\beta)k_2$$

oder

$$x_{n+1} = x_n + h\beta f(t_n, x_n) + h(1-\beta) f(t_n + \alpha h, x_n + \alpha h f(t_n, x_n))$$

z.B.

verbesserte Euler-Verfahren: $\alpha = 1/2, \quad \beta = 0$
das Heun-Verfahren $\alpha = 1, \quad \beta = 1/2$

Frage: *Für welche α, β haben wir Ordnung $p = 1$ oder $p = 2$?*

Die Parameter müssen zusätzlichen Gleichungen genügen, falls das RK-Verfahren eine gewünschte Ordnung p haben soll

Diese Gleichungen sind tatsächlich hinreichend und notwendig.

SATZ (Deuflhard/Bornemann, Satz 4.17, Seite 122)

Ein explizites Runge-Kutta-Verfahren mit Butcher-Tableau

$$\left(\begin{array}{c|c} c & A \\ \hline & b \end{array}\right),$$

das invariant gegen Autonomisierung ist, besitzt für alle $f \in C^p(\mathbb{R}^d, \mathbb{R}^d)$ *und Dimensionen* $d \in \mathbb{N}$

- *genau dann die Ordnung* $p = 1$, *wenn die Koeffizienten des Verfahrens der Bedingungsgleichung*

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

  genügen;

- *genau dann die Ordnung* $p = 2$, *wenn sie zusätzlich der Bedingungsgleichung*

  $$\sum_{i=1}^{s} b_i c_i = \frac{1}{2}$$

  genügen;

- *genau dann die Ordnung* $p = 3$, *wenn sie zusätzlich den zwei Bedingungsgleichungen*

  $$\sum_{i=1}^{s} b_i c_i^2 = \frac{1}{3}$$

  $$\sum_{i=1}^{s} \sum_{j=1}^{s} b_i a_{i,j} c_j = \frac{1}{6}$$

  genügen;

(nächste Seite)

- *genau dann die Ordnung $p = 4$, wenn sie zusätzlich den vier Bedingungsgleichungen*

$$\sum_{i=1}^{s} b_i \, c_i^3 = \frac{1}{4}$$

$$\sum_{i=1}^{s} \sum_{j=1}^{s} b_i c_i a_{i,j} c_j = \frac{1}{8}$$

$$\sum_{i=1}^{s} \sum_{j=1}^{s} b_i \, a_{i,j} \, c_j^2 = \frac{1}{12}$$

$$\sum_{i=1}^{s} \sum_{j=1}^{s} \sum_{k=1}^{s} b_i \, a_{i,j} \, a_{j,k} \, c_k = \frac{1}{24}$$

genügen.

Bemerkung (Siehe z.B. Stuart/Humphries, Seite 236)

Ordnung	1	2	3	4	5	6	7	8	9	10
Anzahl der Bedingungen	1	2	4	8	17	37	85	200	489	1205

Ein guter Grund uns zu $p \leq 4$ zu beschränken!

Beweis (zu Deuflhard/Bornemann) Wir haben

$$c_i = \sum_{j=1}^{s} a_{i,j}, \quad i = 1, \ldots, s$$

wegen der Invarianz gegen Autonomisierung. Deshalb genügt es, nur autonome DGLen

$$\frac{dx}{dt} = f(x) \qquad (*)$$

zu betrachten.

Wir beschränken uns zum 1-dimensionalen Fall – dann gilt der Beweis nur für $p = 1, 2$ und 3. Für höhere Dimensionen und $p \geq 4$ entgegnen wir der Nichtkommutativität verschiedener höherer Ableitungen. In solchen Fällen ist der Beweis nur „hinreichend“ – siehe die Bemerkungen in Deuflhard/Bornemann (Bemerkung 4.18, Seite 122) über die „notwendige“ Richtung.

Die Lösung der DGL $(*)$ genügt der Taylor-Entwicklung

$$\begin{aligned} x(h) &= x(0) + hx'(0) + \frac{h^2}{2!}\, x'(0) + \frac{h^3}{3!}\, x'''(0) + O(h^4) \\ &= x + hf(x) + \frac{h^2}{2!}\, Df(x) + \frac{h^3}{3!}\, D^2 f(x) + O(h^4) \end{aligned}$$

mit $x(0) = x$ und der totalen Ableitung.

$$Du(x) = f(x)u'(x), \qquad ' = \frac{d}{dx}$$

in diesem autonomen skalaren Fall

$$\begin{aligned} \Rightarrow \quad Df(x) &= f(x)f'(x) \\ D^2 f(x) &= f(x)\{f(x)f'(x)\}' = f(x)^2 f''(x) + f(x)f'(x)^2 \end{aligned}$$

Die Taylor-Entwicklung lautet

$$\begin{aligned} x(h) &= x + hf(x) + \frac{1}{2}\, h^2 f(x)f'(x) \\ &\quad + \frac{1}{6}\, h^3 \{f(x)^2 f''(x) + f(x)f'(x)^2\} + O(h^4) \end{aligned}$$

Betrachte jetzt das explizite RK-Verfahren für die Lösung der DGL $(*)$

$$\left\{ \begin{array}{lcl} k_i & = & f\left(x_n + h \sum\limits_{j=1}^{i-1} a_{i,j}\, k_j\right), i = 1, \ldots, s, \qquad \sum_1^0 \equiv 0 \\ x_{n+1} & = & x_n + h\, \sum_{i=1}^{s} b_i\, k_i \end{array} \right.$$

Sei $p = 1$

$$\begin{cases} k_1 &= f(x_n) \\ k_i &= f(x_n) + O(h), \quad i = 2, \ldots, s \end{cases}$$

$$\begin{aligned} \Rightarrow \quad x_{n+1} &= x_n + h \sum_{i=1}^{s} b_i \ [f(x_n) + O(h)] \\ &= x_n + h \ f(x_n) \sum_{i=1}^{s} b_i + O(h^2) \end{aligned}$$

Der lokale Diskretisierungsfehler (für $n = 0$) lautet

$$\begin{aligned} L_0 &= |x(h) - x_1| \\ &= \left| \left\{ x + hf(x) + O(h^2) \right\} - \left\{ x + hf(x) \sum_{i=1}^{s} b_i + O(h^2) \right\} \right| \\ &= h|f(x)| \left| 1 - \sum_{i=1}^{s} b_i \right| + O(h^2) \\ &= O(h^2), \end{aligned}$$

für beliebiges $f \in C^1$,

$$\Leftrightarrow \quad 1 = \sum_{i=1}^{s} b_i$$

d.h. Ordnung $p = 1 \Leftrightarrow$ Konsistenz (mindestens)

Sei $p = 2$

$$\begin{aligned} k_1 &= f(x_n) \\ k_2 &= f(x_n + ha_{2,1} \ k_1) \\ &= f(x_n) + ha_{2,1} \ k_1 \ f'(x_n) + O(h^2) \\ &= f(x_n) + ha_{2,1} \ f(x_n) \ f'(x_n) + O(h^2) \end{aligned}$$

und im Allgemeinen für $i \geq 2$

$$k_i = f\left(x_n + h \sum_{j=1}^{i-1} a_{i,j} \ k_j \right)$$

$$
\begin{aligned}
&= f(x_n) + h \left(\sum_{j=1}^{i-1} a_{i,j}\ k_j \right) f'(x_n) + O(h^2) \\
&= f(x_n) + h \left(\sum_{j=1}^{i-1} a_{i,j} \right) (f(x_n) + O(h))\, f'(x_n) + O(h^2) \\
&\qquad (\text{weil } \ k_j = f(x_n) + O(h)) \\
&= f(x_n) + h\ c_i\ f(x_n) f'(x_n) + O(h^2)
\end{aligned}
$$

weil $c_i = \sum_{j=1}^{i-1}\ a_{i,j} = \sum_{j=1}^{s}\ a_{i,j}$

$$
\begin{aligned}
\Rightarrow \qquad x_{n+1} &= x_n + h \sum_{i=1}^{s}\ b_i\ k_i \\
&= x_n + h \sum_{i=1}^{s}\ b_i \left[f(x_n) + h\ c_i\ f(x_n) f'(x_n) + O(h^2) \right] \\
&= x_n + h \left(\sum_{i=1}^{s}\ b_i \right) f(x_n + h^2 \left(\sum_{i=1}^{s}\ b_i\ c_i \right) f(x_n) f'(x_n) + O(h^3)
\end{aligned}
$$

Hier lautet der lokale Diskretisierungsfehler ($n = 0,\ x_0 = x$)

$$
\begin{aligned}
L_0 &= |x(h) - x_1| \\
&= \Big| \Big\{ x + hf(x) + \frac{1}{2}\ h^2 f(x) f'(x) + O(h^3) \Big\} \\
&\qquad - \Big\{ x + \Big(\sum_{i=1}^{s}\ b_i \Big) f(x) + h^2\ \Big(\sum_{i=1}^{s}\ b_i\ c_i) \Big) f(x) f'(x) + O(h^3) \Big\} \Big| \\
&\leq h|f(x)|\ \cdot \Big| 1 - \sum_{i=1}^{s}\ b_i \Big| \ + \ h^2 |f(x) f'(x)|\ \cdot \Big| \frac{1}{2} \ - \ \sum_{i=1}^{s}\ b_i\ c_i \Big| + O(h^3) \\
&= O(h^3)
\end{aligned}
$$

für ein beliebiges $f \in C^2$.

$$
\Leftrightarrow \left\{ \begin{array}{lcl} 1 & = & \sum_{i=1}^{s}\ b_i \\ \frac{1}{2} & = & \sum_{i=1}^{s}\ b_i\ c_i \end{array} \right.
$$

$\underline{p = 3}$ Wir betrachten jetzt nur die Terme der nächstfolgenden Ordnung

$$
\begin{aligned}
k_1 &= f(x_n) \\
k_2 &= f\left(x_n + h a_{2,1} k_i\right) \\
&= f\left(x_n + h a_{2,1} f(x_n)\right) \\
&= f(x_n) + h a_{2,1} f(x_n) f'(x_n) + \frac{1}{2}\, h^2\, a_{2,1}^2 f(x_n)^2\, f''(x_n) + O(h^3)
\end{aligned}
$$

<u>im Allgemeinen für $i \geq 2$</u>

$$
\begin{aligned}
k_i &= f\left(x_n + h \sum_{j=1}^{i-1} a_{i,j} k_j\right) \\
&= f(x_n) + h \left(\sum_{j=1}^{i-1} a_{i,j} k_j\right) f'(x_n) + \frac{1}{2}\, h^2 \left(\sum_{j=1}^{i-1} a_{i,j} k_j\right)^2 f''(x_n) + O(h^3) \\
&= f(x_n) + h \sum_{j=1}^{i-1} a_{i,j} \left\{f(x_n) + h c_j f(x_n) f'(x_n) + O(h^2)\right\} f'(x_n) \\
&\qquad + \frac{1}{2}\, h^2 \left(\sum_{j=j}^{i-1} a_{i,j} \{f(x_n) + O(h)\}\right)^2 f''(x_n) + O(h^3) \\
&= f(x_n) + h \left(\sum_{j=1}^{i-1} a_{i,j}\right) f(x_n) f'(x_n) \\
&\qquad + h^2 \left(\sum_{j=1}^{i-1} a_{i,j} c_j\right) f(x_n) f'(x_n)^2 + O(h^3) \\
&\qquad + \frac{1}{2}\, h^2 \left(\sum_{j=1}^{i-1} a_{i,j}\right)^2 f(x_n)^2 f''(x_n) + O(h^3) \\
&= f(x_n) + h\, c_i f(x_n) f'(x_n) + h^2 \left(\sum_{j=1}^{i-1} a_{i,j} c_j\right) f(x_n) f'(x_n)^2 \\
&\qquad + \frac{1}{2}\, h^2\, c_i^2\, f(x_n)^2 f''(x_n) + O(h^3)
\end{aligned}
$$

weil $c_i = \sum_{j=1}^{i-1} a_{i,j} = \sum_{j=1}^{s} a_{i,j}$ explizit!

$$\begin{aligned}
\Rightarrow \quad x_{n+1} &= x_n + h \sum_{i=1}^{s} b_i k_i \\
&= x_n + h \left(\sum_{i=1}^{s} b_i \right) f(x_n) \\
&\quad + h^2 \left(\sum_{i=1}^{s} b_i c_i \right) f(x_n) f'(x_n)^2 \\
&\quad + h^3 \left(\sum_{i=1}^{s} \sum_{j=1}^{i-1} b_i a_{i,j} c_j \right) f(x_n) f'(x_n)^2 \\
&\quad + \frac{1}{2} h^3 \left(\sum_{i=1}^{s} b_i c_i^2 \right) f(x_n)^2 f''(x_n) + O(h^4)
\end{aligned}$$

Wie oben, für $L \sim O(h^4)$ brauchen wir

- $\sum_{i=1}^{s} b_i = 1$
- $\sum_{i=1}^{s} b_i c_i = \frac{1}{2}$

sowie (Vergleich der $O(h^3)$-Terme)

- $\sum_{i=1}^{s} \sum_{j=1}^{i-1} b_i a_{i,j} c_j = \frac{1}{6}$
- $\frac{1}{2} \sum_{i=1}^{s} b_i c_i^2 = \frac{1}{6}$

Die zusätzlichen Terme sind die erwünschten Terme, weil das Verfahren explizit ist

$$\Rightarrow \qquad \sum_{j=1}^{i-1} a_{i,j} b_i c_j = \sum_{j=1}^{s} a_{i,j} b_i c_j$$

Im Prinzip geht es wie oben für $p = 4, \ldots$.

4.2 Beispiele expliziter Runge-Kutta-Verfahren

Die Koeffizienten des Butcher-Tableaus $\begin{array}{c|c} c & A \\ \hline & b \end{array}$ eines s-stufigen RK-Verfahrens müssen verschiedenen Bedingungsgleichungen genügen.

- explizit $\quad a_{i,j} = 0 \quad i \le j \le s, \quad i = 1, \ldots, s$
- invariant gegen Autonomisierung $\quad c_i = \sum_{j=1}^{s} a_{i,j} \quad (I_i), \quad i = 1, \ldots, s$
- Ordnung p braucht die folgenden Bedingungen

$$\begin{array}{lll} p = 1 : & (O_1) & \\ p = 2 : & (O_1) & \text{und} \quad (O_2) \\ p = 3 : & (O_1) & \text{bis} \quad (O_4) \\ p = 4 : & (O_1) & \text{bis} \quad (O_8), \end{array}$$

wobei

$$\begin{aligned} &\textstyle\sum_{i=1}^{s} b_i = 1 && (O_1) \\ &\textstyle\sum_{i=1}^{s} b_i c_i = \frac{1}{2} && (O_2) \\ &\textstyle\sum_{i=1}^{s} b_i c_i^2 = \frac{1}{3} && (O_3) \\ &\textstyle\sum_{i=1}^{s} \sum_{j=1}^{s} b_i a_{i,j} c_j = \frac{1}{6} && (O_4) \\ &\textstyle\sum_{i=1}^{s} b_i c_i^3 = \frac{1}{4} && (O_5) \\ &\textstyle\sum_{i=1}^{s} \sum_{j=1}^{s} b_i c_i a_{i,j} c_j = \frac{1}{8} && (O_6) \\ &\textstyle\sum_{i=1}^{s} \sum_{j=1}^{s} b_i a_{i,j} c_j^2 = \frac{1}{12} && (O_7) \\ &\textstyle\sum_{i=1}^{s} \sum_{j=1}^{s} \sum_{k=1}^{s} b_i a_{i,j} a_{j,k} c_k = \frac{1}{24} && (O_8) \end{aligned}$$

(1) das explizite Euler-Verfahren

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array} \qquad s = 1 \qquad c_1 = a_{11} = 0 \qquad (I_1)$$

$$b_1 = 1 \qquad (O_1)$$

$$\Rightarrow \qquad \text{Ordnung} \qquad p \ge 1$$

Betrachte jetzt $(O_2) : \sum_{i=1}^{s} b_i c_i = 1/2$ hier gilt

$$\sum_{i=1}^{s} b_i c_i \;=\; b_1 c_1 \;=\; 1.0 \;=\; 0 \neq 1/2,$$

d.h. die Bedingung (O_2) gilt nicht

$$\Rightarrow \quad p \neq 2$$

$\Rightarrow$ die genaue Ordnung $p = 1$.

(2) <u>das Heun-Verfahren</u>

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \end{array} \qquad s = 2 \qquad \begin{array}{ll} (I_1) & 0 = 0 + 0 \\ (I_2) & 1 = 1 + 0 \\ (O_1) & 1/2 + 1/2 = 1 \end{array}$$

$$\Rightarrow \quad \text{Ordnung} \quad p \geq 1$$

Die Bedingungen (O_2) lautet hier $\sum_{i=1}^{2} b_i c_i = \frac{1}{2}\, 0 + \frac{1}{2}\, 1 = \frac{1}{2}$

$$\Rightarrow \quad \text{Ordnung} \quad p \geq 2$$

Betrachte jetzt die Bedingungen (O_3) : $\sum_{i=1}^{s} b_i c_i^2 = \frac{1}{3}$, die hier lautet

$$\sum_{i=1}^{s=2} b_i c_i^2 = \frac{1}{2}\, 0^2 + \frac{1}{2}\, 1^2 = \frac{1}{2} \;\neq\; \frac{1}{3}$$

$$\Rightarrow \quad \text{Ordnung} \quad p \neq 3$$

<u>d.h.</u> die genaue Ordnung $p = 2$

<u>oder</u> betrachte die Bedingungen (O_4) : $\sum_{i=1}^{s} \sum_{j=1}^{s} b_i a_{i,j} c_j = \frac{1}{6}$, die hier lautet

$$\sum_{i=1}^{2} b_i \left(\sum_{j=1}^{2} a_{ij} c_j \right) = \frac{1}{2}\, (0.0 + 0.1) + \frac{1}{2}\, (1.0 + 0.1) = 0 \neq \frac{1}{6}$$

In diesem Fall gilt (O_4) auch nicht – aber im Allgemeinen muß nur eine der Bedingungen (O_3) und (O_4) nicht gelten, um $p \neq 3$ zu verursachen.

4.3 Herleitung expliziter RK-Verfahren

Die folgenden Fakten sind bekannt.

(1) Die Beschränkung $p \leq s$ ist notwendig für die <u>Lösbarkeit</u> des Gleichungssystems;

(2) das Gleichungssystem besitzt viel <u>Redundanz</u>: es ist stark unterbestimmt für $p \geq 6$;

(3) das Gleichungssystem ist lösbar mit $p = s$ nur für $p = 1, 2, 3, 4$.

Sei s_p die minimale Stufenzahl eines expliziten RK-Verfahrens von Ordnung p. Butcher hat die folgenden Ergebnisse bewiesen:

p	1	2	3	4	5	6	7	8	≥ 9
s_p	1	2	3	4	6	7	9	11	$\geq p+3$

FRAGE *Wie finden wir diese "optimalen" RK-Verfahren von Ordnung p mit nur s_p Stufen?*

$p = s_p = 1$ Das explizite Euler-Verfahren ist die einzige Möglichkeit.

$p = s_p = 2$ Das Heun-Verfahren und das verbesserte Euler-Verfahren sind zwei Möglichkeiten von einer 1-Parameter-Familie mit Butcher-Tableau

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \alpha & \alpha & 0 \\ \hline & 1-\frac{1}{2\alpha} & \frac{1}{2\alpha} \end{array} \qquad (0 < \alpha \leq 1)$$

Für $p \geq 3$ gibt es viele Tricks und Hinweise, wie wir das Gleichungssystem günstig lösen können

z.B. betrachte die AWA

$$\begin{cases} \dfrac{dx}{dt} &= f(t) \\ x(t_0) &= x_0 \end{cases}$$

wobei f nur von t (und nicht von x) abhängt. Die Lösung lautet

$$x(t) = x_0 + \int_{t_0}^{t} f(s)ds$$

oder, mit $t_0 = 0$ und $t = h$,

$$x(h) = x_0 + \int_{0}^{h} f(s)ds.$$

Wir kennen viele Approximationsformeln (und deren Konvergenzordnungen) für solche Integrale:

$$\int_0^h f(t)dt = A_s(h) + O(h^{q_s+1})$$

wobei s die Stützstellenzahl ist, z.B.

$s = q_s = 1$ Rechteckregel (links) $A_1(h) = h\ f(0)$

$s = q_s = 2$ Trapez-Regel $A_2(h) = h\ \left[\frac{1}{2}f(0) + \frac{1}{2}f((h)\right]$

Im Allgemeinen können wir die Newton-Cotes-Formel benutzen:

$$\int_0^h f(t)dt = h \sum_{i=1}^{s} b_i f(c_i h) + O(h^{q_s+1})$$

wobei die b_i Gewichte sind und die Stützstelle $c_i h \in [0, h] \quad \Leftrightarrow \quad c_i \in [0, 1]$.
IDEE: benutze die b_i und c_i hier in einem Butcher-Tableau $\begin{array}{c|c} c & A \\ \hline & b \end{array}$ und versuche die Bedingungsgleichungen für geeignete $a_{i,j}$ zu lösen, um ein explizites RK-Verfahren mit $s_p = s$ Stufen und Ordnung p zu erhalten.

Frage *Ist dies immer möglich?*

Beispiel Betrachte die Simpsons-$\frac{3}{8}$-Regel

$$\int_0^h f(t)dt = h\ \left[\frac{1}{8}\ f(0) + \frac{3}{8}\ f\left(\frac{1}{3}\ h\right) + \frac{3}{8}\ f\left(\frac{2}{3}\ h\right) + \frac{1}{8}\ f(h)\right] + O(h^5)$$

(falls f glatt genug!), d.h.

$$b_1 = \tfrac{1}{8} \quad b_2 = \tfrac{3}{8} \quad b_3 = \tfrac{3}{8} \quad b_4 = \tfrac{1}{8}$$

$$c_1 = 0 \quad c_2 = \tfrac{1}{3} \quad c_3 = \tfrac{2}{3} \quad c_4 = 1$$

Unser Butcher-Tableau lautet

0	0	0	0	0
1/3	$a_{2,1}$	0	0	0
2/3	$a_{3,1}$	$a_{3,2}$	0	0
1	$a_{4,1}$	$a_{4,2}$	$a_{4,3}$	0
	1/8	3/8	3/8	1/8

Die Bedingungsgleichungen für Invarianz gegen Autonomisierung lauten:

$$(I_i) \quad c_i = \sum_{i=1}^{4} a_{i,j}, \qquad i = 1,2,3,4$$

$$(I_1) \quad 0 = 0 + 0 + 0 + 0$$
$$(I_2) \quad \boxed{1/3 = a_{2,1}} + 0 + 0 + 0$$
$$(I_3) \quad \boxed{2/3 = a_{3,1} + a_{3,2}} + 0 + 0$$
$$(I_4) \quad \boxed{1 = a_{4,1} + a_{4,2} + a_{4,3}} + 0$$

Die Koeffizienten $a_{3,1}$, $a_{3,2}$, $a_{4,1}$, $a_{4,2}$, $a_{4,3}$ sind noch frei.

Betrachte jetzt die Bedingungsgleichungen für die Ordnung

$$(O_1) \quad \boxed{\sum_{i=1}^{s} b_i = 1}$$

hier: $\sum_{i=1}^{4} b_i = \frac{1}{8} + \frac{3}{5} + \frac{3}{8} + \frac{1}{8} = 1$

$\Rightarrow$ die Ordnung $p \geq 1$

$$(O_2) \quad \boxed{\sum_{i=1}^{s} b_i c_i = \frac{1}{2}}$$

hier: $\sum_{i=1}^{4} b_i c_i = \frac{1}{8} \cdot 0 + \frac{3}{8} \cdot \frac{1}{3} + \frac{3}{5} \cdot \frac{2}{3} + \frac{1}{8} \cdot 1 = \frac{1}{2}$

$\Rightarrow$ die Ordnung $p \geq 2$

$$(O_3) \quad \boxed{\sum_{i=1}^{s} b_i c_i^2 = \frac{1}{3}}$$

hier:

$$\begin{aligned} \sum_{i=1}^{4} b_i c_i^2 &= \frac{1}{8} \cdot 0^2 + \frac{3}{8} \, (\frac{1}{3})^2 + \frac{3}{8} \, (\frac{2}{3})^2 + \frac{1}{8} \cdot 1^2 \\ &= \frac{3}{8} \cdot \frac{1}{9} + \frac{3}{8} \cdot \frac{4}{9} + \frac{1}{8} \\ &= \frac{1}{24} + \frac{1}{6} + \frac{1}{8} = \frac{8}{24} = \frac{1}{3} \end{aligned}$$

$$(O_4) \quad \boxed{\sum_{i=1}^{s} \sum_{j=1}^{s} b_i a_{i,j} c_j = \frac{1}{6}}$$

hier: $b_1 \sum_{j=1}^{4} a_{1,j} c_j = 0$, weil alle $a_{1,j} = 0$, und

$$
\begin{aligned}
b_2 \sum_{j=1}^{4} a_{2,j}c_j &= b_2[a_{2,1}c_1 + 0] = 0, \text{ weil } c_1 = 0 \\
b_3 \sum_{j=1}^{4} a_{3,j}c_j &= b_3[a_{3,1}c_1 + a_{3,2}c_2 + 0] \\
&= b_3 a_{3,2} \cdot c_2 = \frac{3}{8} \cdot a_{3,2} \cdot \frac{1}{3} \\
b_4 \sum_{j=1}^{4} a_{4j}c_j &= b_4[a_{4,1}c_1 + a_{4,2}c_2 + a_{4,3}c_3 + 0] \\
&= [a_{4,2}c_2 + a_{4,3}c_3] \\
&= \frac{1}{8}\, a_{4,2} \cdot \frac{1}{3} + \frac{1}{8}\, a_{4,3} \cdot \frac{2}{3}
\end{aligned}
$$

$$
(O_4) \Rightarrow \frac{1}{6} = \frac{1}{8}\, a_{3,2} + \frac{1}{24}\, a_{4,2} + \frac{1}{12}\, a_{4,3}
$$

d.h. $\boxed{3\, a_{3,2} + a_{4,2} + 2\, a_{4,3} = 4}$ $\qquad (O_4^*)$

Ordnung $p \geq 3$: $a_{3,1},\ a_{3,2},\ a_{4,1},\ a_{4,2},\ a_{4,3}$ müssen (I_3), (I_4) und (O_4^*) genügen

$$
\begin{cases} 3 \text{ lineare Gleichungen} \\ 5 \text{ Unbekannte} \end{cases}
$$

$\Rightarrow$ 2-Parameter-Familie von Ordnung $p = 3$ oder ≥ 3

d.h. mit b, c wie oben!

$$
(O_5) \qquad \boxed{\textstyle\sum_{i=1}^{5} b_i c_i^3 = \frac{1}{4}}
$$

hier

$$
\begin{aligned}
\sum_{i=1}^{4} b_i c_i^3 &= \frac{1}{8} \cdot 0^3 + \frac{3}{8} \left(\frac{1}{3}\right)^3 + \frac{3}{8} \left(\frac{2}{8}\right)^3 + \frac{1}{8}\, 1^3 \\
&= 0 + \frac{1}{72} + \frac{8}{72} + \frac{1}{8} = \frac{18}{72} = \frac{1}{4}
\end{aligned}
$$

$$(O_6) \quad \boxed{\sum_{i=1}^{s} \sum_{j=1}^{s} b_i c_i a_{i,j} c_j = \tfrac{1}{8}}$$

hier

$$\begin{aligned}
b_1 c_1 \sum_{j=1}^{4} a_{i,j} c_j &= 0 \\
b_2 c_2 \sum_{j=1}^{4} a_{2,j} c_j &= \tfrac{3}{8} \cdot \tfrac{1}{3} \cdot a_{2,1} c_1 = 0 \\
b_3 c_3 \sum_{j=1}^{4} a_{3,j} c_j &= \tfrac{3}{8} \cdot \tfrac{2}{3} \left[a_{3,1} \cdot 0 + a_{3,2} \cdot \tfrac{1}{3}\right] = \tfrac{1}{12}\, a_{3,2} \\
b_4 c_4 \sum_{j=1}^{4} a_{4,j} c_j &= \tfrac{1}{8} \cdot 1 \left[a_{4,1} \cdot 0 + a_{4,2} \cdot \tfrac{1}{3} + a_{4,3} \cdot \tfrac{2}{3}\right] \\
&= \tfrac{1}{24} \left[a_{4,2} + 2a_{4,3}\right]
\end{aligned}$$

$$(O_6) \Leftrightarrow \frac{1}{12}\, a_{3,2} + \frac{1}{24}\, a_{4,2} + \frac{2}{24}\, a_{4,3} = \frac{1}{8}$$

d.h. $\boxed{2\, a_{3,2} + a_{4,2} + 2\, a_{4,3} = 3}$ $\qquad (O_6^*)$

Mit (O_4^*) und (O_6^*) erhalten wir $\boxed{a_{3,2} = 1}$ und damit (aus I_3) $a_{3,1} = 2/3 - a_{3,2} = -\,1/3$, so gilt $\boxed{a_{3,1} = -\tfrac{1}{3}}$. Wir haben noch eine kombinierte Gleichung für $a_{4,2}$, $a_{4,3}$

$$\boxed{a_{4,2} + 2\, a_{4,3} = 1}$$

Wir haben auch (I_4) von oben

$$(I_4) \qquad \boxed{a_{4,1} + a_{4,2} + a_{4,3} = 1}$$

$$(O_7) \quad \boxed{\sum_{i=1}^{s} \sum_{j=1}^{s} b_i a_{i,j} c_j^2 = \tfrac{1}{12}}$$

hier

$$\begin{aligned}
b_1 \sum_{j=1}^{4} a_{1,j} c_j^2 &= 0 \quad \text{weil alle} \quad a_{1,j} = 0 \\
b_2 \sum_{j=1}^{4} a_{2,j} c_j^2 &= b_2 [a_{2,1} 0^2 + 0] = 0 \\
b_3 \sum_{j=1}^{4} a_{3,j} c_j^2 &= b_3 [a_{3,1} 0^2 + a_{s,2} c_2^2] = b_3 a_{3,2} c_2^2 \\
b_4 \sum_{j=1}^{4} a_{4,j} c_j^2 &= b_4 [a_{4,1} 0^2 + a_{4,2} c_2^2 + a_{4,3} c_3^2] \\
&= b_4 a_{4,2} c_2^2 + b_4 a_{4,3} c_3^2
\end{aligned}$$

$$\Leftrightarrow \quad \frac{3}{8}\, a_{3,2} \left(\frac{1}{3}\right)^2 + \frac{1}{8}\, a_{4,2} \left(\frac{1}{3}\right)^2 + \frac{1}{8}\, a_{4,3} \left(\frac{2}{3}\right)^2 = \frac{1}{12}$$

d.h. mit $a_{3,2} = 1$ gilt

$$\boxed{a_{4,2} + 4a_{4,3} = 3} \qquad (O_7^*)$$

Wir lösen (O_4^*), (O_6^*) und (O_7^*) und erhalten

$$\boxed{a_{4,2} = -1} \qquad \boxed{a_{4,3} = 1}$$

Mittels (I_4) bekommen wir $\boxed{a_{4,1} = 1}$

Nun haben wir alle freie $a_{i,j}$ bestimmt, d.h. wir haben das Butcher-Tableau

0	0	0	0	0
1/3	1/3	0	0	0
2/3	−1/3	1	0	0
1	1	−1	1	0
	1/8	3/8	3/8	3/8

Zum Schluß müssen wir bestätigen, dass diese Koeffizienten der Bedingungsgleichung

$$(O_8) \qquad \boxed{\sum_{i=1}^{s} \sum_{j=1}^{s} \sum_{k=1}^{s} b_i a_{i,j} a_{j,k} c_k = \tfrac{1}{24}}$$

genügen $\Rightarrow$ JA

Das explizite RK-Verfahren mit dem obigen Butcher-Tableau genügt den Bedingungsgleichungen $(O_1) - (O_8)$. Deshalb besitzt das Verfahren die Konvergenzordnung $p \geq 4$.

Aber $4 \leq p \leq s \leq 4$.

$$\Rightarrow \quad p = 4 \qquad \text{ist die genaue Ordnung.}$$

Beispiel Simpsons Regel

$$\int_0^h f(t)dt = h\ \left[\frac{1}{6}\ f(0) + \frac{4}{6}\ f\left(\frac{1}{2}\ h\right) + \frac{1}{6}\ f(h)\right] + O(h^5)$$

Dies sieht aus, als ob wir nur 3 Stufen haben. Aber wir trennen den Mittelterm

$$\rightarrow\quad h\ \left[\frac{1}{6}\ f(0) + \frac{2}{6}\ f\left(\frac{1}{2}\ h\right) + \frac{2}{6}\ f\left(\frac{1}{2}\ h\right) + \frac{1}{6}\ f(h)\right]$$

und fangen wie oben an, mit

$$\begin{array}{llll} c_1 = 0, & c_2 = c_3 = \frac{1}{2} & c_4 = 1 \\ b_1 = \frac{1}{6}, & b_2 = b_3 = \frac{2}{6} & b_4 = \frac{1}{6} \end{array}$$

Wir leiten das folgende Butcher-Tableau (<u>nicht</u> eindeutig!) her

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

$\Leftrightarrow$ das "klassische" Runge-Kutta-Verfahren

$$\left\{\begin{array}{lcll} k_1 &=& f(t_n, x_n) & \\ k_2 &=& f(t_n + \frac{1}{2}\ h_n, x_n + \frac{1}{2}\ h_n k_1) & \leftarrow\ k_1 \text{ hier} \\ k_3 &=& f(t_n + \frac{1}{2}\ h_n, x_n + \frac{1}{2}\ h k_2) & \leftarrow\ k_2 \text{ hier} \\ k_4 &=& f(t_n + h_n, x_n + h_n k_3) & \end{array}\right.$$

mit

$$x_{n+1} = x_n + \frac{1}{6}\ h_n(k_1 + 2k_2 + 2k_3 + k_4)$$

<u>hier</u> mit $s = 4$ Stufen und Ordnung $p = 4$.

4.4 Eingebettete Runge-Kutta-Verfahren

Betrachte eine AWA

$$\left.\begin{array}{lcl} \frac{dx}{dt} &=& f(t,x) \\ x(t_0) &=& x_0 \end{array}\right\}\quad x \in \mathbb{R}^d,\ t \in [t_0, T]$$

und ein Einschrittverfahren p-ter Ordnung

$$x_{n+1} = x_n + h_n \Phi(h_n, t_n, x_n)$$

Fragen

1. *Wie genau ist diese numerische Approximation?*
2. *Wie sollen wir die Schrittweiten h_n wählen, um eine erwünschte Genauigkeit zu versichern?*

Die Lösung $x(t, t_0, x_0)$ der AWA ist meistens nicht bekannt $\Rightarrow$ wir können den globalen Diskretisierungsfehler (DF) nicht direkt berechnen. Die Abschätzung des globalen DF lautet

$$|x(t_n, t_0, x_0) - x_n| \leq K_T h^p, \qquad h = \max_n \ h_n$$

ist auch nicht besonders hilfreich – die Konstante K_T ist meistens unbekannt oder wir haben (am besten) nur eine sehr grobe Abschätzung von oben für K_T.

Algorithmen für Schrittweitensteuerung benutzen ein zweites Verfahren höherer Ordnung, um den lokalen Diskretisierungsfehler des ersten Verfahrens abzuschätzen.

Betrachte jetzt ein zweites Einschrittverfahren q-ter Ordnung mit $q > p$:

$$x_{n+1} = x_n + h_n \Phi^*(h_n, t_n, x_n)$$

Berechne:

$$\begin{cases} x_{n+1}^{(p)} & := \quad x_n + h_n \Phi(h_n, t_n, x_n) \\ x_{n+1}^{(q)} & := \quad x_n + h_n \Phi^*(h_n, t_n, x_n) \end{cases}$$

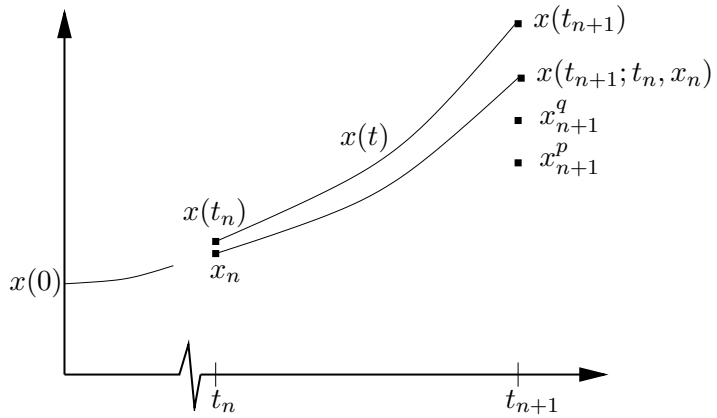
für die selben h_n, t_n und x_n.

Definiere:

$$E_{(p,q)}^{(n+1)} \ := \ \left| x_{n+1}^{(q)} - x_{n+1}^{(p)} \right|$$

Dann ist

$$E_{(p,q)}^{(n+1)} \simeq |x(t_{n+1}; t_n, x_n) - x_{n+1}^{(p)}|,$$



eine Näherung des lokalen Diskretisierungsfehler des ersten Verfahrens, weil $x^{(q)}_{n+1}$ eine genauere Approximation für die gegebenen Daten t_{n+1}, t_n, x_n ist.

Dabei bedeutet $x(t_{n+1}, t_n, x_n)$ die exakte Lösung zu $\frac{dx}{dt} = f(t, x)$ mit $x(t_n) = x_n$ zum Zeitpunkt t_{n+1}.

Sei TOL die erwünschte Genauigkeit

1. Fall $\qquad E^{(n+1)}_{(p,q)} \;>\; \text{TOL}$

$\Rightarrow$ wiederhole die Berechnung mit der halbierten Schrittweite $\widetilde{h}_n = \frac{1}{2}\, h_n$.

2. Fall $\qquad E^{(n+1)}_{(p,q)} \;\leq\; \text{TOL}$

$\Rightarrow$ nächster Schritt $x_{n+1} = x^{(p)}_{n+1}$, $t_{n+1} = t_n + h_n$

$\Rightarrow$ berechne $E^{(n+2)}_{(p,q)}$ mit x_{n+1}, t_{n+1} und $h_{n+1} = h_n$.

Verfeinerung: Falls $E^{(n+1)}_{(p,q)} \;<<\; \text{TOL}$, wiederhole die Berechnung von $E^{(n+1)}_{(p,q)}$ mit der verdoppelten Schrittweite $\widetilde{h}_n = 2h_n$ (und den selben t_n, x_n!).

Die Algorithmen sind natürlich viel komplizierter, aber die Grundidee ist wie oben.

Frage *Wie sollen wir das zweite Verfahren höherer Ordnung wählen?*

Im Prinzip beliebig! Aber in der Praxis sollen wir versuchen, den zusätzlichen rechnerischen Aufwand zu verringern. Zum Beispiel: das zweite Verfahren könnte Funktionenauswertungen von dem ersten Verfahren verwenden – soviele wie möglich.

Für explizite RK-Verfahren bedeutet dies, dass der $(c|A)$-Teil des Butcher-Tableaus des zweiten Verfahrens den entsprechenden $(c|A)$-Teil des ersten Verfahrens enthalten soll.

Beispiele

1. Explizites Euler-Verfahren/Heun-Verfahren

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

$(p = s_p = 1)$ $\qquad$ $(q = s_q = 2)$

2. Verfahren ohne Name

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 2/3 & 2/3 & 0 \\ \hline & 1/4 & 3/4 \end{array} \qquad \begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 2/3 & 2/3 & 0 & 0 \\ \hline 2/3 & 0 & 2/3 & 0 \\ \hline & 1/4 & 3/8 & 3/8 \end{array}$$

$(p = s_p = 2)$ $\qquad$ $(q = s_q = 3)$

Solche Verfahren heißen eingebettete Runge-Kutta-Verfahren.

Bemerkungen Im Allgemeinen sind die b-Vektoren nicht eingebettet und die optimalen Stufenzahlen s_p und s_q nicht möglich.

Betrachte das folgende Paar eingebetteter Runge-Kutta-Verfahren mit Butcher-Tableaux

$$\begin{array}{c|c} c^{(p)} & A^{(p)} \\ \hline & b^{(p)} \end{array} \quad \text{und} \quad \begin{array}{c|c} c^{(q)} & A^{(q)} \\ \hline & b^{(q)} \end{array}$$

Ordnung p
$s^{(p)} \geq s_p$ Stufen

Ordnung q
$s^{(q)} \geq s_q$ Stufen

Hier gelten:

(1) $p < q$ und $s^{(p)} \leq s^{(q)}$

(2) $c_i^{(q)} = c_i^{(p)}, \quad i = 1, \ldots, s^{(p)}$,

(3) $a_{i,j}^{(q)} = a_{i,j}^{(p)}, \quad i, j = 1, \ldots, s^{(p)}$.

Statt 2 Butcher-Tableaux definiert man ein kombiniertes Butcher-Tableau

$c^{(q)}$	$A^{(q)}$
	$(b^{(p)}, 0)$
	$b^{(q)}$

$\leftarrow$ mit $s^{(q)} - s^{(p)}$ Nullstellen

(1) explizites Euler-Heun-Verfahren

0	0	0
1	1	0
	1	0
	1/2	1/2

(2) (von oben)

0	0	0	0
2/3	2/3	0	0
2/3	0	2/3	0
	1/4	3/4	0
	1/4	3/8	3/8

Die obigen Beispiele sind nur Lehrbuch-Verfahren.

Ein berühmtes, oft benutztes Beispiel ist das RK-Fehlberg-4(5)-Paar, von Fehlberg vorgeschlagen, mit dem kombinierten Butcher-Tableau

0	0	0	0	0	0	0
$\frac{1}{4}$	$\frac{1}{4}$	0	0	0	0	0
$\frac{3}{8}$	$\frac{3}{32}$	$\frac{9}{32}$	0	0	0	0
$\frac{12}{13}$	$\frac{1932}{2197}$	$-\frac{7200}{2197}$	$\frac{7296}{2197}$	0	0	0
1	$\frac{439}{216}$	-8	$\frac{3680}{513}$	$-\frac{845}{4104}$	0	0
$\frac{1}{2}$	$-\frac{8}{27}$	2	$-\frac{3544}{2565}$	$\frac{1859}{4104}$	$-\frac{11}{40}$	0
	$\frac{25}{216}$	0	$\frac{1408}{2565}$	$\frac{2197}{4104}$	$-\frac{1}{5}$	0
	$\frac{16}{135}$	0	$\frac{6656}{12825}$	$\frac{28561}{56430}$	$-\frac{9}{50}$	$\frac{2}{55}$

mit $p = 4,\ s^{(p)} = 5$ und $q = 5,\ s^{(q)} = 6$

Die Ordnung mit Schrittweitesteuerung ist $p = 4$, sonst $q = 5$.

Bemerkung Die Koeffizienten sehen etwas komisch aus. Die sind gewählt, um den Ordnungsbedingungen zu genügen und die Konstante des lokalen Diskretisierungsfehlers zu minimisieren.

Fehlberg hat auch einen Trick ("Fehlberg-Trick") erfunden, um den Rechneraufwand zu verringern.

Sei $s^{(p)} = s,\ s^{(q)} = s + 1$ mit $q \geq p + 1$.

Wähle

- $c_{s+1}^{(q)} = 1$
- $a_{s+1,j}^{(q)} = \begin{cases} b_j^{(p)}, & j = 1, \ldots, s \\ 0, & j = s+1 \end{cases}$

Dann haben wir

$$k_i^{(q)}(h_n, t_n, x_n) \equiv k_i^{(p)}(h_n, t_n, x_n), \quad i = 1, \ldots, s,$$

sowie

$$\begin{aligned} k_{s+1}^{(q)}(h_n, t_n, x_n) &= f\left(t_n + h_n, x_n + h_n \ \textstyle\sum_{j=1}^{s+1} \ a_{s+1,j}^{(q)} \ k_j^{(q)}\right) \\ &= f\left(t_n + h_n, x_n + h_n \ \textstyle\sum_{j=1}^{s+1} \ b_j^{(p)} k_j^{(p)}\right) \\ &= f\left(t_{n+1}, x_{n+1}^{(p)}\right) \\ &= k_1^{(p)}\left(h_{n+1}, t_{n+1}, x_{n+1}^{(p)}\right) \end{aligned}$$

d.h. die letzte Stufe zur Zeit t_n wird die erste Stufe des Verfahrens p-ter Ordnung zur Zeit t_{n+1}. Deswegen sparen wir eine Funktionenauswertung pro Schritt

z.B. das RK-Dormand-Prince-4(5)-Paar (siehe Lehrbücher für das Tableau) mit

$$s = 6 \ \text{ hier}, \quad p = 4, \quad q = 5$$

Bemerkung Wir können auch implizite Runge-Kutta-Verfahren in eingebetteten Paaren benutzen.

4.5 Die Ordnungsbedingungen (nochmal)

Sei $\begin{array}{c|c} c & A \\ \hline & b \end{array}$ das Butcher-Tableau eines RK-Verfahrens (explizit oder implizit) p-ter Ordnung und mit s Stufen, das invariant gegen Autonomisierung ist.

Definiere:

$$C = \text{diag}\,(c_1, \ldots, c_s) = \begin{bmatrix} c_1 & & \bigcirc \\ & c_2 & \\ & \ddots & \\ \bigcirc & & c_s \end{bmatrix} \quad s \times s \text{ diagonale Matrix}$$

$$\mathbf{1} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \quad s - \text{dimensionaler Vektor}$$

Dann haben die Bedingungsleichungen $(O_1) - (O_5)$ und $(O_7) - (O_8)$ eines expliziten RK-Verfahrens die Form

$$\boxed{bA^j \; C^{\ell-1} \; \mathbf{1} = \frac{(\ell-1)!}{(\ell+j)!}} \qquad (O_{(j,\ell)})$$

mit $1 \leq j + \ell \leq p$, $\ell \geq 1$, $j \geq 0$ und $p = 1, \ldots, 4$.

Leider ist die Bedingung (O_6) nicht von dieser Form, sondern der Form

$$bCAC\mathbf{1} = \frac{1}{8}.$$

Daher sind die Bedingungen $(O_{(j,l)})$ mit $1 \leq j + \ell \leq p$ und $\ell \geq 1$, $j \geq 0$ nur notwendige Bedingungen für die Ordnung $p = 1, 2, 3, 4$.

Es folgt sofort, dass $p \leq s$ für ein explizites RK-Verfahren!.

Warum? $\quad a_{i,j} = 0 \; \forall j \geq i \quad \Rightarrow \quad A^s \equiv 0$

$$(O_{(s,1)}) \qquad \Rightarrow \quad bA^s\ \mathbf{1} = 0 \neq \frac{1}{(1+s)!}$$

Die Bedingungen $(O_{(j,\ell)})$ mit $1 \leq j + l \leq p$ sind auch notwendige Bedingungen für die Ordnung p mit $p \neq 4$ und für implizierte RK-Verfahren.

Betrachte die AWA

$$\begin{cases} \dfrac{dx}{dt} &= x + t^{\ell-1} \\ x(0) &= 0 \end{cases} \qquad (\ell \geq 1)$$

Die Lösung lautet

$$x(t) = \int_0^t e^{t-s}\ s^{\ell-1}\, ds$$

mit Ableitungen

$$\begin{cases} x^{(i)}(0) &= 0, & i = 1, \ldots, \ell - 1 \\ x^{(\ell+j)}(0) &= (\ell-1)!, & j \geq 0 \end{cases}$$

$\Rightarrow$ Taylor-Entwicklung

$$x(h) = \sum_{j=0}^{\infty} \frac{(\ell-1)!}{(\ell+j)!}\ h^{\ell+j}$$

Der erste Schritt des RK-Verfahrens mit $x_0 = 0$ und $h_0 = h$ für die Funktion

$$f(t,x) = x + t^{\ell-1}$$

lautet

$$\begin{cases} k_i &= h\ \sum_{j=1}^{s}\ a_{ij}k_j + (c_i h)^{\ell-1}, & i = 1, \ldots, s, \\ x_1 &= h\ \sum_{i=1}^{s}\ b_i k_i \end{cases}$$

Definiere $K = \begin{pmatrix} k_1 \\ \vdots \\ k_s \end{pmatrix}$

Dann lautet das RK-Verfahren

$$\Rightarrow \qquad \begin{cases} K &= hAK + h^{\ell-1}\ C^{\ell-1}\mathbf{1}, \\ x_1 &= hbK \qquad \text{(Skalar-Produkt)} \end{cases}$$

Aber $K = h^{\ell-1}(I - hA)^{-1}C^{\ell-1}\mathbf{1}$ invertierbar, weil $h \ll 1$.

$$\begin{aligned} x_1 &= h^\ell b(I - hA)^{-1}C^{\ell-1}\mathbf{1} \\ &= h^\ell b(I + hA + \ldots + h^j A^j + \ldots)C^{\ell-1}\mathbf{1} \\ &= \textstyle\sum_{j=0}^{\infty} h^{\ell+j} bA^j C^{\ell-1}\ \mathbf{1} \end{aligned}$$

Der lokale Diskretisierungsfehler genügt dann:

$$L_0 = |x(h) - x_1| = \sum_{j=0}^{\infty} h^{\ell+j} \left(\frac{(\ell-1)!}{(\ell+j)!} - bA^j C^{\ell-1}\ \mathbf{1} \right)$$

$$\Rightarrow \qquad L_0 \sim 0(h^{p+1})$$

wenn die $(O_{(j,\ell)})$ gelten für $1 \le \ell + j \le p$.

Insbesondere müssen die Bedingungen

$$(O_{(0,\ell)}) \qquad bC^{\ell-1}\ \mathbf{1} = \frac{1}{\ell}, \qquad \ell = 1, \ldots, p$$

gelten.

Diese sind tatsächlich reine Integrationsbedingungen. Betrachte die AWA

$$\begin{cases} \dfrac{dx}{dt} &= f(t) \qquad\qquad (\text{kein } x \text{ in } f \text{ hier!}) \\ x(0) &= 0 \end{cases}$$

$$\begin{cases} \text{Lösung}: & x(h) &= \int_0^h f(t)\,dt \\ \text{RK-V}: & x_1 &= h\ \sum_{i=1}^{s} b_i f(c_i h) \end{cases}$$

$$\Rightarrow \qquad \left| \int_0^h f(t)\,dt - h \sum_{i=1}^{s} b_i f(c_i h) \right| \sim 0(h^{p+1})$$

d.h. die Integrationsformel

$$\int_0^h f(t)\,dt \simeq h \sum_{l=1}^{s} b_i f(c_i h)$$

hat Ordnung $p+1$

$$\boxed{f(t) = t^{\ell-1}, \quad \ell = 1, \ldots, p} \quad \Rightarrow \begin{cases} x(h) &= \frac{1}{\ell}\, h^\ell \\ x_1 &= h \, \sum_{i=1}^s \, b_i (c_i h)^{\ell-1} \\ &= h^\ell \, \sum_{i=1}^s \, b_i c_i^{\ell-1} = h^\ell b C^{\ell-1} \, \mathbf{1} \end{cases}$$

$(O_{(0,\ell)})$ mit $\ell = 1, \ldots, p$ $\Rightarrow$ Integrationsformel ist genau für Polynome bis zum Grad $p-1$.

Kapitel 5

Implizite Runge-Kutta-Verfahren

Ein s-stufiges Runge-Kutta-Verfahren mit Butcher-Tableau $\begin{array}{c|c} c & A \\ \hline & b \end{array}$ ist ein explizites Verfahren, falls

$$a_{i,j} = 0 \quad \forall j \geq i.$$

Sonst ist es ein implizites Verfahren.

Beispiele

(1) Runge-Kutta-Gauß-Verfahren

$$\begin{array}{c|cc} \frac{3-\sqrt{3}}{6} & \frac{1}{4} & \frac{3-2\sqrt{3}}{12} \\ \frac{3+\sqrt{3}}{6} & \frac{3+\sqrt{3}}{12} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \qquad \left\{ \begin{array}{l} \text{hier} \\ s = 2 \\ p = 2s = 4 \end{array} \right.$$

(2) Runge-Kutta-Radau-Verfahren (stets $c_s = 1$)

$$\begin{array}{c|cc} \frac{1}{3} & \frac{5}{12} & -\frac{1}{12} \\ 1 & \frac{3}{4} & \frac{1}{4} \\ \hline & \frac{3}{4} & \frac{1}{4} \end{array} \qquad \left\{ \begin{array}{l} \text{hier} \\ s = 2 \\ p = 2s - 1 = 3 \end{array} \right.$$

(3) Runge-Kutta-Lobatto-Verfahren (stets $c_1 = 0$ und $c_s = 1$)

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \qquad \left\{ \begin{array}{l} \text{hier} \\ s = 2 \\ p = 2s - 2 = 2 \end{array} \right. \qquad \text{(Trapez-Verfahren!)}$$

$\Rightarrow$ ein implizites RK-Verfahren kann Ordnung $p > s$ haben

5.1 Ordnung, Stufenanzahl und Lösbarkeit

SATZ Für ein implizites RK-Verfahren p-ter Ordnung mit s Stufen gilt $p \leq 2s$

Beweis

Betrachte die AWA

$$\frac{dx}{dt} = x, \quad x(0) = 1,$$

mit Lösung $x(t) = e^t$.

$$\Rightarrow \quad x(h) = 1 + h + \ldots + \frac{h^p}{p!} + O(h^{p+1})$$

Für die Funktion $f(t, x) = x$ lautet das RK-Verfahren mit Butcher-Tableau $\begin{array}{c|c} c & A \\ \hline & b \end{array}$

$$\left\{ \begin{array}{lcll} k_i & = & 1 + h \sum_{i=1}^{s} a_{i,j} k_j, & i = 1, \ldots, s \\ x_1 & = & 1 + h \sum_{i=1}^{s} b_i k_i & \end{array} \right.$$

für $x_0 = 1$, $t_0 = 0$ und $h_0 = h$.

d.h.

$$\left\{ \begin{array}{lcl} K & = & \mathbf{1} + hAK \\ x_1 & = & 1 + hbK \end{array} \right. \qquad \text{wobei} \quad K = \begin{pmatrix} k_1 \\ \vdots \\ k_s \end{pmatrix}$$

$\Rightarrow \quad x_1 = 1 + hb(I - hA)^{-1}\, \mathbf{1}$

Definiere $R(z) = 1 + zb(I - zA)^{-1}\,\mathbf{1}, \quad z \in \mathbb{C}$

Hilfsatz 6.30 (Deuflhard/Bornemann, Seite 230)

$$R(z) = \frac{P(z)}{Q(z)}\ ,$$

wobei $P, Q \in \mathcal{P}_s$ (Polynome von Grad $\leq s$) mit $P(0) = Q(0) = 1$ sind.

Später in Kapitel 9 beweisen wir diese Aussage.

Hilfsatz 6.4 (Deuflhard/Bornemann, Seite 201)

$$R(z) - e^z \sim O(z^{p+1}) \text{ mit } \ p \leq \ \text{Grad}(P) + \ \ \text{Grad}(Q)$$

In unserem Fall gilt $p \leq s + s = 2s$

SATZ Genüge die Vektorfeldfunktion $f : \ [t_0, T] \times \mathbb{R}^d \to \mathbb{R}^d$ einer globalen Lipschitz-Bedingung mit Lipschitz-Konstante L und genüge die Schrittweite h der Ungleichung

$$h < \frac{1}{L\|A\|_\infty},$$

wobei $\|A\|_\infty = \max_i \ \sum_{j=1}^{s} \ |a_{i,j}|$.
Dann sind die Stufengleichungen des impliziten Runge-Kutta-Verfahrens eindeutig lösbar.

Beweis

Die Stufengleichungen lauten

$$k_i = f\left(t + c_i h,\ x + h\ \sum_{j=1}^{s}\ a_{i,j}k_j\right), \quad i = 1, \ldots, s, \quad (h, t, x \ \text{fest})$$

Schreibe $K = \begin{pmatrix} k_1 \\ \vdots \\ k_s \end{pmatrix} \in \mathbb{R}^{sd}$ und $\|K\| = \max_i |k_i|_d$.

Definiere eine Abbildung

$$F : \ \mathbb{R}^{sd} \to \mathbb{R}^{sd}$$

durch $Z = \begin{pmatrix} z_1 \\ \vdots \\ z_s \end{pmatrix}$, $Y = \begin{pmatrix} y_1 \\ \vdots \\ y_s \end{pmatrix}$ und

$$z_i = f\left(t + c_i h,\ x + h \sum_{j=1}^{s} a_{i,j} y_j\right), \quad i = 1, \ldots, s.$$

Für $Z = F(Y)$ und $Z' = F(Y')$ gilt dann

$$\begin{aligned} |z_i - z_i'|_d &= \left| f\left(t + c_i h,\ x + h \sum_{j=1}^{s} a_{i,j} y_j\right) - f\left(t + c_i h,\ x + h \sum_{j=1}^{s} a_{i,j} y_j'\right) \right|_d \\ &\leq Lh \left| \sum_{j=1}^{s} a_{i,j}(y_j - y_j') \right|_d \qquad \text{globale Lipschitz-Bedingung} \\ &\leq Lh \left(\sum_{j=1}^{s} |a_{i,j}| \right) \max_j |y_j - y_j'|_d \end{aligned}$$

$$\Rightarrow \quad \max_i |z_i - z_i'|_d \ \leq\ Lh \max_i \left(\sum_{j=1}^{s} |a_{i,j}| \right) \ \max_j |y_j - y_j'|_d$$

d.h. $\|Z - Z'\| \leq Lh\|A\|_\infty \ \|Y - Y'\|$

$\Rightarrow$ die Abbildung F ist eine Kontraktion und besitzt einen eindeutigen Fixpunkt

$$Z^* = F(Z^*)$$

Bemerkungen

(1) Der Fixpunkt $Z^* = Z^*(h)$ hängt von h ab. Man kann auch beweisen, dass $h \to Z^*(h)$ stetig ist.

(2) Im Prinzip können wir Z^* durch sukzessive Approximationen berechnen – in der Praxis benutzen wir die Newton'sche Methode.

Frage: *Wie können wir ein implizites RK-Verfahren einer erwünschten Ordnung herleiten?* -

Wir haben nur notwendige Bedingungsgleichungen für die Koeffizienten!

Hinweis Integrationsformel, z.B. Gauß-Quadraturformel.

5.2 Kollokation

Betrachte eine skalare AWA

$$\begin{cases} \dfrac{dx}{dt} &= f(t,x) \\ x(t_0) &= x_0 \end{cases} \qquad x \in \mathbb{R}^1, t \in [t_0, T]$$

und eine Unterteilung des Intervalls $[t_0, T]$,

$$t_0 < t_1 < \ldots < t_n < t_{n+1} < \ldots < t_N = T$$

Setze voraus, dass

(1) $0 \leq c_1 < c_2 < \ldots < c_s \leq 1$

(2) $y \in \mathbb{R}^1,\ \tau, \tau + h \in [t_0, T],\ h > 0$

Dann existiert ein eindeutiges Polynom $\phi \in \mathcal{P}_s$ (Grad $\leq s$) mit

$$\begin{cases} \phi(\tau) = y \\ \dfrac{d\phi}{dt}\,(\tau + c_i h) = f\left(\tau + c_i h, \phi(\tau + c_i h)\right), \quad i = 1, \ldots, s \end{cases}$$

Diese Konstruktion ergibt ein eindeutiges implizites RK-Verfahren mit s Stufen.

Seien $\tau = t_n$, $y = x_n$, $h = h_n$ und $t_{n+1} = t_n + h_n$.

Definiere $k_i = f\left(t_n + c_i h_n, \phi(t_n + c_i h_n)\right), \quad i = 1, \ldots, s$

Dann ist $\phi'(t)$ ein Polynom höchsten Grades $s-1$ mit den s Datenstellen

$$\left(t_n + c_i h_n, k_i\right), \qquad i = 1, \ldots, s$$

d.h. $\phi'(t_n + c_i h_n) = k_i, \qquad i = 1, \ldots, s$

$\Rightarrow$ Wir können ϕ' durch ein Lagrange'sches Interpolationspolynom darstellen

$$(*) \qquad \boxed{\phi'(t) = \textstyle\sum_{j=1}^{s}\ k_j L_j\ \left(\dfrac{t - t_n}{h_n}\right)}$$

wobei

$$L_j(r) = \prod_{\substack{i=1 \\ i \neq j}}^{s} \frac{r - c_i}{c_j - c_i}, \qquad r \in [0,1]$$

das j-te Lagrange'sche Polynom auf dem Intervall $[0,1]$ mit Stützstellen $0 \leq c_1 < \ldots < c_j \leq 1$ ist.

Bemerkung

Für eine s-mal stetig differenzierbare Funktion $F: [0,1] \to \mathbb{R}$ haben wir

$$F(r) = \sum_{j=1}^{s} F(c_j) L_j(r) + \text{ Fehler}.$$

Das Interpolationspolynom ist fehlerfrei, d.h. genau, für alle Polynome höchsten Grades $s-1$. Insbesondere gilt für $r \in [0,1]$

$$(**) \qquad \boxed{r^{k-1} = \sum_{j=1}^{s} c_j^{k-1} L_j(r)} \qquad k = 1, \ldots, s$$

Definiere

$$\begin{aligned} a_{i,j} &= \int_0^{c_i} L_j(r)\, dr \qquad i,j = 1, \ldots, s \\ b_j &= \int_0^{1} L_j(r)\, dr \qquad j = 1, \ldots, s \end{aligned}$$

Jetzt integriere $(*)$ von t_n bis $t_n + c_i h_n$

$$\begin{aligned} \phi(t_n + c_i h_n) - \phi(t_n) &= \int_{t_n}^{t_n + c_i h_n} \sum_{j=1}^{s} k_j L_j\left(\frac{t - t_n}{h_n}\right) dt \\ &= h_n \sum_{j=1}^{s} k_j \int_0^{c_i} L_j(r)\, dr \\ &= h_n \sum_{j=1}^{s} a_{i,j} k_j \end{aligned}$$

Aber $\phi(t_n) = x_n$.

$$\Rightarrow \quad \phi(t_n + c_i h_n) = x_n + h_n \sum_{j=1}^{s} a_{i,k} k_j, \qquad i = 1, \ldots, s$$

wobei

$$k_i = f(t_n + c_i h_n,\ \phi(t_n + c_i h_n)), \qquad i = 1, \ldots, s$$

$$\Rightarrow \quad k_i = f(t_n + c_i h_n, x_n + h_n \sum_{j=1}^{s} a_{ij} k_j), \qquad i = 1, \ldots, s$$

Endlich integriere $(*)$ von t_n bis $t_{n+1} = t_n + h_n$

$$\begin{aligned} \phi(t_{n+1}) - \phi(t_n) &= \int_{t_n}^{t_{n+1}} \sum_{j=1}^{s} k_j L_j \left(\frac{t - t_n}{h_n} \right) dt \\ &= h_n \sum_{j=1}^{s} k_j \int_0^1 L_j(r)\, dr \\ &= h_n \sum_{j=1}^{s} b_j k_j \end{aligned}$$

<u>d.h.</u> $\phi(t_{n+1}) = \phi(t_n) + h_n \sum\limits_{j=1}^{s} b_j k_j$

<u>oder</u>

$$x_{n+1} = x_n + h_n \sum_{j=1}^{s} b_j k_j$$

Die Parameter c, A, b definieren ein RK-Verfahren mit Butcher-Tableau

c	A
	b

SATZ

Sei $\begin{array}{c|c} c & A \\ \hline & b \end{array}$ das Butcher-Tableau eines durch Kollokation definierten RK-Verfahrens.

(1) das RK-Verfahren ist invariant gegen Autonomisierung;

(2) die Koeffizienten b, A genügen den Gleichungen

(a) $\sum_{j=1}^{s} b_j c_j^{k-1} = \frac{1}{k}, \quad k = 1, \ldots, s$

und

(b) $\sum_{j=1}^{s} a_{i,j} c_j^{k-1} = \frac{1}{k}\, c_i^k, \quad i, k = 1, \ldots, s$

Beweis Von den Definitionen

$$\begin{aligned} \sum_{j=1}^{s} a_{i,j} &= \sum_{j=1}^{s} \int_0^{c_i} L_j(r)\, dr \\ &= \int_0^{c_i} \Big[\sum_{j=1}^{s} c_j^0 L_j(r) \Big]\, dr = \int_0^{c_i} r^0\, dr = c_i \end{aligned}$$

$\Rightarrow$ Invariant gegen Autonomisierung.
Für die zweite Behauptung haben wir

$$\begin{aligned} \sum_{j=1}^{s} b_j c_j^{k-1} &= \sum_{j=1}^{s} c_j^{k-1} \int_0^{1} L_j(r)\, dr \\ &= \int_0^{1} \Big[\sum_{j=1}^{s} c_j^{k-1} L_j(r) \Big]\, dr = \int_0^{1} r^{k-1}\, dr = \frac{1}{k} \end{aligned}$$

und

$$\sum_{j=1}^{s} a_{i,j} c_j^{k-1} = \sum_{j=1}^{s} c_j^{k-1} \int_0^{c_i} L_j(r)\, dr$$

$$= \int_0^{c_i} \Big[\sum_{j=1}^{s} c_j^{k-1} L_j(r)\Big]\, dr = \int_0^{c_i} r^{k-1}\, dr = \frac{1}{k}\, c_i^k$$

Bemerkung

Für ein RK-Verfahren des Kollokationstyps haben wir s vorgegeben Koeffizienten $c_1, \ldots, c_s$.

Dann bestimmen wir die anderen s^2+s Koeffizienten $a_{1,1}, \ldots, a_{s,s}, b_1, \ldots, b_s$ durch Integration der Lagrange'schen Interpolationspolynome oder durch die $s^2 + s$ Bedingungsgleichungen (2a) und (2b).

NB wir haben die Gleichungen (2a) schon gesehen, d.h. in der Form

$$b\, C^{k-1}\, \mathbf{1} = \frac{1}{k}$$

d.h. $(O_{(0,k)})$, wobei $(O_{(j,k)})$ lautet: $b\, A^j\, C^{k-1}\, \mathbf{1} = \dfrac{(k-1)!}{(k+j)!}$

Diese sind Bedingungsgleichungen für die Integrationsformel

$$\int_0^h f(t)dt = h \sum_{j=1}^{s} b_j f(c_j h) + \text{ Fehler}.$$

SATZ Ein durch Kollokation definiertes RK-Verfahren ist genau von der Ordnung p, wenn die Ordnung der Integrationsformel mit Gewichten b_1, $\ldots$, b_s und Stützstellen $c_1, \ldots, c_s$ p ist.

Beweis Siehe Deuflhard/Bornemann, Satz 6-40, Seite 244 +

Korollar Ein Runge-Kutta-Kollokationsverfahren mit s Stufen hat Ordnung $p \geq s$

Beispiele

Die RK-Gauß/Radau/Lobatto-Verfahren sind alle Verfahren des Kollokationstyps mit Ordnung

$p = 2s$	Gauß	(alle c_i frei)
$p = 2s-1$	Radau	$(c_s = 1)$
$p = 2s-2$	Lobatto	$(c_1 = 0,\ c_s = 1)$

Ein "Nicht"-Beispiel

Das implizite RK-Verfahren mit Butcher-Tableau

$$\begin{array}{c|cc} 0 & \frac{1}{4} & -\frac{1}{4} \\ \frac{2}{3} & \frac{3}{12} & \frac{5}{12} \\ \hline & \frac{1}{4} & \frac{3}{4} \end{array}$$

ist kein Kollokationsverfahren – das eindeutige RK-Kollokationsverfahren mit $c = (0, 2/3)^T$ hat das Butcher-Tableau

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

Kein Verbrechen!

Aber, wir haben eine schöne, ziemlich vollständige Theorie für RK-Kollokationsverfahren.

5.3 Implementierung impliziter RK-Verfahren

Ein implizites RK-Verfahren mit s Stufen und Butcher-Tableau $\begin{array}{c|c} c & A \\ \hline & b \end{array}$ lautet

$$\begin{cases} k_i &= f\left(t_n + c_i h_n, x_n + h_n \sum_{j=1}^{s} a_{i,j} k_j\right), \quad i = 1, \ldots, s \\ x_{n+1} &= x_n + h_n \sum_{i=1}^{s} b_i k_i \end{cases}$$

Wir haben ein sd-dimensionales System implizierter (meistens nichtlinearer) Gleichungen für die s Unbekannten k_1, ..., k_s, falls die Differentialgleichung (d.h. die Abbildung f) d-dimensional ist. Wir können dieses System durch sukzessive Iterationen oder (besser!) die Newton'sche Methode approximativ lösen. Dafür ist die folgende äquivalente Darstellung des impliziten RK-Verfahrens günstiger:

$$\begin{cases} y_i &= x_n + h_n \sum_{j=1}^{s} a_{i,j} f\left(t_n + c_j h_n, y_j\right), \qquad i = 1, \ldots, s, \\ x_{n+1} &= x_n + h_n \sum_{i=1}^{s} b_i f(t_n + c_i h_n, y_i) \end{cases}$$

Definiere $Y = \begin{pmatrix} y_1 \\ \vdots \\ y_s \end{pmatrix} \in \mathbb{R}^{sd}$ und $F : \mathbb{R}^{sd} \to \mathbb{R}^{sd}$

durch

$$F(Y) = \begin{pmatrix} F_1(Y) \\ \vdots \\ F_s(Y) \end{pmatrix}$$

wobei $F_i(Y) := x_n + h_n \sum_{j=1}^{s} a_{i,j} f(t_n + c_j h_n, y_j)$

Wir müssen den eindeutigen Fixpunkt

$$\overline{Y} = F(\overline{Y})$$

der sd-dimensionalen Abbildung F berechnen oder äquivalent die Nullstelle $G(\overline{Y}) = 0$ der sd-dimensionalen Abbildung

$$G(Y) = Y - F(Y)$$

finden.

Dafür können wir die vektorwertige Version der Newton'schen Methode benutzen

$$\boxed{Y^{(\nu+1)} = Y^{(\nu)} - \nabla G(Y^{(\nu)})^{-1} G(Y^{(\nu)})} \qquad \nu = 0, 1, 2, \ldots$$

mit z.B. $Y^{(0)} = \begin{pmatrix} x_n \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^{sd}$, wobei $\nabla G(Y) = \left[\frac{\partial G_i}{\partial Y_j}(Y)\right]$, eine $sd \times sd$-Matrix mit $d \times d$-Blöcke

$$\frac{\partial G_i}{\partial Y_j}(Y) = \delta_{i,j}\, I_d - h_n\, a_{i,j}\, \frac{\partial f}{\partial x}\,(t_n + c_j h_n, y_j) \quad i, j = 1, \ldots, s,$$

mit $\delta_{i,j}$ = Kronecker-Delta-Symbol, $I_d = d \times d$-Identitätsmatrix, und

$$\frac{\partial f}{\partial x}\,(t, x) = \left[\frac{\partial f_p}{\partial x_q}\,(t, x)\right], \qquad d \times d - \text{Jacobi-Matrix von } f \text{ (bzg. } x)$$

In der Praxis lösen wir das lineare Gleichungssystem mit sd Gleichungen

$$\boxed{\nabla G(Y^{(\nu)})\ \delta^{(\nu)} = -G(Y^{(\nu)})} \qquad \nu = 0, 1, 2, \ldots$$

für $\delta^{(\nu)} \in \mathbb{R}^{sd}$, z.B. mit einer LR-Zerlegung

$$\Rightarrow \quad \text{dann gilt} \quad \boxed{Y^{(\nu+1)} = Y^{(\nu)} + \delta^{(r)}}$$

ABER dies ist sehr aufwendig!

Wir müssen

(1) die Jacobi-Matrix $\frac{\partial f}{\partial x} = \left[\frac{\partial f_p}{\partial x_q}\right]$ als Funktion finden:

- nicht einfach, falls $d \gg 1$ ist,
- ein Grund, um einen RK-Verfahren statt einem Taylor-Verfahren zu benutzen, war: RK-Verfahren brauchen keine Ableitungen von f!

(2) die s Jacobi-Matrizen

$$J_i = \frac{\partial f}{\partial x}\ (t_n + c_i h_n, Y_i)$$

auswerten.

Wir können diesen Rechneraufwand mit Tricks verringern, z.B.

(i) Benutze J_1 für alle J_i, $i = 1, \ldots, s$

(ii) benutze dasselbe J (z.B. $= J_1$) für mehrere Zeitschritte.

Wie genau sind die Ergebnisse?

Gute Nachricht: mit dem Anfangswert $Y^{(0)} = \begin{pmatrix} x_n \\ x_n \\ \vdots \\ x_n \end{pmatrix}$ konvergiert die Newton'sche Methode sehr schnell.

$\Rightarrow$ wir brauchen im Allgemeinen nur ein Paar Iterationen!

ABER für $d \gg 1$ ist der Rechneraufwand noch zu groß; insbesondere, falls fast alle $a_{i,j}$ nicht gleich null sind.

5.3.1 DIRK-Verfahren

Die DIRK-Verfahren versuchen diesen Aufwand zu verringern.

DIRK	=	Diagonal-Implizit-Runge-Kutta

Hier lautet

$$a_{i,j} = \begin{cases} 0 & \text{falls} \quad j > i \\ \alpha & \text{falls} \quad j = i \\ \text{irgendwas} & \text{falls} \quad j < i \end{cases}$$

Beispiele

(1)
$$\begin{array}{c|cc} \alpha & \alpha & 0 \\ 1-\alpha & 1-2\alpha & \alpha \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \qquad \alpha = \frac{3+\sqrt{3}}{6}, \quad (s=2,\ p=3)$$

(2)
$$\begin{array}{c|cc} \alpha & \alpha & 0 \\ 1 & 1-\alpha & \alpha \\ \hline & 1-\alpha & \alpha \end{array} \qquad \alpha = \frac{2-\sqrt{2}}{2}, \quad (s=2,\ p=2)$$

Hier lauten die (umgeschriebenen) Stufengleichungen

$$\begin{cases} Y_1 - \alpha h_n f(t_n + c_1 h_n, Y_1) &= x_n \\ Y_i - \alpha h_n f(t_n + c_i h_n, Y_i) &= x_n + h_n \sum_{j=1}^{i-1} a_{ij} f(t_n + c_j h_n, Y_j) \\ & \qquad\qquad i = 2, \ldots, s \end{cases}$$

wobei die Y_j in der i-ten Summe schon aus den vorherigen Komponenten bekannt sind.

Die Struktur des Gleichungssystems ist jetzt viel einfacher – schneller lösbar.

ABER

SATZ Ein DIRK-Verfahren mit s Stufen hat Ordnung $p \leq s+1$

Beweis (Hinweis)

Betrachte die rationale Funktion

$$R(z) = \frac{P(z)}{Q(z)} = 1 + zb(I - zA)^{-1}\ \mathbf{1}.$$

Hier ist $Q(z) = \det(I - zA) = (1 - \alpha z)^s$.

Frage *Warum sollen wir ein implizites Runge-Kutta-Verfahren benutzen, wenn wir so viele Schwierigkeiten überwinden müssen?*

5.3.2 Numerische Instabilität und A–Stabilität

Wir betrachten eine Klasse komplexwertiger „Test"-Differentialgleichungen

$$\frac{dz}{dt} = \lambda z$$

wobei $\lambda = \alpha + \imath\beta \in \mathbb{C}$ und $z = x + \imath y \in \mathbb{C}$, wobei $\imath = \sqrt{-1}$.

Die Lösung mit Anfangswert $z(0) = z_0$ lautet

$$z(t) = e^{\lambda t} z_0 = e^{\alpha t}\ e^{\imath\beta t} z_0$$

Dann gilt

$$\begin{aligned} |z(t)| &= e^{\alpha t}|z_0| \qquad \text{weil } |e^{\imath\beta t}| = 1 \\ &\to 0 \text{ für } t \to \infty \ \ \forall z_0 \end{aligned}$$

genau dann, wenn

$$\alpha = \mathrm{Re}(\lambda) < 0.$$

In diesem Fall sagen wir, dass die Nulllösung $z(t) \equiv 0$ asymptotisch stabil ist.

Bemerkung: wir können die obige DGL in ein reellwertiges System von DGLen umschreiben. Sei $z(t) = x(t) + \imath y(t)$. Dann gilt

$$\frac{d}{dt}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} \alpha & -\beta \\ \beta & \alpha \end{bmatrix}\begin{pmatrix} x \\ y \end{pmatrix}$$

Betrachte jetzt ein RK-Verfahren mit Butcher-Tableau $\begin{array}{c|c} c & A \\ \hline & b \end{array}$. Für die Test-Funktion $f(z) = \lambda z$ und konstante Schrittweiten $h_n \equiv h$ haben wir

$$\begin{cases} k_i &= \lambda\left(z_n + h\ \sum_{j=1}^{s}\ a_{i,j}k_j\right), \qquad i = 1, \ldots, s \\ z_{n+1} &= z_n + h\ \sum_{j=1}^{s}\ b_j k_j \end{cases}$$

Schreibe $K = \begin{pmatrix} k_1 \\ \vdots \\ k_s \end{pmatrix} \in \mathbb{C}^s$, $\mathbf{1} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} 1 + \imath 0 \\ \vdots \\ 1 + \imath 0 \end{pmatrix} \in \mathbb{C}^s$.

Dann gilt

$$\begin{cases} K &= \lambda z_n \mathbf{1} + \lambda h A K \\ z_{n+1} &= z_n + hbK \end{cases}$$

($b = (b_1, \ldots, b_s)$ ist hier ein Zeilenvektor)

$$\begin{cases} K &= \lambda z_n (I - \lambda h A)^{-1}\mathbf{1} \\ z_{n+1} &= z_n + h\lambda z_n b(I - \lambda h A)^{-1}\mathbf{1} \end{cases}$$

oder

$$\boxed{z_{n+1} = R(h\lambda)z_n}$$

wobei

$$\boxed{R(z) = 1 + zb(I - zA)^{-1}\mathbf{1}}$$

eine komplexwertige Abbildung $R:\ \mathbb{C} \to \mathbb{C}$ ist.

Insbesondere gilt

$$|z_n| = |R(h\lambda)|^n\ |z_0| \to 0 \quad \text{für} \quad n \to \infty$$

genau dann, wenn $|R(h\lambda)| < 1$ ist.

d.h. die Nulllösung $z_n \equiv 0 + \imath 0 \in \mathbb{C}$ des RK-Verfahrens ist auch asymptotisch stabil genau dann, wenn die Schrittweite $h > 0$ der Ungleichung

$$|R(h\lambda)| < 1$$

genügt.

Definition *Die Menge*

$$S_R := \{z \in \mathbb{C} : \ |R(z)| < 1\}$$

heißt Stabilitätsgebiet des RK-Verfahrens mit Abbildung R.

Definition *Ein RK-Verfahren mit Abbildung R heißt A-stabil, falls*

$$\mathbb{C}^- \subset S_R,$$

wobei $\mathbb{C}^- = \{z \in \mathbb{C} : \ Re(z) < 0\}$.

Beispiele

(1) das explizite Euler-Verfahren $\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$

$$R(z) = 1 + z1(1 - z0)^{-1}1 = 1 + z$$

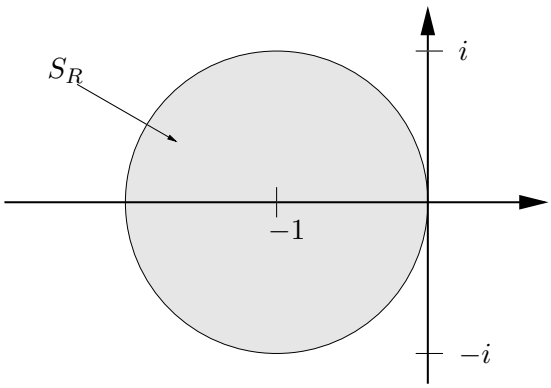
$|R(z)| = |1 + z| < 1$ ist die Innere des Kreises mit und Zentrum $z = -1 + \imath 0$ und Radius $r = 1$.

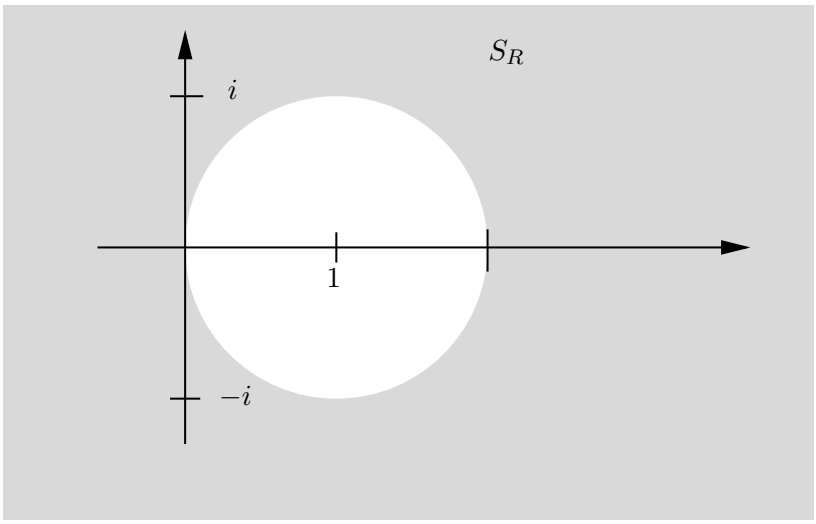
$\Rightarrow$ das explizite Euler-Verfahren ist nicht A-stabil.

(2) das implizite Euler-Verfahren $\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$

$$R(z) = 1 + z1(1 - z1)^{-1}1 = 1 + z(1 - z)^{-1} = \frac{1}{1 - z}$$

$$|R(z)| < 1 \ \Leftrightarrow \ |1 - z| > 1$$


S_R
i
-1
$-i$


S_R
i
1
$-i$

S_R = ist die Externe (d.h., Aussengebiet) des Kreises mit Zentrum $1 + \imath 0$ und Radius 1

$\mathbb{C}^- \subset S_R$ hier

$\Rightarrow$ das implizite Euler-Verfahren ist A-stabil.

Wir werden jetzt den folgenden Satz beweisen.

SATZ (Deuflhard/Bornemann, Satz 30, Seite 230)

$R(z) = 1 + zb(I - zA)^{-1}\mathbf{1}$ ist eine rationale Funktion mit

$$R(z) = \frac{P_s(z)}{Q_s(z)}$$

wobei $P_s, Q_s \in \mathcal{P}_s$ (Polynome von Grad $\leq s$) mit $P_s(0) = Q_s(0) = 1$ sind.

Beweis $R(z) = 1 + zbK$ wobei K die Lösung von

$$(I - zA)K = \mathbf{1}$$

ist, d.h. wegen der Cramer'schen Regel gilt

$$K = \frac{\mathrm{Adj}(I - zA)\mathbf{1}}{\det(I - zA)}$$

$$R(z) = \frac{\det(I - zA) + zb\mathrm{Adj}(I - zA)\mathbf{1}}{\det(I - zA)}$$

Korollar $|R(z)| < 1 \quad \Leftrightarrow \quad |P_s(z)| < |Q_s(z)|$

SATZ Kein explizites RK-Verfahren ist A-stabil.

Beweis Betrachte ein explizites RK-Verfahren mit Butcher-Tableau $\begin{array}{c|c} c & A \\ \hline & b \end{array}$, d.h.,

$$a_{i,j} = 0, \qquad \forall j \geq i$$

$$\Rightarrow \quad \det(I - zA) = \det \begin{bmatrix} 1 & 0 & \dots & 0 \\ -za_{2,1} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ -za_{n,1} & -za_{n,2} & \dots & 1 \end{bmatrix} = 1$$

d.h. $Q_s(z) \equiv 1$ ist. Aber dann ist

$$|R(z)| = |P_s(z)| < 1, \quad \forall z \in \mathbf{C}^-$$

nicht möglich, weil $P_s(z)$ ein Polynom von Grad ≥ 1 ist (d.h. $P_s(z)$ ist keine konstante Funktion).

Für ein implizites RK-Verfahren haben wir immer

$$\text{Grad } (Q_s) \geq 1$$

Dann statt $|P_s(z)| < |Q_s(z)|$ für jedes $z \in \mathbb{C}^-$ zu prüfen, können wir das folgende Ergebnis benutzen.

SATZ (Iserles, Lemma 4.3, Seite 61)

$|R(z)| < 1$ für alle $z \in \mathbb{C}^-$ genau dann, wenn

(1) $\text{Re}(z_{pol}) > 0$ für jeden Pol z_{pol} von R, d.h. mit $|R(z_{pol})| = \infty$
und
(2) $|R(\imath t)| \leq 1$ für alle $t \in \mathbb{R}$

Beweis

($\Rightarrow$ Richtung): fast trivial

($\Leftarrow$ Richtung): d.h. (1) + (2) gelten

(1) $\Rightarrow$ R hat keine Pole in $\mathbb{C}^-$ $\Rightarrow$ R ist analytisch auf $\overline{\mathbb{C}^-} = \mathbb{C}^- \cup \{0 + \imath\mathbb{R}\}$

Aber R ist keine konstante Funktion.

Wegen des Maximumprinzips liegt das Maximum von R auf dem Rand von $\overline{\mathbb{C}^-}$, d.h. auf der imaginären Achse

$$|R(\imath t)| \leq 1, \quad \forall t \in \mathbb{R}, \qquad \Rightarrow \qquad |R(z)| < 1, \quad \forall z \in \mathbb{C}^-$$

Beispiel

RK-Gauß-Verfahren $(s = 2, p = 2s = 4)$

$$\begin{array}{c|cc} \frac{3-\sqrt{3}}{6} & \frac{1}{4} & \frac{3-2\sqrt{3}}{12} \\ \frac{3+\sqrt{3}}{6} & \frac{3+2\sqrt{3}}{12} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

Hier gilt

$$R(z) = \frac{1 + \frac{1}{2}\, z + \frac{1}{12}\, z^2}{1 - \frac{1}{2}\, z + \frac{1}{12}\, z^2}$$

(1) Die Pole von R sind $z_{pol} = 3 \pm i\sqrt{3}$

$$\Rightarrow \quad \mathrm{Re}(z_{pol}) > 0$$

(2) Die Funktion $t \to R(\imath t)$ ist gegeben durch

$$\begin{aligned} R(\imath t) &= \frac{1 + \frac{1}{2}\, \imath t - \frac{1}{12}\, t^2}{1 - \frac{1}{2}\, \imath t - \frac{1}{12}\, t^2} \\ &= \frac{(1 - \frac{1}{12}\, t^2) + \imath(\frac{1}{2}\, t)}{(1 - \frac{1}{12}\, t^2) - \imath(\frac{1}{2}\, t)} \end{aligned}$$

$$\Rightarrow \quad |R(\imath t)| \equiv 1 \quad \forall t \in \mathbb{R}$$

Von dem obigen Satz $\Rightarrow$ das RK-Gauß-Verfahren mit $s = 2$ ist A-stabil.

Im Allgemeinen sind alle RK-Gauß-Verfahren A-stabil, d.h. für alle Stufenzahlen $s \geq 1$. (Aber der Beweis ist nicht einfach!)

Bemerkung A-Stabilität ist ein nützlicher Begriff. Aber sie ist nicht das letzte Wort über die numerische Stabilität. Es gibt auch viele andere nützliche Stabilitätsbegriffe.

Kapitel 6

Mehrschrittverfahren

Literatur Schwarz: Kap. 9.2, Stummel/Hainer: Kap. 12

Wir betrachten eine Anfangswertaufgabe (AWA)

$$\frac{dx}{dt} = f(t,x), \quad x(t_0) = x_0.$$

Um ein 1-Schrittverfahren

$$x_{n+1} = x_n + h_n \Phi(h_n, t_n, x_n, x_{n+1})$$

herzuleiten, haben wir entweder die Ableitung $x'(t)$ oder die Integrandfunktion $f(t,x(t))$ in der Integralgleichungsdarstellung der AWA, d.h.

$$x(t_{n+1}) = x(t_n) + \int_{t_n}^{t_{n+1}} f(t,x(t))dt,$$

auf dem Teilintervall $[t_n, t_{n+1}]$ approximiert mit der aktuellen Information, d.h. x_n und (noch unbekannt) x_{n+1}.

Mehrschrittverfahren verwenden die vorhandene Information auch an vorhergehenden Stützstellen $t_{n-1}, \ldots, t_{n-m}$, d.h. die vorhergehenden Berechnungen $x_{n-1}, \ldots, x_{n-m}$. z.B.

(1) Ableitungsapproximation $\Rightarrow$ BDF-Verfahren

BDF $\equiv$ Backwards Difference Formula (Rückwartsdifferenzenformel).

(2) Integralapproximation

$$\Rightarrow \begin{cases} \text{Adams-Bashford-Verfahren} & \text{(explizit)} \\ \text{Adams-Moulton-Verfahren} & \text{(implizit)} \end{cases}$$

6.1 Adams-Bashford-Verfahren

Wir werden stets voraussetzen, dass die Stützstellen $t_0, t_1, t_2, \ldots$ äquidistant sind, d.h. $t_j = t_0 + jh$ (h = Schrittweite), und werden die Abkürzung

$$f_j = f(t_j, x_j)$$

benutzen.

Das Interpolationspolynom für die Daten

$$(t_{n-m}, f_{n-m}), \ldots, (t_n, f_n)$$

lautet

$$\sum_{j=o}^{m} f_{n-m+j} L_{n;m,j}(t)$$

mit den Lagrange'schen Interpolationspolynomen m-ten Grades

$$L_{n;m,j}(t) = \prod_{\substack{i=0 \\ i \neq j}}^{m} \frac{t - t_{n-m+i}}{t_{n-m+j} - t_{n-m+i}}$$

Vorher haben wir ein solches Interpolationspolynom nur auf dem Definitionsintervall $[t_{n-m}, t_n]$ benutzt, aber jetzt werden wir es auf dem nächstfolgenden Teilintervall $[t_n, t_{n+1}]$ verwenden als eine Approximation – tatsächlich eine Extrapolation – für $f(t, x(t))$

$$x(t_{n+1}) = x(t_n) + \int_{t_n}^{t_{n+1}} f(t, x(t)) dt$$

$\Rightarrow$ **$(m+1)$-Schritt-Adams-Bashford-Verfahren**

$$\begin{aligned} x_{n+1} &= x_n + \int_{t_n}^{t_{n+1}} \sum_{j=0}^{m} f_{n-m+j} L_{n;m,j}(t) dt \\ &= x_n + \sum_{j=0}^{m} \left[\int_{t_n}^{t_{n+1}} L_{n;m,j}(t) dt \right] f_{n-m+j} \end{aligned}$$

d.h.

$$\boxed{x_{n+1} = x_n + h \sum_{j=0}^{m} \beta_j^{(m)} f_{n-m+j}}$$

wobei

$$\boxed{\beta_j^{(m)} = \frac{1}{h} \int_{t_n}^{t_{n+1}} L_{n;m,j}(t) dt, \quad j = 0, 1, \ldots, m}$$

d.h.

$$\begin{aligned} \beta_j^{(m)} &= \frac{1}{h} \int_{t_n}^{t_{n+1}} \prod_{\substack{i=0 \\ i \neq j}}^{m} \frac{t - t_{n-m+i}}{t_{n-m+j} - t_{n-m+i}} \, dt \\ &= \frac{1}{h} \int_{0}^{1} \prod_{\substack{i=0 \\ i \neq j}}^{m} \frac{(sh + t_0 + nh) - (t_0 + (n-m+i)h)}{(t_0 + (n-m+j)h) - (t_0 + (n-m+i)h)} \, hds \end{aligned}$$

mit $t = sh + t_n = sh + nh \Rightarrow dt = hds$

und $t_k = t_0 + kh$ usw.

$$= \int_{0}^{1} \prod_{\substack{i=0 \\ i \neq j}}^{m} \frac{s + m - i}{j - i} \, ds$$

NB die $\beta_j^{(m)}$ hängen nicht von n, h ab.

z.B. **4-Schritt-Adams-Bashford-Verfahren**

$$x_{n+1} = x_n + \frac{h}{24} \left(-9 f_{n-3} + 37 f_{n-2} - 59 f_{n-1} + 55 f_n \right)$$

Wie ist die Ordnung eines solchen Verfahren. Betrachte den lokalen Diskretisierungsfehlers des obigen 4-Schritt-Verfahrens.

$$x(t_{n+1}) - x(t_n) - \frac{h}{24}\begin{bmatrix} -9f(t_{n-3}, x(t_{n-3})) + 37f(t_{n-2}, x(t_{n-2})) \\ -59f(t_{n-1}, x(t_{n-1})) + 55f(t_n, x(t_n)) \end{bmatrix}$$

$$= x(t_{n+1}) - x(t_n) - \frac{h}{24}\Big[-9x'(t_{n-2}) + 37x'(t_{n-2}) - 59x'(t_{n-1}) + 55x'(t_n)\Big]$$

$$= \quad \text{(durch eine Taylor-Entwicklungen um } \ t = t_n)$$

$$= hx' + \frac{1}{2}h^2x'' + \frac{1}{6}h^3x''' + \frac{1}{24}h^4x^{(4)} + \frac{1}{120}h^5x^{(5)} + O(h^6)$$

$$+\frac{9h}{24}\left(x' - 3hx'' + \frac{9}{2}h^2x''' - \frac{9}{2}h^3x^{(4)} + \frac{27}{8}h^4x^{(5)} + O(h^5)\right)$$

$$-\frac{37h}{24}\left(x' - 2hx'' + 2h^2x''' - \frac{4}{3}h^3x^{(4)} + \frac{2}{3}h^4x^{(5)} + O(h^5)\right)$$

$$-\frac{59h}{24}\left(x' - hx'' + \frac{1}{2}h^2x''' - \frac{1}{6}h^3x^{(4)} + \frac{1}{24}h^4x^{(5)} + O(h^5)\right)$$

$$-\frac{55h}{24}x'$$

$$= \frac{251}{720}h^5x^{(5)}(t_n) + O(h^6) = O(h^5)$$

Wie im Einschrittverfahrensfall, verlieren wir hier auch eine Potenz zwischen dem lokalen und globalen Diskretisierungsfehler.

$\Rightarrow$ Das 4-Schritt-Adams-Bashford-Verfahren hat 4.te Ordnung!

Im Allgemeinen: Ein M-Schritt-Adams-Bashford-Verfahren hat Ordnung $p = M$ $(= m + 1$ oben).

6.2 Adams-Moulton-Verfahren

Jetzt benutzen wir auch die Information

$$t_{n+1}, f_{n+1} = f(t_{n+1}, x_{n+1}),$$

wobei x_{n+1} noch unbekannt ist, in dem Interpolationspolynom sowie die Daten (t_{n-m}, f_{n-m}), ..., (t_n, f_n).

$\Rightarrow$ $(m+1)$ **Schritt-Adams-Moulton-Verfahren**

$$x_{n+1} = x_n + \sum_{j=0}^{m+1} f_{n-m+1} \int_{t_n}^{t_{n+1}} L_{n+1;m+i,j}(t)dt$$

d.h.

$$\boxed{x_{n+1} = x_n + h \sum_{j=0}^{m+1} \gamma_j^{(m)} f_{n-m+j}}$$

wobei

$$\boxed{\gamma_j^{(m)} = \frac{1}{h} \int_{t_n}^{t_{n+1}} L_{n+1;m+1,j}(t)dt}$$

hängt nicht von n oder h ab.

Beispiel: 4-Schritt-Adams-Moulton-Verfahren

$$x_{n+1} = x_n + \frac{h}{720} \left(-19 f_{n-3} + 106 f_{n-2} + 264 f_{n-1}\right)$$

Prädiktor-Korrektor-Verfahren sind explizite Verfahren, die das "x_{n+1}" an der rechten Seite eines M- schritt-Adams-Moulton-Verfahren durch das x_{n+1} eines M-Schritt-Adams-Bashford-Verfahren ersetzen.

Vergleich: Wie wir das Heun-Verfahren hergeleitet haben – mit dem Euler für x_{n+1} in dem impliziten Trapez-Verfahren.

6.3 BDF-Verfahren

Jetzt approximieren wir die Lösung $x(t)$ statt der Funktion $f(t, x(t))$ durch ein Interpolationspolynom, um eine Approximation der Ableitung $x'(t_{n+1})$ in

$$x'(t_{n+1}) = f(t_{n+1}, x(t_{n+1}))$$

zu finden.

Dafür ist die Newton'sche Darstellung des Interpolationspolynoms mit Daten

$$(t_{n-m}, x_{n-m}), \ldots, (t_{n+1}, x_{n+1})$$

Es ist günstig zu definieren :

$$X_{n+1}(t) = x_{n+1} + \sum_{j=1}^{m+1} \frac{1}{j!h^j} \nabla^j x_{n+1}(t - t_{n+1}) \dots (t - t_{n+1-j})$$

Dann gilt

$$X'_{n+1}(t_{n+1}) = \frac{1}{h} \sum_{j=1}^{m+1} \frac{1}{j} \nabla^j x_{n+1}$$

Hier ist ∇ der Rückwärtsdifferenzenoperator

$$\nabla x_{n+1} = x_{n+1} - x_n, \qquad \nabla^2 x_{n+1} = \nabla x_{n+1} - \nabla x_n \text{ usw.}$$

Das $(m+1)$**-Schritt-BDF-Verfahren** lautet

$$\boxed{\tfrac{1}{2} \sum_{j=1}^{m+1} \tfrac{1}{j} \nabla^j x_{n+1} = f(t_{n+1}, x_{n+1}),}$$

das wir umschreiben können als

$$\sum_{j=0}^{m+1} \alpha_j^{(m)} x_{n+1-j} = h\beta_0^{(m)} f(t_{n+1}, x_{n+1}),$$

mit $\beta_0^{(m)}$ gewählt, so dass $\alpha_0^{(m)} = 1$. z.B.

(1) $m = 0$ $\Rightarrow$ 1-Schrittverfahren

$$x_{n+1} - x_n = hf(t_{n+1}, x_{n+1})$$

(tatsächlich, das implizites Euler-Verfahren!)

(2) $m = 1$ $\Rightarrow$ 2-Schrittverfahren

$$\frac{1}{2}\left((x_{n+1} - x_n) - (x_n - x_{n-1})\right) + \frac{1}{1}\left(x_{n+1} - x_n\right) = hf(t_{n+1}, x_{n+1})$$

d.h.

$$x_{n+1} - \frac{4}{3}x_n + \frac{1}{3}x_{n-1} = \frac{2}{3}hf(t_{n+1}, x_{n+1})$$

Die BDF-Verfahren sind alle implizit (wegen der Konstruktion).

6.4 Allgemeine lineare Mehrschrittverfahren

Die Adams-/BDF-Verfahren sind Beispiele der Familie der linearer Mehrschrittverfahren mit der allgemeinen M-Schritt-Form

$$\boxed{\sum_{j=0}^{M} a_{M-j} x_{n+1} - j = h \sum_{j=0}^{M} b_{M-j} f_{n+1-j}}$$

wobei $a_M = 1$ ist.

Sie heißen "linear" wegen der linearen Kombination der f_j an der rechten Seite – die DGL, d.h. die Funktion $f(t, x)$, muß nicht linear in x sein.

Bemerkungen:

(1) $b_m = 0 \quad \Leftrightarrow \quad$ explizites Verfahren
$b_m \neq 0 \quad \Leftrightarrow \quad$ implizites Verfahren

(2) Um eine Berechnung anzufangen brauchen wir $x_0, \ldots, x_M$. Aber nur x_0 ist gegeben.
Wir können $x_1, \ldots, x_M$ mit einem 1-Schrittverfahren berechnen. Dieses Startverfahren soll die gleiche Ordnung, oder höhere Ordnung, wie das Mehrschrittverfahren haben, um die Ordnung des Gesamtverfahrens zu erhalten.

(3) Konsistenz ist wie im 1-Schrittfall definiert, d.h. mit

$$L_n(h)/h \to 0 \quad \text{als} \quad h \to 0$$

für den lokalen Diskretisierungsfehler $L_n(h)$

Eine äquivalente Bedingung lautet

$$\boxed{\rho(1) = 0 \quad \text{und} \quad \rho'(1) = \sigma(1)}$$

wobei $\rho(z) = \sum_{j=0}^{M} a_j z^j$ und $\sigma(z) = \sum_{j=0}^{M} b_j z^j$.

Der Beweis folgt durch Taylor-Entwicklungen usw. Die Notwendigkeit dieser Bedingung folgt aus der Tatsache. dass das Verfahren exact sein soll für die einfachen Verfahren

(i) $x' \equiv 0 \quad \Rightarrow \quad \rho(1) = 0$

(ii) $x' \equiv 1 \quad \Rightarrow \quad \rho'(1) = \sigma(1)$

Aber der Satz für 1-Schrittverfahren

„Konsistenz $\Leftrightarrow$ Konvergenz“

ist im Allgemeinen falsch für Mehrschrittverfahren.

Gegenbeispiel: Betrachte das MS-Verfahren

$$x_{n+1} + 4x_n - 5x_{n-1} = 4hf_n + 2hf_{n-1}$$

$$\Rightarrow \rho(z) = z^2 + 42 - 5 = (z-1)(2+5), \qquad \sigma(z) = 4z + 2$$

$$\Rightarrow \rho'(1) = \sigma(1) = 6, \qquad \rho(1) = 0$$

Das Verfahren ist konsistent. Betrachte jetzt die Anfangswertaufgabe

$$x' \equiv 0, \quad x(t_0) = x_0$$

mit Lösung $x(t) \equiv x_0$. Das Verfahren hier lautet

$$x_{n+1} + 4x_n - 5x_{n-1} = 0$$

und $x_n \equiv x_0$ ist eine Lösung.

Die lineare Differenzengleichung

$$x_{n+1} + 4x_n - 5x_{n-1} = 0$$

hat die allgemeine Lösung

$$x_n = A + B(-5)^n$$

Für $x_0 \equiv 0$ und $x_1 = \varepsilon$ (Abrundungsfehler), z.B. haben wir

$$x_n = \frac{\varepsilon}{6}\,(1 - (-5)^n) \to \pm\infty \ \text{ alternierend } \ \text{für } \ n \to \infty$$

d.h. wir können keine Konvergenz hier erwarten.

Die Schwierigkeit hier ist wegen der Tatsache, dass $z = -5$ eine Nullstelle des Polynoms $\rho(2)$ ist.

Wir sagen, dass ein Mehrschrittverfahren streng stabil ist, falls die Nullstellen von ρ nicht gleich 1 der Bedingung

$$|z| < 1$$

genügen. $\Rightarrow$ alle Fehler wie oben sterben aus.

z.B. das M-Schritt-Adams-Bashford-Verfahren hat

$$\rho(z) = z^M - z^{m-1} = z^{m-1}(z-1)$$

mit Nullstellen $z = 1$ und $z = 0$. Es ist streng stabil.

Der entsprechende Satz hier lautet

Konsistenz + strenge Stabilität $\Leftrightarrow$ Konvergenz.

Kapitel 7

Shooting method for BVP for ODE

Consider the following boundary value problem for a second order scalar ODE

$$\frac{d^2y}{dx^2} = f\left(x, y, \frac{dy}{dx}\right) \tag{7.1}$$

for $a \leq x \leq b$ with

$$y(a) = \alpha, \qquad y(b) = \beta. \tag{7.2}$$

before we consider numerical methods for finding solutions, we must be sure that the BVP (7.1)–(7.2) does indeed have a solution. The following theorem provides sufficient conditions for the existence of a unique solution. Such a soution may, of course, exist under much weaker conditions. In the theorem we denote the variable in the function f by (x, y, z) and later we replace z by $\frac{dy}{dx}$ or, more briefly, by y'.

Theorem *Suppose that f and its partial derivatives $\frac{\partial f}{\partial y}$ and $\frac{\partial f}{\partial z}$ are continuous on the set $\mathcal{D} := [a, b] \times \mathbb{R}^2$ and that*

A) $\dfrac{\partial f}{\partial y}(x, y, z) > 0$ *for all* $(x, y, z) \in \mathcal{D}$*;*

B) there exists a constant $M > 0$ such that

$$\left|\frac{\partial f}{\partial z}(x, y, z)\right| \leq M \quad \textit{for all} \quad (x, y, z) \in \mathcal{D}.$$

Then the BVP (7.1)–(7.2) has a unique solution.

As an example consider the BVP for the linear ODE

$$\frac{d^2y}{dx^2} = y, \qquad y(0) = y(1) = 0.$$

The ODE has the explicit general solution $y(x) = Ae^x + Be^{-x}$ and the BVP has the unique solution $y(x) \equiv 0$. The Theorem applies to this example, i.e. with

$$f(x, y, z) = y, \qquad \frac{\partial f}{\partial y} \equiv 1, \qquad \frac{\partial f}{\partial z} \equiv 0.$$

On the other hand the BVP

$$\frac{d^2y}{dx^2} = -y, \qquad y(0) = y(1) = 0.$$

The Theorem does not apply to this example since

$$f(x, y, z) = -y, \qquad \frac{\partial f}{\partial y} \equiv -1 < 0, \qquad \frac{\partial f}{\partial z} \equiv 0.$$

7.1 Linear ODE

When the function f in the ODE has the special form

$$f(x, y, z) = p(x)z + q(x)y + r(x)$$

then the second order ODE is linear with respect to the unknowns $y(x)$ and $\frac{dy}{dx}(x)$, i.e. has the form

$$\frac{d^2y}{dx^2} = p(x)\frac{dy}{dx} + q(x)y + r(x). \tag{7.3}$$

In this case, if

A1) $p(x)$, $q(x)$ and $r(x)$ are continuous on $[a, b]$,
B1) $q(x) > 0$ on $[a, b]$

then the conditions of the above theorem hold and the BVP for the linear ODE (7.2)–(7.3) has a unique solution.

Suppose that $y_1(x)$ is the solution of the initial value problem

$$\frac{d^2y}{dx^2} = p(x)\frac{dy}{dx} + q(x)y + r(x), \qquad y(a) = \alpha, \quad \frac{dy}{dx}(a) = 0, \tag{7.4}$$

and that $y_2(x)$ is the solution of the initial value problem

$$\frac{d^2y}{dx^2} = p(x)\frac{dy}{dx} + q(x)y, \qquad y(a) = 0, \quad \frac{dy}{dx}(a) = 1 \tag{7.5}$$

on the interval $a \leq x \leq b$. Such solutions exist and are unique by the above assumptions A1) and B1).

Then the function $y(x)$ on the interval $a \leq x \leq b$ defined by

$$y(x) := y_1(x) + \frac{\beta - y_1(b)}{y_2(b)}\, y_2(x), \qquad a \leq x \leq b, \tag{7.6}$$

is the unique solution of the BVP for the linear ODE (7.2)–(7.3) provided $y_2(b) \neq 0$.

Note we can exclude the case $y_2(b) = 0$. It never arises, since the unique solution of the BVP

$$\frac{d^2y}{dx^2} = p(x)\frac{dy}{dx} + q(x)y, \qquad y(a) = 0, \quad y(b) = 0,$$

is $y(x) \equiv 0$, so there is no solution $y_2(x)$ to the IVP (7.5) with $y_2(b) = 0$.

Let us now show that $y(x)$ defined by (7.6) is a solution of the BVP (7.2)–(7.3). First consider the boundary conditions.

$$\begin{aligned}
\mathbf{x = a} \qquad y(a) &= y_1(a) + \frac{\beta - y_1(b)}{y_2(b)} \cdot y_2(a) \\
&= \alpha + \frac{\beta - y_1(b)}{y_2(b)} \cdot 0 = \alpha, \\
\mathbf{x = b} \qquad y(b) &= y_1(b) + \frac{\beta - y_1(b)}{y_2(b)} \cdot y_2(b) \\
&= y_1(b) + \beta - y_1(b) = \beta.
\end{aligned}$$

And now note that

$$\begin{aligned}
\frac{d^2y}{dx^2} &= \frac{d^2y_1}{dx^2} + \frac{\beta - y_1(b)}{y_2(b)}\frac{d^2y_2}{dx^2} \\
&= \left[p(x)\frac{dy_1}{dx} + q(x)y_1 + r(x)\right] + \frac{\beta - y_1(b)}{y_2(b)}\left[p(x)\frac{dy_2}{dx} + q(x)y_2\right]
\end{aligned}$$

$$= p(x)\left[\frac{dy_1}{dx} + \frac{\beta - y_1(b)}{y_2(b)}\frac{dy_2}{dx}\right] + q(x)\left[y_1 + \frac{\beta - y_1(b)}{y_2(b)}y_2\right] + r(x)$$

$$= p(x)\frac{dy}{dx} + q(x)y + r(x),$$

so the function $y(x)$ defined by (7.6) satisfies the linear ODE (7.3).

Thus to solve the BVP (7.2)–(7.3) for the linear ODE, we can apply, say, a Runge-Kutta scheme to each of the initial value problems (7.4) and (7.5) and use their numerical solutions in the formula (7.6) which defines $y(x)$ to obtain a numerical approximation of $y(x)$, i.e. a numerical solution for th BVP:

$$y_n = y_n^{(1)} + \frac{\beta - y_N^{(1)}}{y_N^{(2)}}\, y_n^{(2)}, \qquad n = 0, 1, 2 \ldots, N,$$

where N is the final step such that $x_N = b$.

Note that to apply a Runge-Kutta scheme to the second order scalar ODE (7.1).we should first write it as a system first order ODEs

$$\frac{dy}{dx} = z, \qquad \frac{dz}{dx} = f(x, y, z).$$

For example the Euler scheme is then

$$y_{n+1} = y_n + z_n\,\Delta_n, \qquad z_{n+1} = z_n + f(x_n, y_n, z_n)\,\Delta_n,$$

where $\Delta_n = x_{n+1} - x_n$.

7.2 Nonlinear ODE

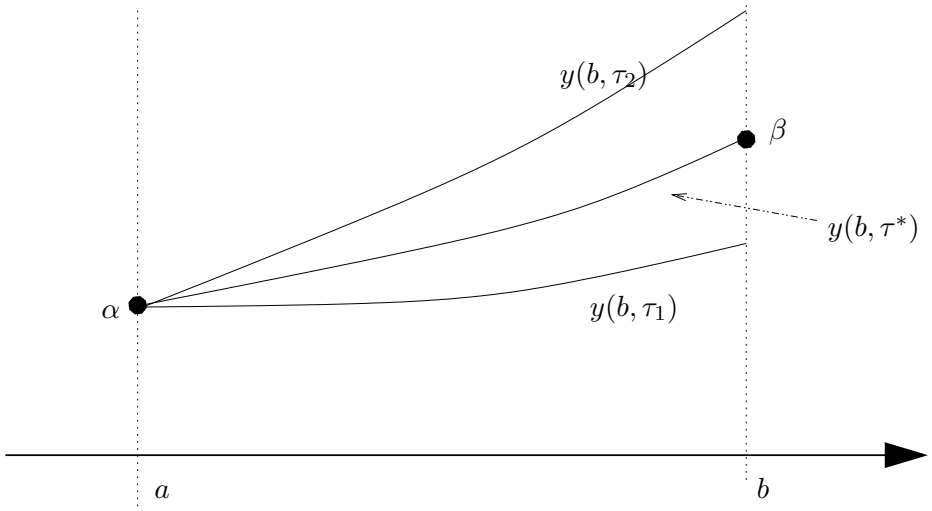
For a nonlinear function f in the ODE we have little chance of finding a formula such as (7.6) which relates the solution of the BVP (7.1)–(7.2) with that of initial value problems for the same ODE. Nevertheless we can still use numerical solutions of appropriate initial value problems for the give ODE to obtain a numerical approximation for the solution of the BVP. Consider the initial value problem

$$\frac{d^2y}{dx^2} = f\left(x, y, \frac{dy}{dx}\right), \qquad y(a) = \alpha, \quad \frac{dy}{dx}(a) = \tau, \tag{7.7}$$

where τ is a parameter. Denote the solution by $y(x, \tau)$. We want to construct a sequence of τ_k such that

$$\lim_{k\to\infty} y(b, \tau_k) = \beta.$$

This method is called the shooting method for obvious reasons — see the picture.



In fact, we want to find a zero of the algebraic equation

$$y(b,\tau) - \beta = 0.$$

Assuming that we have two initial approximations τ_0 and τ_1 such that

$$(y(b,\tau_0) - \beta)\,(y(b,\tau_1) - \beta) < 0,$$

i.e. one solution overshoots the boundary condition and the other undershoots it, then the secant method

$$\tau_k = \tau_{k-1} + \frac{[y(b,\tau_{k-1}) - \beta](\tau_{k-1} - \tau_{k-2})}{y(b,\tau_{k-1}) - y(b,\tau_{k-2})}, \qquad k = 2,3,\ldots,$$

will in principle converge to the desired value of $\tau = \tau^*$ such that $y(b,\tau^*) = \beta$. The solution $y(x,\tau^*)$ of the IVP with this value τ^* is the desired solution of the BVP.

Of course, we do not know $y(b,\tau)$ analytically and need to use a numerical method to find an approximation for it, i.e. by applying, say, a Runge-Kutta scheme to the IVP (7.7). Errors will thus arise from the Runge-Kutta scheme, the secant method and round-off error.

An alternative is to use the Newton method which has a higher order of convergence. This has the form

$$\tau_k = \tau_{k-1} - \frac{y(b,\tau_{k-1}) - \beta}{\dfrac{dy}{d\tau}(b,\tau_{k-1})}, \qquad k = 1,2,\ldots,$$

which requires only a single starting value and the slope of the chord

$$\tau_0 = \frac{\beta - \alpha}{b - a}$$

is a reasonable choice. The difficulty here is that both $y(b, \tau_{k-1})$ and $\frac{dy}{d\tau}(b, \tau_{k-1})$ are unknown.

However, let us differentiate the nonlinear ODE (7.1), which we write as

$$y''(x,\tau) = f\left(x, y(x,\tau), y'(x,\tau)\right) \quad \text{where} \quad ' = \frac{d}{dx}$$

by τ. Hence

$$\begin{aligned}\frac{\partial}{\partial\tau}y''(x,\tau) &= \frac{\partial f}{\partial x}\left(x, y(x,\tau), y'(x,\tau)\right)\frac{\partial x}{\partial\tau} + \frac{\partial f}{\partial y}\left(x, y(x,\tau), y'(x,\tau)\right)\frac{\partial y}{\partial\tau} \\ &\quad + \frac{\partial f}{\partial y'}\left(x, y(x,\tau), y'(x,\tau)\right)\frac{\partial y'}{\partial\tau},\end{aligned}$$

which reduces to

$$\frac{\partial}{\partial\tau}y'' = \frac{\partial f}{\partial y}\left(x, y(x,\tau), y'(x,\tau)\right)\frac{\partial y}{\partial\tau} + \frac{\partial f}{\partial y'}\left(x, y(x,\tau), y'(x,\tau)\right)\frac{\partial y'}{\partial\tau},$$

since x and τ are independent variables so $\frac{\partial x}{\partial\tau} \equiv 0$. Assuming that we can interchange the order of partial differentiation in x and τ we obtain

$$\frac{d^2}{dx^2}\left(\frac{\partial y}{\partial\tau}\right) = \frac{\partial f}{\partial y}\left(x, y, y'\right)\frac{\partial y}{\partial\tau} + \frac{\partial f}{\partial y'}\left(x, y, y'\right)\frac{d}{dx}\left(\frac{\partial y}{\partial\tau}\right),$$

with

$$\frac{\partial y}{\partial\tau}(a,\tau) = \frac{\partial}{\partial\tau}\alpha = 0$$

and

$$\frac{d}{dx}\left(\frac{\partial y}{\partial\tau}\right)(a,\tau) = \frac{\partial}{\partial\tau}\left(\frac{dy}{dx}(a,\tau)\right) = \frac{\partial}{\partial\tau}\tau = 1.$$

Writing

$$z(x,\tau) := \frac{\partial y}{\partial\tau}(x,\tau)$$

we see that z satisfies the linear ODE

$$\frac{d^2z}{dx^2} = \frac{\partial f}{\partial y}(x,y,y')z + \frac{\partial f}{\partial y'}(x,y,y')\frac{dz}{dx}, \tag{7.8}$$

with the initial value

$$z(a) = 0, \qquad \frac{dz}{dx}(a) = 1. \tag{7.9}$$

Its solution $z(x, \tau)$ at $x = b$ gives us

$$z(b, \tau) = \frac{\partial y(b, \tau)}{\partial \tau}.$$

Newton's method now reads

$$\tau_k = \tau_{k-1} - \frac{y(b, \tau_{k-1}) - \beta}{z(b, \tau_{k-1})}, \qquad k = 1, 2, \ldots.$$

Thus we have to solve a nonlinear IVP for $y(x, \tau)$ for a given τ and we use its values in the coefficients of the linear IVP for $z(x, \tau)$. Of course, we have to solve both of these IVP with a numerical method such as a Runge-Kutta scheme.

Kapitel 8

Partielle Differentialgleichungen

Die allgemeine Form einer linearen partiellen Differentialgleichung zweiter Ordnung auf $\mathbb{R}^d (d \geq 2)$ lautet:

$$\boxed{\sum_{i,j=1}^{d} a_{i,j} \frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{j=1}^{d} b_j \frac{\partial u}{\partial x_j} + cu + f = 0}$$

auf einem Gebiet $G \subset \mathbb{R}^d$, wobei sind:

1) $u : G \to \mathbb{R}$ 2-mal stetig differenzierbar

2) $\sum_{i,j=1}^{d} a_{i,j}^2 \neq 0$, d.h. nicht alle $a_{i,j} \equiv 0 \Rightarrow$ PDGL 2-ter Ordnung!

3) $a_{i,j}$, b_j, c, f Funktionen $G \to \mathbb{R}$ oder Konstanten.

Es gibt 3 Klassen solcher PDGLen:

elliptisch, parabolisch, hyperbolisch

falls die quadratische Form auf $\mathbb{R}^d$

$$\sum_{i,j=1}^{d} a_{i,j} Z_i Z_j + \sum_{j=1}^{d} b_j Z_j$$

eine Ellipse, Parabel oder Hyperbel ist.

Bemerkung Die Klasse ist fest, falls alle $a_{i,j}$, b_j Konstanten sind. Sonst betrachten wir Teilgebiete G_e, G_p, G_h, wo die entsprechenden Eigenschaften gelten.

BEISPIELE $(d = 2)$

(1) elliptischer Fall $x_1 = x,\ x_2 = y$

$$\boxed{\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y)} \qquad \text{Poisson-Gleichung}$$

$$(f \equiv 0 \Rightarrow \text{ Laplace-Gleichung})$$

$$\text{quadratische Form } \Rightarrow \quad 1 \cdot Z_1^2 + 1 \cdot Z_2^2 \qquad \text{Ellipse}$$

(2) parabolischer Fall $x_1 = x$, $x_2 = t$ (Zeit)

$$\boxed{\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}} \qquad \text{Wärme-Gleichung}$$

$$\text{quadratische Form } \Rightarrow \quad 1 \cdot Z_1^2 + 0 \cdot Z_2^2 - 1 \cdot Z_2 \qquad \text{Parabel}$$

(3) hyperbolischer Fall $x_1 = x,\ x_2 = t$ (Zeit)

$$\boxed{\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2}} \qquad \text{Wellen-Gleichung}$$

$$\text{quadratische Form } \Rightarrow \quad 1 \cdot Z_1^2 - 1 \cdot Z_2^2 \qquad \text{Hyperbel}$$

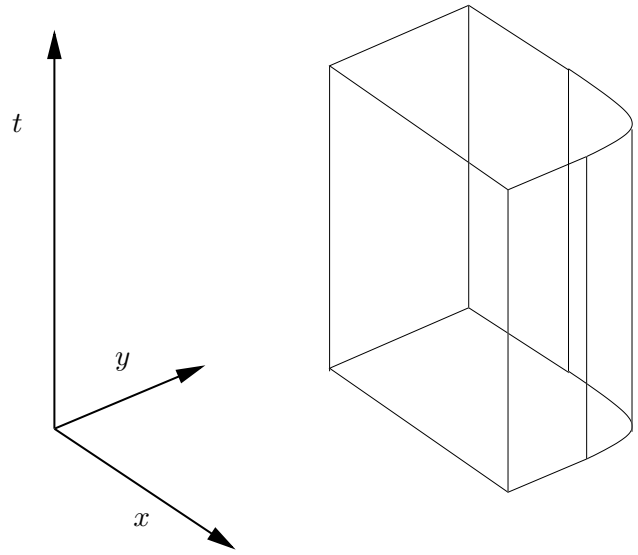
Diese 3 Beispiele sind die Grundgleichungen der mathematischen Physik und der Ingenieurwissenschaften.

Verallgemeinerungen

Der d-dimensionale Differentialoperator (d = räumliche Dimension)

$$\Delta \ := \ \frac{\partial^2 u}{\partial x_1^2} + \ldots + \frac{\partial^2 u}{\partial x_d^2}$$

heißt d-dimensionaler Laplace-Operator. Dann heißt



(1) $\Delta u = f$ d-dimensionale Poisson-Gleichung

(2) $\dfrac{\partial u}{\partial t} = \Delta u$ d-dimensionale Wärme-Gleichung

(3) $\dfrac{\partial^2 u}{\partial t^2} = \Delta u$ d-dimensionale Wellen-Gleichung

Die Wärme- und Wellen-Gleichungen hier sind tatsächlich PDGLen auf $\mathbb{R}^{d+1}$ statt $\mathbb{R}^d$. Wir betrachten diese PDGLen auf einem Zylindergebiet

$$(x,t) \in G \times [0,T] \subset \mathbb{R}^{d+1}$$

wobei G ein Gebiet in $\mathbb{R}^d$ ist.

Die „Ruhelagen“ der Wärme- oder Wellen-Gleichung (d.h. mit $\dfrac{\partial u}{\partial t} \equiv 0$, bzw. $\dfrac{\partial^2 u}{\partial t^2} \equiv 0$) sind Lösungen der entsprechenden Laplace-Gleichung

$$\Delta u = 0 \quad \text{auf} \quad G.$$

Randbedingungen

Betrachte die 1-dimensionale Laplace-Gleichung auf dem Intervall $[0,1]$:

$$\frac{d^2u}{dx^2} = 0, \qquad x \in [0,1],$$

(tatsächlich eine gewöhnliche Differentialgleichung!)

Die allgemeine Lösung lautet $u(x) = Ax + B$

d.h. eine Familie von Lösungen mit 2 Parametern A, B. Um eine bestimmte Lösung auszuwählen, können wir fordern, z.B. dass die Lösung einer gegebenen Randbedingung genügt: z.B.

$$u(0) = 0,\ u(1) = 0 \quad \Rightarrow \quad A = B = 0 \quad \Rightarrow \quad u(x) \equiv 0$$

$$u(0) = 0,\ \frac{du}{dx}\,(1) = 1 \quad \Rightarrow \quad A = 1,\ B = 0 \quad \Rightarrow \quad u(x) = x$$

Der allgemeine d-dimensionale Fall ist ähnlich, aber ein bisschen komplizierter, z.B.

(1) Dirichlet-Randbedingung

$$\boxed{u = 0 \ \text{ auf } \ \partial G}$$

(2) Neumann-Randbedingung

$$\boxed{\frac{\partial u}{\partial n} = 0 \ \text{ auf } \ \partial G}$$

$\frac{\partial u}{\partial n} = n\nabla u$ ist die normalen Ableitung, dabei ist $n = n(x)$ der Normalenvektor

(3) Gemischte-Randbedingung

$$\alpha u + \beta\,\frac{\partial u}{\partial n} = 0 \ \text{ auf } \ \partial G$$

wobei $\alpha^2 + \beta^2 \neq 0$ $\qquad$ ($\alpha = \alpha(x),\ \beta = \beta(x)$ erlaubt).

Anfangsbedingungen

Wärme- wie auch Wellen-Gleichungen brauchen zusätzliche „Zeit“-Randbedingungen, d.h. bzg. der t-Variable – eine Bedingung für jede t-Ableitung

(1) Wärme-Gleichung

$$\frac{\partial u}{\partial t} = \Delta u \quad \text{auf} \quad G \times [0,T] \qquad \boxed{u(x,0) = u_0(x) \quad \text{auf} \quad G}$$

(2) Wellen-Gleichung

$$\frac{\partial^2 u}{\partial t^2} = \Delta u \quad \text{auf} \quad G \times [0,T] \qquad \boxed{\left.\begin{array}{lcl} u(x,0) & = & u_0(x) \\ \dfrac{\partial u}{\partial t}\,(x,0) & = & v_0(x) \end{array}\right\} \quad \text{auf} \quad G}$$

Hier sind u_0 und v_0 gegebene Funktionen, die der Randbedingung unter Betracht auch genügen.

Wir benutzen die selben Rand/Anfangsbedingungen für solche PDGLen zweiter Ordnung mit zusätzlichen Termen niedriger Ordnung – die auch nichtlinear sein dürfen, z.B.

Burgers-Gleichung

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + u\,\frac{\partial u}{\partial x}$$

Reaktion-Diffusion-Gleichung

$$\frac{\partial u}{\partial t} = \triangle u + f(u) \qquad \text{z.B.} \quad f(u) = u(1-u)$$

8.1 Explizite Lösungen

Manchmal können wir eine explizite Lösung für eine lineare PDGL finden, aber meistens nur für einfache Gebiete G wie Intervalle oder Rechtecke.

BEISPIEL 1-dimensionale Wärme-Gleichung

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \qquad x \in [0,1], \quad t \geq 0,$$

$$\left.\begin{array}{rcl} u(0,t) & = & 0 \\ u(1,t) & = & 0 \end{array}\right\} \quad \underline{\text{Randbedingungen}} \ \ \forall t \in [0,T]$$

$$u(x,0) = u_0(x) \qquad \underline{\text{Anfangsbedingung}} \ \ \forall x \in [0,1]$$

Betrachte eine trennbare Lösung der Form

$$u(x,t) = X(x)T(t)$$

<u>d.h.</u>, das Produkt einer Funktion X von x und einer Funktion T von t. Eine solche Lösung genügt der obigen Randbedingung, falls

$$X(0) = X(1) = 0,$$

und sie genügt der PDGL, falls

$$X(x)\ \frac{dT}{dt}\ (t) = \frac{d^2X(x)}{dx^2}\ T(t)$$

für alle $x \in [0,1]$ und $t \in [0,T]$, <u>d.h.</u> falls

$$\frac{1}{T(t)}\ \frac{dT}{dt}\ (t) \equiv \frac{1}{X(x)}\ \frac{d^2X}{dx^2}\ (x)$$

für alle $x \in [0,1]$ und $t \in [0,T]$. Aber dies ist nur möglich, falls die beiden Seiten <u>konstant</u> sind (weil t und x unabhängige Variablen sind).

Sei λ eine Konstante mit

$$\frac{1}{T(t)}\ \frac{dT}{dt}\ (t) \equiv \lambda \equiv \frac{1}{X(x)}\ \frac{d^2X}{dx^2}(x)$$

für alle $x \in [0,1]$ und $t \in [0,T]$. Dann haben wir

(1) <u>Anfangswertaufgabe</u>

$$\frac{dT}{dt}\ (t) - \lambda T(t) = 0 \qquad \Rightarrow \quad \text{Lösung} \quad T(t) = T(0)e^{\lambda t}$$

(2) <u>Randwertaufgabe</u>

$$\frac{d^2X}{dx^2}\ (x) - \lambda X(x) = 0 \qquad \text{mit} \quad X(0) = X(1) = 0$$

$X(x) \equiv 0$ ist eine Lösung für alle $\lambda \in \mathbb{R}$. Aber es gibt nichttriviale Lösungen nur für bestimmte Konstanten $\lambda_1, \lambda_2, \ldots, \lambda_k$

$$\Rightarrow \ \underline{\text{Eigenwert}}/\underline{\text{Eigenfunktionen}} \ !$$

Die DG besitzt die allgemeine Lösung

$$X(x) = \begin{cases} Ae^{\alpha x} + Be^{-\alpha x} & \text{falls} \quad \lambda = \alpha^2 > 0 \\ A + Bx & \text{falls} \quad \lambda = 0 \\ A\cos\beta x + B\sin\beta x & \text{falls} \quad \lambda = -\beta^2 < 0 \end{cases}$$

Aufgrund der Randbedingungen für X erhalten wir im letzten Fall nur mit

$$\lambda = \lambda_k = -k^2\pi^2, \qquad k = 1, 2, 3, \ldots \quad (\underline{\text{Eigenwerte}}!)$$

eine <u>nichttriviale</u> Lösung

$$X_k(x) = \sin(k\pi x) \qquad (\text{Eigenfunktionen})$$

Dann sind

$$\boxed{u_k(x,t) = T_k(0)e^{-k^2\pi^2 t}\sin(k\pi x)} \qquad k = 1, 2, 3, \ldots$$

Lösungen der PDGL mit den gegebenen Randbedingungen.

<u>ABER</u> die PDGL ist <u>linear</u> und die RBen sind <u>homogen</u> (d.h., $= 0$).

$\Rightarrow$ jede <u>lineare Kombination</u> (unendlich auch, wenn sie konvergiert) der u_k ist eine Lösung der PDGL mit RBen, <u>z.B.</u>

$$u(x,t) = \sum_{k=1}^{\infty} b_k e^{-k^2\pi^2 t}\sin(k\pi x)$$

Wir haben auch eine <u>Anfangsbedingung</u>

$$u(x,0) = u_0(x), \qquad 0 \le x \le 1.$$

Deshalb brauchen wir $(t = 0)$

$$u_0(x) = \sum_{k=1}^{\infty} b_k \sin(k\pi x) \qquad \underline{\text{Fourier-Sinus-Reihe}}$$

Die <u>Eigenfunktionen</u> $\sin(k\pi x)$, $k = 1, 2, 3, \ldots$ sind paarweise <u>orthogonal</u> im Sinne:

$$\int_0^1 \sin(k\pi x)\sin(\ell\pi x)\,dx = \begin{cases} 0, & \text{falls} \quad k \neq \ell \\ 1, & \text{falls} \quad k = \ell \end{cases}$$

$$\Rightarrow \qquad \int_0^1 u_0(x)\sin(\ell\pi x)\,dx = \sum_{k=1}^{\infty} b_k \int_0^1 \sin(\ell\pi x)\sin(k\pi x)\,dx = b_\ell$$

d.h.

$$\boxed{b_k = \int_0^1 u_0(x)\sin(k\pi x)dx} \qquad k = 1, 2, 3\ldots$$

BEISPIEL

$$u_0(x) = \begin{cases} 2x, & \text{falls} \quad 0 \le x \le \frac{1}{2} \\ 2(1-x), & \text{falls} \quad \frac{1}{2} \le x \le 1 \end{cases}$$

$$\Rightarrow \qquad b_k = \frac{8}{\pi^2 k^2} \sin\left(\frac{k\pi}{2}\right)$$

(NB $\quad b_{2\ell} = 0 \quad k = 2\ell$ gerade!)

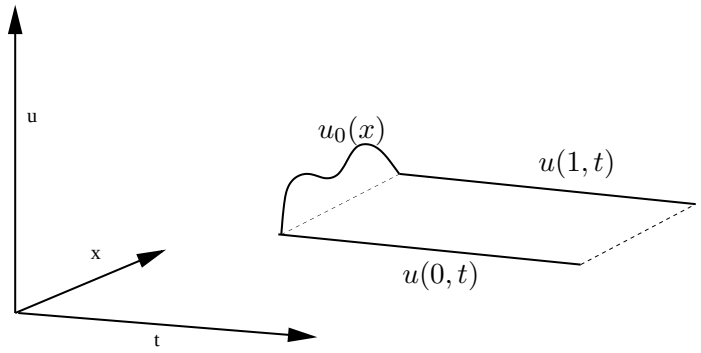
$\Rightarrow$ die explizite Lösung lautet

$$\boxed{u(x,t) = \frac{8}{\pi^2} \sum_{k=1}^{\infty} \frac{1}{k^2} \sin\left(\frac{k\pi}{2}\right) e^{-k^2\pi^2 t} \sin(k\pi x)}$$

8.2 Die 1-dimensionale Wärmegleichung

Betrachte die Anfangsrandwertaufgabe (ARWA)

$$\begin{cases} \frac{\partial u}{\partial t} &= \frac{\partial^2 u}{\partial x^2}, & 0 < x < 1, \quad 0 \le t \le T \\ u(0,t) &= u(1,t) = 0, & 0 \le t \le T \\ u(x,0) &= u_0(x), & 0 \le x \le 1 \end{cases}$$



In diesem sehr speziellen Fall gibt es eine explizite Lösung

$$u(x,t) = \sum_{k=1}^{\infty} b_k e^{-k^2\pi t} \sin(k\pi x)$$

mit $b_k = \int_0^1 u_0(x)\sin(k\pi x)dx, \quad k = 1,2,3,\ldots$

Im Allgemeinen gibt es keine explizite Lösungen, z.B. mit anderen Randbedingungen oder mit zusätzlichen Termen niedriger Ordnung in den PDGLen.

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + f\left(t, x, u, \frac{\partial u}{\partial x}\right)$$

Dann brauchen wir eine numerische Methode. Hier werden wir Differenzenmethoden herleiten, d.h. die Ableitungen werden durch Differenzenquotienten ersetzt.

Wir werden 2 Arten von Differenzenmethoden betrachten:

(1) vollständige Diskretisierung

- ersetze alle (d.h. nach t und x) Ableitungen durch Differenzenquotienten

  $\Rightarrow$ vektorwertige Differenzengleichungen

(2) partielle Diskretisierung oder Semi-Diskretisierung

- ersetze nur die x-Ableitungen durch Differenzenquotienten

  $\Rightarrow$ vektorwertige gewöhnliche Differentialgleichung

- verwende dann ein Runge-Kutta-Verfahren oder ein Mehrschritt-Verfahren

Diese Methode heißt auch Linienmethode oder Methode von Rothe

8.2.1 Differenzenquotienten

Von der deterministischen Taylor-Entwicklung haben wir

1) für die Zeitvariable t

$$f(t+\Delta t) = f(t) + \frac{df}{dt}\,(t)\Delta t + O((\Delta t)^2)$$

$$\Rightarrow \qquad \frac{df}{dt}\,(t) = \frac{f(t+\Delta t) - f(t)}{\Delta t} + O(\Delta t)$$

Vorwärtsdifferenzenquotient

2) für die räumliche Variable x

$$\begin{aligned} g(x \pm \Delta x) &= g(x) \pm \frac{dg}{dx}\,(x)\Delta x + \frac{1}{2!}\,\frac{d^2g}{dx^2}\,(x)(\Delta x)^2 \\ &\quad \pm\frac{1}{3!}\,\frac{d^3g}{dx^3}\,(x)(\Delta x)^3 + O((\Delta x)^4) \end{aligned}$$

addiere und umforme $\Rightarrow$ der zentralisierte Differenzenquotient

$$\frac{d^2g}{dx^2}\,(x) = \frac{g(x+\Delta x) - 2g(x) + g(x-\Delta x)}{(\Delta x)^2} + O((\Delta x)^2)$$

Bemerkung Der Fehler lautet $O((\Delta x)^2)$ hier statt nur $O(\Delta x)$, weil auch die Terme dritter Ordnung in den Taylor-Entwicklungen sich auslöschen.

Deshalb haben wir für $u = u(x,t)$

$$\frac{\partial u}{\partial t}\,(x,t) = \frac{u(x,t+\Delta t) - u(x,t)}{\Delta t} + O(\Delta t)$$

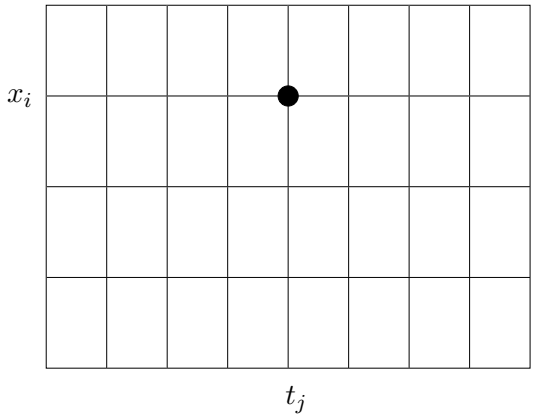
und

$$\frac{\partial^2 u}{\partial x^2}\,(x,t) = \frac{u(x+\Delta x,t) - 2u(x,t) + u(x-\Delta x,t)}{(\Delta x)^2} + O((\Delta x)^2)$$

8.2.2 Vollständige Diskretisierung

Betrachte ein gleichmäßiges Gitter (x_i, t_j) in $[0,1]\times[0,T]$ mit konstanten Schrittweiten

$$\Delta x = h, \qquad \Delta t = k$$



$$x_i = ih, \qquad i = 0, 1, 2, \ldots, N := 1/h$$

$$t_j = jk, \qquad j = 0, 1, 2, \ldots, K := T/k$$

Die PDGL gilt für alle $(x,t) \in (0,1) \times (0,T)$, insbesondsere für die Gitterpunkte $(x_i, t_j) \in (0,1) \times (0,T)$, d.h.

$$\frac{\partial u}{\partial t}\,(x_i, t_j) = \frac{\partial^2 u}{\partial x^2}\,(x_i, t_j).$$

Mit den obigen Differenzenquotienten haben wir dann

$$\frac{u(x_i, t_j + k) - u(x_i, t_j)}{k} = \frac{u(x_i + h, t_j) - 2u(x_i, t_j) + u(x_i - h, t_j)}{h^2} + O(k + h^2)$$

oder

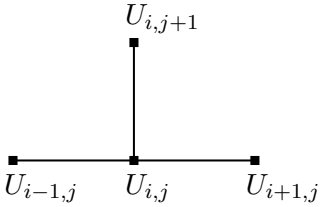
$$\frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{k} = \frac{u(x_{i+1}, t_j) - 2u(x_i, t_j) + u(x_{i-1}, t_j)}{h^2} + O(k + h^2)$$

weil $t_j + k = t_{j+1}$ und $x_i \pm h = x_{i\pm 1}$.

Ersetze $u(x_i, t_j)$ durch $U_{i,j}$ und schneide den lokalen Diskretisierungsfehler $O(k + h^2)$ ab

$$\Rightarrow \qquad \frac{U_{i,j+1} - U_{i,j}}{k} = \frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{h^2}$$

oder



$$\boxed{U_{i,j+1} = U_{i,j} + \frac{k}{h^2}\,(U_{i+1,j} - 2U_{i,j} + U_{i-1,j})}$$

für $i = 1, 2, \ldots, N-1$ und $j = 0, 1, \ldots$.

Bemerkungen

1) Von dem Anfangswert definieren wir $U_{i,0} = u_0(x_i), \quad i = 0, 1, \ldots, N$

2) Von der Randbedingung haben wir

$$U_{0,j} = U_{N,j} = 0, \qquad j = 0, 1, 2, \ldots, K$$

Wir können die Abhängigkeit der verschiedenen $U_{i,j}$ graphentheoretisch darstellen.

Folge: Die Matrix der vektorwertigen Darstellung des obigen Systems linearer Differenzengleichungen ist eine 3-bändige Matrix.

Sei I die $(N-1) \times (N-1)$-Identitäts-Matrix und definiere

$$T = \begin{bmatrix} -2 & 1 & & & & \bigcirc \\ 1 & -2 & 1 & & & \\ & 1 & -2 & \ddots & & \\ & & 1 & \ddots & 1 & \\ & & & \ddots & -2 & 1 \\ \bigcirc & & & & 1 & -2 \end{bmatrix} \quad (N-1) \times (N-1) \text{ 3-bändige Matrix}$$

sowie $r = \dfrac{k}{h^2}$ und $U_j = \begin{pmatrix} U_{1,j} \\ \vdots \\ U_{N-1,j} \end{pmatrix} \in \mathbb{R}^{N-1}$.

Das obige System linearer Differenzengleichungen ist äquivalent der vektorwertigen Differenzengleichung

$$U_{j+1} = A\, U_j$$

wobei

$$A = I + r\, T = \begin{bmatrix} 1-2r & r & & & & & \bigcirc \\ r & 1-2r & r & & & & \\ & r & 1-2r & \ddots & & & \\ & & r & \ddots & r & & \\ & & & \ddots & 1-2r & r & \\ \bigcirc & & & & r & 1-2r & \end{bmatrix}$$

auch 3-bändig!

$$\Rightarrow \qquad \boxed{U_j = A^j\, U_0} \qquad j = 1, 2, 3 \ldots$$

Aber A ist eine $(N-1) \times (N-1)$-Matrix mit $N \gg 1$

$\Rightarrow$ theoretisch günstig, aber nicht immer praktisch.

8.2.3 Linienmethode

Hier ersetzen wir nur die räumliche(n) Ableitung(en) durch Differenzenquotienten. Betrachte eine gleichmäßige Zerlegung des Intervalls $[0, 1]$ mit Schrittweite $h = \Delta x = 1/N$

$$\Rightarrow \qquad x_i = ih, \qquad i = 0, 1, \ldots, N$$

Von der PDGL mit $x = x_i \in (0, 1)$, d.h.

$$\frac{\partial u}{\partial t}\,(x_i, t) = \frac{\partial^2 u}{\partial x^2}\,(x_i, t),$$

und dem zentralisierten Differenzenquotienten erhalten wir

$$\begin{aligned} \frac{\partial u}{\partial t}\,(x_i, t) &= \frac{u(x_i + h, t) - 2u(x_i, t) + u(x_i - h, t)}{h^2} + O(h^2) \\ &= \frac{u(x_{i+1}, t) - 2u(x_i, t) + u(x_{i-1}, t)}{h^2} + O(h^2) \end{aligned}$$

Ersetze $u(x_i, t)$ durch $U_i(t)$ durch $U_i(t)$ $(i = 0, 1, \ldots, N)$ und schneide den lokalen räumlichen Diskretisierungsfehler ab.

$$\Rightarrow \qquad \boxed{\frac{d}{dt}\, U_i(t) = \frac{1}{h^2}\, (U_{i+1}(t) - 2U_i(t) + U_{i-1}(t))}$$

mit

$$U_0(t) \equiv U_N(t) \equiv 0 \qquad \text{Randbedingung}$$

$$U_i(0) = u_0(x_i), \quad i = 0, 1, \ldots, N, \qquad \text{Anfangswert}$$

Definiere

$$U(t) = \begin{pmatrix} U_1(t) \\ \vdots \\ U_{N-1}(t) \end{pmatrix} \in \mathbb{R}^{N-1}$$

Das obige System linearer Differentialgleichungen ist äquivalent der vektorwertigen linearen Differentialgleichung

$$\boxed{\frac{d}{dt}\, U(t) = \frac{1}{h^2}\, TU(t)}$$

wobei T die obige $(N-1) \times (N-1)$-Bandmatrix ist.

Dann versuchen wir diese Differentialgleichung numerisch zu lösen, z.B. mit einem Runge-Kutta-Verfahren.

Beispiel Das Euler-Verfahren mit konstanter Zeitschrittweite $\Delta t \equiv k$ lautet

$$V_{j+1} = V_j + k\, \frac{1}{h^2}\, TV_j$$

mit $V_j \simeq U(t_j)$, d.h.

$$V_{j+1} = (I + rT)V_j \qquad \text{mit} \quad r = \frac{k}{h^2}$$

$\Rightarrow$ genau wie im Fall vollständiger Diskretisierung!

ABER wir können auch ein Runge-Kutta-Verfahren höherer Ordnung verwenden, z.B. das Heun-Verfahren

$$\begin{aligned} V_{j+1} &= V_j + \frac{1}{2}\, k \left[\frac{1}{h^2}\, TV_j + \frac{1}{h^2}\, T \left(V_j + \frac{k}{h^2}\, TV_j \right) \right] \\ &= \left[I + \frac{k}{h^2}\, T + \frac{1}{2}\, \frac{k^2}{h^4}\, T^2 \right]\, V_j \end{aligned}$$

d.h., nochmal mit $r = \dfrac{k}{h^2}$,

$$\boxed{V_{j+1} = \left[I + rT + \frac{1}{2}\, r^2 T^2 \right] V_j}$$

Kapitel 9

Differenzenmethoden für PDGLen

9.1 Numerische Stabilität

Wir betrachten nochmals die Anfangsrandwertaufgabe der parabolischen PDGL

$$\begin{cases} \underline{\text{PDGL}} & \dfrac{\partial u}{\partial t} = \dfrac{\partial^2 u}{\partial x^2}, & 0 < x < 1, 0 \le t \le T \\ \underline{\text{RB}} & u(0,t) = u(1,t) = 0, & 0 \le t \le T \\ \underline{\text{AB}} & u(x,0) = u_0(x), & 0 \le x \le 1 \end{cases}$$

und ein gleichmäßiges Gitter auf $(0,1) \times (0,T)$ mit konstanten Schrittweiten

$$k = \Delta t, \quad h = \Delta x$$

$$\begin{aligned} \Rightarrow \quad t_j &= jk, \qquad j = 0, 1, \ldots, \quad K := T/k \\ x_i &= ih, \qquad i = 0, 1, \ldots, \quad N := 1/h \end{aligned}$$

Wir haben gesehen, dass die <u>volldiskretisierte Differenzenmethode</u> und die <u>Linienmethode</u> mit dem <u>Euler-Verfahren</u> die vektorwertige Differenzengleichung

$$U_{j+1} = (I + rT)\, U_j$$

ergeben, wobei

$$T = \begin{bmatrix} -2 & 1 & & & & \bigcirc \\ 1 & -2 & 1 & & & \\ & 1 & -2 & \ddots & & \\ & & 1 & \ddots & 1 & \\ & & & \ddots & -2 & 1 \\ \bigcirc & & & & 1 & -2 \end{bmatrix} \quad 3-\text{bändige } (N-1)\times(N-1)-\text{ Matrix}$$

sowie $r = \dfrac{k}{h^2}$ und $U_j = \begin{pmatrix} U_{1,j} \\ \vdots \\ U_{N-1,j} \end{pmatrix} \in \mathbb{R}^{N-1}$.

Definiere

$$A = I + r\ T = \begin{bmatrix} 1-2r & r & & & & & \bigcirc \\ r & 1-2r & r & & & & \\ & r & 1-2r & \ddots & & & \\ & & & r & \ddots & r & \\ & & & & \ddots & 1-2r & r \\ \bigcirc & & & & & r & 1-2r \end{bmatrix}$$

Dann gilt

$$U_{j+1} = AU_j \qquad \text{oder} \quad U_j = A^j U_0$$

für $j = 0, 1, 2, \ldots$.

Betrachte jetzt einen Abrundungs- oder Datenfehler E_0 in U_0. Statt U_0 haben wir dann den Anfangswert $U_0^* = U_0 + E_0$ und statt U_j erhalten wir

$$U^* = A^j\ U_0^*.$$

Wegen der Linearität genügt der Fehler $E_j = U_j^* - U_j$ der selben Gleichung, d.h.

$$E_j = A^j\ E_0$$

Wir sagen, dass die numerische Methode numerisch stabil ist, wenn eine Konstante B existiert mit

$$|E_j| \leq B\ |E_0|$$

für alle $j = 0, 1, 2, \ldots$.

$\Rightarrow$ Die Ruhelage $\overline{X} \equiv 0$ der Differenzengleichung $X_{j+1} = AX_j$ ist Ljapunov stabil.

$\Rightarrow$ Die Eigenwerte der Matrix A genügen der Bedingung

1) $|\lambda| < 1$
oder
2) $|\lambda| = 1$ und λ ist halbeinfach

FRAGEN *Wie sind die Eigenwerte der 3-bändigen Matrix $A = I + rT$? Wie hängen die Eigenwerte von r ab?*

HILFSATZ Sei $a, b, c \in \mathbb{R}$ mit $bc > 0$. Dann besitzt die 3-bändige $L \times L$ Matrix

$$M_L = \begin{bmatrix} a & b & & & & \bigcirc \\ c & a & b & & & \\ & c & a & \ddots & & \\ & & c & \ddots & b & \\ & & & \ddots & a & b \\ \bigcirc & & & & c & a \end{bmatrix}$$

die Eigenwerte

$$\lambda_k = a + 2\sqrt{bc}\ \cos\left(\frac{k\pi}{L+1}\right), \qquad k = 1, 2, \ldots, L$$

Beweis (Skizze) Sei $L \geq 3$. Dann gilt

$$\det(M_L - \lambda I_L) = \det \begin{bmatrix} a-\lambda & b & & & & \bigcirc \\ c & a-\lambda & b & & & \\ & c & a-\lambda & \ddots & & \\ & & c & \ddots & b & \\ & & & \ddots & a-\lambda & b \\ \bigcirc & & & & c & a-\lambda \end{bmatrix}$$

$$= (a-\lambda)\det(M_{L-1} - \lambda I_{L-1}) - c \det \begin{bmatrix} b & 0 & & & & \bigcirc \\ c & a-\lambda & b & & & \\ & c & a-\lambda & \ddots & & \\ & & c & \ddots & b & \\ & & & \ddots & a-\lambda & b \\ \bigcirc & & & & c & a-\lambda \end{bmatrix}$$

$$= (a-\lambda)\det(M_{L-1} - \lambda I_{L-1}) - bc\det(M_{L-2} - \lambda I_{L-2})$$

<u>d.h.</u> $\boxed{D_L = (a-\lambda)D_{L-1} - bcD_{L-2}}$ $(L \geq 3)$ $D_L = \det(M_L - \lambda I_L)$, usw.

Aber

$$D_1 = \det[a-\lambda] = a - \lambda \qquad \Rightarrow \qquad \lambda = a = a + 0\cos\frac{\pi}{2}$$

und

$$D_2 = \det \begin{bmatrix} a-\lambda & b \\ c & a-\lambda \end{bmatrix} = (a-\lambda)^2 - bc$$

$$\Rightarrow \qquad \lambda = a \pm \sqrt{bc} = a + 2\sqrt{bc}\cos\left(\frac{k\pi}{3}\right)$$

Mit $L = 3$ erhalten wir dann:

$$\begin{aligned} D_3 &= (a-\lambda)\,[(a-\lambda)^2 - bc] - bc(a-\lambda) \\ &= (a-\lambda)\,[(a-\lambda)^2 - 2bc] \end{aligned}$$

mit Eigenwert (d.h. $D_3 = 0$) $\quad \Rightarrow \quad \lambda = a,\, a \pm \sqrt{bc}$

<u>d.h.</u> $\lambda_k = a + 2\sqrt{bc}\cos\left(\frac{k\pi}{4}\right),\ k = 1, 2, 3$

weil $\cos\frac{\pi}{4} = \frac{1}{\sqrt{2}},\ \cos\frac{2\pi}{4} = 0,\ \cos\frac{3\pi}{4} = -\frac{1}{\sqrt{2}}$ □

Betrachte jetzt die 3-bändige $(N-1)\times(N-1)$-Matrix

$$T = \begin{bmatrix} -2 & 1 & & & & \bigcirc \\ 1 & -2 & 1 & & & \\ & 1 & -2 & \ddots & & \\ & & 1 & \ddots & 1 & \\ & & & \ddots & -2 & 1 \\ \bigcirc & & & & 1 & -2 \end{bmatrix}$$

Hier ist $a = -2$, $b = c = 1$, $L = N - 1$

$$\begin{aligned} \Rightarrow \qquad \lambda_k(T) &= -2 + 2\cos\left(\frac{k\pi}{(N-1+1)}\right) \\ &= -2 + 2\cos\left(\frac{k\pi}{N}\right) \\ &= -2\left[1 - \cos\left(2\frac{k\pi}{2N}\right)\right], \end{aligned}$$

d.h., $\qquad \boxed{\lambda_k(T) = -4\sin^2\left(\frac{k\pi}{2N}\right)} \qquad k = 1, 2, \ldots, N-1.$

In ähnlicherweise erhalten wir für die Matrix $A = I + rT$

$$\boxed{\lambda_k(A) = 1 - 4r\sin^2\left(\frac{k\pi}{2N}\right)} \qquad k = 1, 2, \ldots, N-1.$$

Wir haben

$$|\lambda_k(A)| \le 1$$

genau dann, wenn $\boxed{r \le 1/2}$ weil (für alle N)

$$-1 \le 1 - 4r\sin^2\left(\frac{k\pi}{2N}\right) \le +1 \quad \Leftrightarrow \quad 4r\sin^2\left(\frac{k\pi}{2N}\right) \le 2 \quad \Leftrightarrow \quad r \le 1/2$$

SATZ Die volldiskretisierte Differenzenmethode ist numerisch stabil genau dann, wenn

$$r = \frac{k}{h^2} \leq 1/2 \qquad \text{Courant-Stabilitätskriterion}$$

$\Rightarrow$ die Zeitschrittweite $k = \Delta t$ muß oft sehr klein sein, z.B.

$$h = \Delta x = 10^{-4} \quad \Rightarrow \quad k \leq \frac{1}{2}\, h^2 \leq \frac{1}{2}\, 10^{-8}$$

$\Rightarrow$ Computer-Aufwand sehr hoch!

9.2 Die Methode von Crank-Nicolson

Wir wissen, dass implizite numerische Verfahren oft stabiler für größere Schrittweiten sind.

Betrachte die gewöhnliche DGL

$$\frac{dx}{dt} = f(x)$$

- explizites Euler-Verfahren $\qquad x_{n+1} = x_n + kf(x_n)$
- implizites Euler-Verfahren $\qquad x_{n+1} = x_n + kf(x_{n+1})$

Im Allgemeinen

- θ-Verfahren mit $\theta \in [0,1]$ lautet

$$x_{n+1} = x_n + k\,[(1-\theta)f(x_n + \theta f(x_{n+1})]$$

Für $\theta = 0$ das explizite Euler-Verfahren, $\theta = 1$ das implizierte Euler-Verahren und $\theta = 1/2$ das Trapez-Verfahren

$$x_{n+1} = x_n + \frac{1}{2}\ [f(x_n) + f(x_{n+1})]$$

Das Trapez-Verfahren für die gewöhnliche DGL der Linienmethode, d.h.

$$\frac{d}{dt}\ U(t) = \frac{1}{h^2}\ TU(t)$$

lautet

$$V_{j+1} = V_j + \frac{1}{2}\, k \left[\frac{1}{h^2}\, TV_j + \frac{1}{h^2}\, TV_{j+1} \right]$$

weil $f(U) = \dfrac{1}{h^2}\, TU$ hier.

Dann mit $r = k/h^2$ haben wir

$$V_{j+1} = V_j + \frac{1}{2}\, r\, [TV_j + TV_{j+1}]$$

Diese Differenzengleichung ist implizit, aber linear. Wir erhalten

$$\boxed{V_{j+1} = (2I - rT)^{-1}(2I + rT)\, V_j}$$

d.h., die Methode von Crank-Nicolson (1957).

In diesem Fall haben wir

$$V_{j+1} = A\, V_j$$

mit der $(N-1) \times (N-1)$-Matrix

$$A = (2I - rT)^{-1}(2I + rT)$$

HILFSATZ (Siehe z.B. Gantmacher)
Sei λ ein Eigenwert der $L \times L$-Matrix M. Dann ist $f(\lambda)$ ein Eigenwert von $f(M)$, wo f eine rationale Funktion ist.

Wir wissen, dass die $(N-1) \times (N-1)$-Matrix T die Eigenwerte

$$\lambda_k = -4 \sin^2 \left(\frac{k\pi}{2N} \right), \qquad k = 1, 2, \ldots, N-1,$$

besitzt.

Definiere $f(x) = (2 - rx)^{-1}(2 + rx)$.

Dann gilt

$$A = F(T) = (2I - rT)^{-1}(2I + rT)$$

und für die Eigenwerte von A benutzen wir den Hilfsatz und erhalten

$$\lambda_k(A) = f(\lambda_k(T)) = \frac{2 - 4r\sin^2\left(\frac{k\pi}{2N}\right)}{2 + 4r\sin^2\left(\frac{k\pi}{2N}\right)},$$

für $k = 1, 2, \ldots, N-1$,

$$\Rightarrow \qquad \boxed{|\lambda_k(A)| < 1 \quad \text{für} \quad r > 0}$$

Die Methode von Crank-Nicolson ist absolut stabil,

d.h. numerisch stabil ohne Beschränkung von $k = \Delta t$ oder $h = \Delta x$.

Um diese Methode auszuführen brauchen wir die LR-Zerlegung – Gauß-Elimination – nur einmal zu berechnen, um die $(N-1)$-dimensionale Gleichung

$$(2I - rT)x = (2I + rT)b$$

beliebig oft zu lösen.

Die LR-Zerlegung ist ziemlich einfach, weil die Matrix $2I - rT$ 3-bändig ist. Falls die Matrix zusätzlich symetrisch und positive definite ist, empfiehlt sich eine Cholesky Zerlegung.

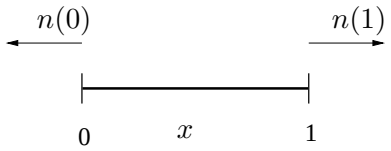
9.3 Andere Randbedingungen

Betrachte jetzt die Anfangsrandwertaufgabe

$$\text{PDGL} \qquad \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad 0 < x < 1, \; 0 \le t \le T$$

$$\text{RBen} \quad \begin{cases} \frac{\partial u}{\partial n}(0,t) + \alpha u(0,t) &= 0, \\ \\ \frac{\partial u}{\partial n}(1,t) + \beta u(1,t) &= 0, \end{cases} \qquad 0 \le t \le T,$$

$$\text{AB} \qquad u(x,0) = u_0(x), \quad 0 \le x \le 1$$



mit α, $\beta \geq 0$, insbesondere mit den Neumann-RBen ($\alpha = \beta = 0$) oder den gemischten Cauchy-RBen ($\alpha^2 + \beta^2 \neq 0$).

Sei $n(x)$ der Normalenvektor zum Rand $\{0, 1\}$ des Intervalles $(0, 1)$:

Dann lauten die normalen Ableitungen

$$\underline{x = 0} \qquad \frac{\partial u}{\partial n}(0,t) = n(0)\nabla u(0,t) = -\frac{\partial u}{\partial x}(0,t)$$

$$\underline{x = 1} \qquad \frac{\partial u}{\partial n}(1,t) = n(1)\nabla u(1,t) = +\frac{\partial u}{\partial x}(1,t)$$

Wir schreiben die Randbedingungen um und erhalten

$$\begin{array}{lrcll} \underline{x = 0} & \frac{\partial u}{\partial x}(0,t) & = & \alpha u(0,t) & \\ & & & & (0 \leq t \leq T) \\ \underline{x = 1} & \frac{\partial u}{\partial x}(1,t) & = & -\beta u(1,t) & \end{array}$$

Die volldiskretisierte Differenzenmethode für die obige PDGL auf dem Gitter

$$(x_i, t_j) = (ih, jk), \qquad i = 0, 1, \ldots, N := 1/h, \quad j = 0, 1, \ldots, K := T/k$$

lautet (wie vorher)

$$\boxed{U_{i,j+1} = U_{i,j} + r\,(U_{i+1,j} - 2U_{i,j} + U_{i-1,j})}$$

wobei $r = k/h^2$ und $U_{i,j} \approx u(x_i, t_j)$.
Der räumliche Diskretisierungsfehler hier ist $O(h^2)$. Um diese Ordnung zu behalten, müssen wir einen Differenzenquotienten für $\frac{\partial u}{\partial x}$ in den RBen mit Fehler $O(h^2)$ benutzen. Deswegen brauchen wir den zentralisierten Differenzenquotienten

$$\frac{\partial u}{\partial x}(x,t) = \frac{u(x+h,t) - u(x-h,t)}{2h} + O(h^2).$$

Warum? Betrachte die Taylor-Entwicklungen

$$f(x \pm h) = f(x) \pm hf'(x) + \underbrace{\frac{1}{2}\, h^2 f''(x)}_{\text{wichtig!}} + O(h^3)$$

und subtrahiere:

$$f(x+h) - f(x-h) = 2hf'(x) + O(h^3)$$

$$\Rightarrow \qquad f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2).$$

Mit dieser Approximation lauten unsere RBen jetzt

$$\begin{aligned} \underline{x=0} \qquad \alpha u(0,t) &= \frac{u(h,t) - u(-h,t)}{2h} + O(h^2) \\ \underline{x=1} \qquad -\beta u(1,t) &= \frac{u(1+h,t) - u(1-h,t)}{2h} + O(h^2) \end{aligned}$$

Problem $u(-h,t)$ und $u(1+h,t)$ existieren nicht!

Lösung führe künstliche Gitterpunkte

$$x_{-1} = -h, \qquad x_{N+1} = (N+1)h = 1 + h$$

ein und benutze die obigen RBen ohne die $O(h^2)$-Terme, um

$$U_{-1,j} \simeq u(-h, t_j), \qquad U_{N+1,j} \simeq u(1+h, t_j)$$

zu definieren! d.h.

$$\begin{aligned} \underline{x=0} \qquad \alpha U_{0,j} &= \frac{U_{1,j} - U_{-1,j}}{2h} \\ \Rightarrow \qquad & \boxed{U_{-1,j} = U_{1,j} - 2\alpha h U_{0,j}} \\ \underline{x=1} \qquad -\beta U_{N,j} &= \frac{U_{N+1,j} - U_{N-1,j}}{2h} \\ \Rightarrow \qquad & \boxed{U_{N+1,j} = U_{N-1,j} - 2\beta h U_{N,j}} \end{aligned}$$

Wir benutzen diese Definitionen mit der Differenzengleichung

$$U_{i,j+1} = U_{i,j} + r\,(U_{i+1,j} - 2U_{i,j} + U_{i-1,j})$$

für $i = 0, 1, \ldots, N$ und $j = 0, 1, 2, \ldots$, (d.h. jetzt auch mit $i = 0$ und $i = N$).

$$\begin{aligned}
\underline{i=0} \qquad U_{0,j+1} &= U_{0,j} + r\,(U_{1,j} - 2U_{0,j} + U_{-1,j}) \\
&= U_{0,j} + r\,(2U_{1,j} - 2(1+\alpha h)U_{0,j}) \\
\underline{i=N} \qquad U_{N,j+1} &= U_{N,j} + r\,(U_{N+1,j} - 2U_{N,j} + U_{N-1,j}) \\
&= U_{N,j} + r\,(-2(1+\beta h)U_{N,j} + 2U_{N-1,j})
\end{aligned}$$

Die vektorwertige Version lautet

$$\boxed{U_{j+1} = (I + rS)U_j}$$

wobei I die $(N+1)\times(N+1)$ Identitäts-Matrix ist, $U_j = \begin{pmatrix} U_{0,j} \\ U_{1,j} \\ \vdots \\ U_{N,j} \end{pmatrix} \in \mathbb{R}^{N+1}$,

$$S = \begin{bmatrix}
-2(1+\alpha h) & 2 & & & & & & & 0 \\
1 & -2 & 1 & & & & & & \\
& 1 & -2 & \ddots & 1 & & & & \\
& & 1 & \ddots & -2 & \ddots & 1 & & \\
& & & \ddots & 1 & \ddots & -2 & 1 & \\
& & & & & \ddots & 1 & -2 & 1 \\
0 & & & & & & & 2 & -2(1-\beta h)
\end{bmatrix}$$

Definiere

$$A = I + rS = \begin{bmatrix}
1-2r(1+\alpha h) & 2r & & & & & & & 0 \\
r & 1-2r & r & & & & & & \\
& r & 1-2r & \ddots & r & & & & \\
& & r & \ddots & 1-2r & \ddots & r & & \\
& & & \ddots & r & \ddots & 1-2r & r & \\
& & & & & \ddots & r & 1-2r & r \\
0 & & & & & & & 2r & 1-2r(1-\beta h)
\end{bmatrix}$$

d.h., A ist eine 3-bändige $(N+1)\times(N+1)$ Matrix

$$\Rightarrow \qquad \boxed{U_{j+1} = AU_j}$$

Die volldiskretisierte Differenzenmethode für diese ARWA ist numerisch stabil genau dann, wenn die Eigenwerte $\lambda_k(A)$ der Matrix A den folgenden Bedingungen genügen:

1) $|\lambda_k(A)| < 1$ oder 2) $|\lambda_k(A)| = 1$ mit $\lambda_k(A)$ halbeinfach

Wie sind die Eigenwerte $\lambda_k(A)$?

Offensichtlich gilt $\lambda_k(A) = 1 + r\lambda_k(S)$.

Wie sind die Eigenwerte $\lambda_k(S)$?

Die $(N+1)\times(N+1)$-Matrix S ist nicht symmetrisch.

ABER S ist ähnlich zu einer symmetrischen Matrix $\widetilde{S} = D^{-1}SD$, wobei

$$D = \begin{bmatrix} \sqrt{2} & & & & 0 \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ 0 & & & & \sqrt{2} \end{bmatrix} \quad (N+1)\times(N+1) \text{ diagonale Matrix}$$

und

$$\widetilde{S} = \begin{bmatrix} -2(1+\alpha h) & \sqrt{2} & & & & & 0 \\ \sqrt{2} & -2 & 1 & & & & \\ & 1 & -2 & \ddots & & & \\ & & 1 & \ddots & 1 & & \\ & & & \ddots & -2 & 1 & \\ & & & & 1 & -2 & \sqrt{2} \\ 0 & & & & & \sqrt{2} & -2(1+\beta h) \end{bmatrix}$$

$\Rightarrow$ die Eigenwerte $\lambda_k(\widetilde{S}) \equiv \lambda_k(S)$ sind alle reellwertig

$\Rightarrow$ Die Eigenwerte $\lambda_k(A)$ sind alle reellwertig!

Aber die Bandmatrizen A, S, $\widetilde{S}$ sind nicht der tridiagonalen Form

$$\begin{bmatrix} a & b & & & \\ c & a & b & & \\ & c & a & b & \\ & & \ddots & \ddots & \ddots \end{bmatrix}$$

des Hilfsatzes.

$\Rightarrow$ keine einfache Formeln für die Eigenwerte!

Wir müssen die Eigenwerte abschätzen.

SATZ von Gerschgorin (z.B. Stummel/Hainer, Seite 104)

Die Eigenwerte einer $L \times L$-Matrix $A = [a_{i,j}]$ liegen in der Vereinigungsmenge

$$\bigcup_{j=1}^{L} \{z \in \mathbb{C} : \ |z - a_{j,j}| \leq \rho_j\}$$

der Kreisscheiben mit Zentren $a_{j,j}$ und $\rho_j = \sum_{\substack{k=1 \\ k \neq j}}^{L} |a_{j,k}|$, $j = 1, \ldots, L$

Die obige $(N+1) \times (N+1)$-Matrix $A = I + rS$ besitzt nur reellwertige Eigenwerte. Daher müssen wir nur Intervalle statt Kreisscheiben in dem Satz von Gerschgorin betrachten,

$j = 0$ $\qquad$ $\boxed{|\lambda - (1 - 2r(1 + \alpha h)| \leq 2r}$

$$\Leftrightarrow \quad -2r \leq \lambda - 1 + 2r(1 + \alpha h) \leq 2r$$

$$\Leftrightarrow \quad 1 - 2r(2 + \alpha h) \leq \lambda \leq 1 - 2r\alpha h$$

Betrachte die Endwerte

(i) $\lambda = 1 - 2r(2 + \alpha h)$

$$
\begin{aligned}
|\lambda| \le 1 \quad &\Leftrightarrow \quad -1 \le 1 - 2r(2+\alpha h) \le 1 \\
&\Leftrightarrow \quad -2 \le -2r(2+\alpha h) \le 0 \\
&\Leftrightarrow \quad 2r(2+\alpha h) \le 2 \\
&\Leftrightarrow \quad r \le \frac{1}{2+\alpha h}
\end{aligned}
$$

(ii) $\underline{\lambda = 1 - 2r\alpha h}$

$$
\begin{aligned}
|\lambda| \le 1 \quad &\Leftrightarrow \quad -1 \le 1 - 2r\alpha h \le 1 \\
&\Leftrightarrow \quad 2r\alpha h \le 2 \\
&\Leftrightarrow \quad r \le \frac{1}{\alpha h}
\end{aligned}
$$

$\underline{j = 1, \ldots, N-1}$ $\qquad \boxed{|\lambda - (1-2r)| \le 2r}$

$$\Leftrightarrow \qquad -2r \le \lambda - 1 + 2r \le 2r \qquad \Leftrightarrow 1 - 4r \le \lambda \le 1$$

Daher $|\lambda| \le 1$ gilt, wenn $-1 \le 1 - 4r \quad \Leftrightarrow \quad 4r \le 2 \quad \Leftrightarrow \quad r \le 1/2$

$\underline{j = N}$ $\qquad \boxed{|\lambda - (1 - 2r(1+\beta h)| \le 2r}$

wie im $(j = 0)$-Fall, aber mit β statt α.

Alle Fälle zusammen ergeben

$$|\lambda| \le 1 \;\Leftrightarrow\; r \le \min\left\{ \frac{1}{2+\alpha h},\ \frac{1}{\alpha h},\ \frac{1}{2},\ \frac{1}{2+\beta h},\ \frac{1}{\beta h} \right\}$$

Aber $\alpha, \beta \ge 0$ und $h \ll 1 \;\Rightarrow\; \dfrac{1}{\alpha h},\ \dfrac{1}{\beta h} \ge \dfrac{1}{2}$, $\underline{\text{d.h.}}$

$$|\lambda| \le 1 \qquad \Leftrightarrow \qquad \boxed{r \le \min\left\{ \frac{1}{2+\alpha h},\ \frac{1}{2+\beta h} \right\}}$$

für numerische Stabilität.

Hier muß $r < \frac{1}{2}$! Dies ist *schlechter* als mit den Dirichlet-Randbedingungen.

Bermerkung Die Crank-Nicolson-Methode ist absolut stabil für die neuen RBen – d.h. $\forall r > 0$.

9.4 Zusätzliche Terme niedriger Ordnung

Betrachte die Anfangsrandwertaufgabe für die parabolische PDGL

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + f\left(t, x, u, \frac{\partial u}{\partial x}\right), \quad 0 < x < 1, 0 \le t \le T$$

mit

$$\text{Randbedingungen} \qquad u(0,t) = u(1,t) = 0, \quad 0 \le t \le T,$$

$$\text{Anfangsbedingung} \qquad u(x,0) = u_0(x), \quad 0 \le x \le 1,$$

wobei $f\left(t, x, u, \frac{\partial u}{\partial x}\right)$ eine gegebene Funktion ist, z.B.

1) $f\left(t, x, u, \frac{\partial u}{\partial x}\right) = u \frac{\partial u}{\partial x}$

$\Rightarrow$ PDGL $\boxed{\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + u \frac{\partial u}{\partial x}}$ Burgers-Gleichung

2) $f\left(t, x, u, \frac{\partial u}{\partial x}\right) = u(1-u)$

$\Rightarrow$ PDGL $\boxed{\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + u(1-u)}$ Reaktion-Diffusions-Gleichung

Diese PDGLen sind jetzt nichtlinear, aber die Nichtlinearitäten sind nur in den zusätzlichen Termen niedriger Ordnung – der führende Teil der PDGL $\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$, der die „Parabolizität“ der PDGL bestimmt, bleibt unverändert und linear – solche PDGLen heißen quasi-linear.

Wir werden jetzt voll- und semi-diskretisierten Differenzenmethoden für diese PDGLen herleiten.

Gitter $(x_i, t_j) = (ih, jk), \quad i = 0, 1, \ldots, N := 1/h, \quad j = 0, 1, \ldots, K := T/k$

Zusätzliche Terme, die nicht von $\dfrac{\partial u}{\partial x}$ abhängen, d.h. $F(t, x, u)$ sind leicht zu behandeln.

9.4.1 Volldiskretisierung

Addiere $f(t_j, x_i, u(x_i, t_j)) \simeq f(jk, ih, u_{i,j})$ zu dem i-ten Komponenten der rechten Seite

$$\Rightarrow \qquad \boxed{\begin{array}{rcll} u_{i,j+1} & = & u_{i,j} & + \quad r(u_{i+1,j} - 2u_{ij} + u_{i-1,j}) \\ & & & + \quad kf(jk, ih, u_{i,j}) \end{array}}$$

9.4.2 Linienmethode

addiere $f(t, x_i, u(x_i t)) \simeq f(t, ih, u_i(t))$

$$\Rightarrow \qquad \boxed{\begin{array}{rcl} \dfrac{d}{dt}\, u_i(t) & = & \dfrac{1}{h^2}\,(u_{i+1}(t) - 2u_i(t) + u_{i-1}(t)) \\ & & + \, f(t, ih, u_i(t)) \end{array}}$$

Bemerkung Der zusätzliche Term ist nur "lokal" bestimmt, d.h. der i-te Komponent hängt nur den i-ten Komponenten x und u ab (x_i, u_i).

Beispiel $\quad f(t, x, u) = u(1 - u)$

addiere $\quad$ VD $\quad \ldots + ku_{i,j}(1 - u_{i,j}), \qquad$ LM $\quad \ldots + u_i(t)(1 - u_i(t))$

Hängt der zusätzliche Term von $\dfrac{\partial u}{\partial x}$ ab, dann müssen wir eine geeignete Differenzenquotientenformel für $\dfrac{\partial u}{\partial x}$ benutzen, um die räumliche Ordnung $O(h^2)$ des Hauptteils zu behalten. Die richtige Formel hängt von der Form der Funktion $f\left(t, x, u\,\dfrac{\partial u}{\partial x}\right)$ ab.

BEISPIEL $\boxed{f\left(t, x, u\,\dfrac{\partial u}{\partial x}\right) = u\,\dfrac{\partial u}{\partial x}}$ $\Rightarrow$ Burgers-Gleichung

Der zentralisierte Differenzenquotient für $\dfrac{\partial u}{\partial x}$ besitzt Ordnung $O(h^2)$.

9.4.3 Volldiskretisierung der Burgers-Gleichung

$$
\begin{aligned}
u(x_i,t_j)\,\frac{\partial u}{\partial x}\,(x_i,t_j) &\rightarrow u(x_i,t_j)\left\{\frac{u(x_i+h,t_j)-u(x_i-h,t_j)}{2h}\right\} + O(h^2)\\
&\rightarrow u(x_i,t_j)\left\{\frac{u(x_{i+1},t_j)-u(x_{i-1},t_j)}{2h}\right\} + O(h^2)\\
&\rightarrow u_{i,j}\left\{\frac{u_{i+1,j}-u_{i-1,j}}{2h}\right\}.
\end{aligned}
$$

Die volldiskretisierte Differenzenmethode lautet

$$
\boxed{\begin{aligned}
u_{i,j+1} &= u_{i,j} + r(u_{i+1,j}-2u_{i,j}+u_{i-1,j})\\
&\quad + \frac{k}{2h}\,u_{i,j}(u_{i+1,j}-u_{i-1,j})
\end{aligned}}
$$

$r = k/h^2 \;\Rightarrow\; k/h = rh$ klein, weil $r \le 1/2$ für Stabilität ist.

9.4.4 Linienmethode für die Burgers-Gleichung

$$
\begin{aligned}
u(x_i,t)\,\frac{\partial u}{\partial x}\,(x_i,t) &\rightarrow u(x_i,t)\left\{\frac{u(x_i+h,t)-u(x_i-h,t)}{2h}\right\} + O(h^2)\\
&\rightarrow u(x_i,t)\left\{\frac{u(x_{i+1},t)-u(x_{i-1},t}{2h}\right\} + O(h^2)\\
&\rightarrow U_i(t)\left\{\frac{U_{i+1}(t)-U_{i-1}(t)}{2h}\right\}
\end{aligned}
$$

Die Linienmethode lautet

$$
\boxed{\begin{aligned}
\frac{d}{dt}\,U_i(t) &= \frac{1}{h^2}\,[U_{i+1}(t)-2U_i(t)+U_{i-1}(t)]\\
&\quad + \frac{1}{2h}\,U_i(t)\,[U_{i+1}(t)-U_{i-1}(t)]
\end{aligned}}
$$

Wir haben Dirichlet-RBen hier $\Rightarrow \quad U_0(t) = U_N(t) \equiv 0$ for all $t \geq 0$.

Definiere $U = \begin{pmatrix} U_1 \\ \vdots \\ U_{N-1} \end{pmatrix} \in \mathbb{R}^{N-1}$,

$$T = \begin{bmatrix} -2 & 1 & & & & \bigcirc \\ 1 & -2 & 1 & & & \\ & 1 & -2 & \ddots & & \\ & & 1 & \ddots & 1 & \\ & & & \ddots & -2 & 1 \\ \bigcirc & & & & 1 & -2 \end{bmatrix} \qquad (N-1) \times (N-1) \text{ 3-bändig}$$

und eine Abbildung $F: \mathbb{R}^{N-1} \to \mathbb{R}^{N-1}$ durch

$$F(U) = \begin{pmatrix} F_1(U_1, \ldots, U_{N-1}) \\ \vdots \\ F_{N-1}(U_1, \ldots, U_{N-1}) \end{pmatrix} \text{ mit } \boxed{F_i(U_1, \ldots, U_{N-1}) = U_i(U_{i+1} - U_{i-1})}$$

Die vektorwertige Version des obigen Systems von gewöhnlichen DGLen lautet

$$\boxed{\frac{d}{dt}\, U(t) = \frac{1}{h^2}\, T\, U(t) + \frac{1}{2h}\, F(U(t))}$$

Die entsprechende DGL der Linienmethode für die Reaktion-Diffusions-PDGL

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + u(1-u)$$

mit Dirichlet-RBen (wie oben!) lautet

$$\boxed{\frac{d}{dt}\, U(t) = \frac{1}{h^2}\, T\, U(t) + G(U(t))}$$

wobei $G: \mathbb{R}^{N-1} \to \mathbb{R}^{N-1}$ ist definiert durch

$$G(U) = \begin{pmatrix} G_1(U_1, \ldots, U_{N-1}) \\ \vdots \\ G_{N-1}(U_1, \ldots, U_{N-1}) \end{pmatrix} \quad \text{mit} \quad \boxed{G_i(U_1, \ldots, U_{N-1}) = U_i(1 - U_i)}$$

ABER, wie in dem linearen Fall, sind explizite numerische Verfahren (z.B. Euler, Heun) für die obigen Daten nur numerisch stabil, wenn die Zeit-Schrittweite sehr klein ist: $r = k/h^2 \leq 1/2$.

$\Rightarrow$ Wir sollen ein implizites Verfahren benutzen!

9.4.5 Die Crank-Nicolson-Methode

Die Crank-Nicolson-Methode ist tatsächlich nur das Trapez-Verfahren für die gewöhnliche DGL in der Linienmethode. Aber jetzt haben wir eine nichtlineare DGL der Form

$$\boxed{\frac{d}{dt}\, U(t) = \frac{1}{h^2}\, TU(t) + N(U(t))}$$

wobei

$$N(u) = \begin{cases} \dfrac{1}{2h}\, F(u) & \text{Burgers-Gl.} \\ G(u) & \text{Reaktion-Diffusions-DGL} \end{cases}$$

Hier lautet die Trapez-Regel $V_j \simeq U(t_j))$

$$V_{j+1} = V_j + \frac{1}{2}\, k \left[\frac{1}{h^2}\, T\, V_j + N(V_j)\right] + \frac{1}{2}\, k \left[\frac{1}{h^2}\, T\, V_{j+1} + N(V_{j+1})\right]$$

$\Rightarrow$ eine nichtlineare Gleichung von Dimension $N - 1$

Schreibe mit $r = k/h^2$ um $\Rightarrow$

$$\underbrace{(2I - rT)V_{j+1} - kN(V_{j+1})}_{\text{wir wollen } V_{j+1} \text{ finden}} = \underbrace{(2I + rT)V_j + kN(V_j)}_{\text{bekannt}}$$

$\Rightarrow$ Newtons-Methode!

Aber $N \gg 0$. Trotzdem nicht hoffnungslos, weil die Funktion

$$H(V) = (2I - rT)V - kN(V) - \{(2I + rT)V_j + kN(V_j)\}\,,$$

deren Nullstelle wir berechnen wollen, eine ziemlich einfach bändige Jacobi-Matrix

$$\nabla H = \left[\frac{\partial H_i}{\partial V_j}\right]$$

besitzt

aber dennoch sehr aufwendig!

9.5 Linearimplizite Verfahren

Betrachte eine gewöhnliche DGL der Form

$$\frac{dx}{dt} = Ax + n(x),$$

d.h. mit einem nichttrivial linearen Teil Ax und einem nichtlinearen Teil $n(x)$.

Ein Vorschlag von Rosenbrock: konstruiere implizite Verfahren, für welche nur der lineare Teil der DGL impliziert ist.

BEISPIELE

(1) linearimplizites Euler-Verfahren

$$x_{n+1} = x_n + k\,[A\,x_{n+1} + n(x_n)] \qquad \Rightarrow \qquad [I - kA]x_{n+1} = x_n + kn(x_n)$$

lineares System! Besser noch: die selbe Matrix in dem linearen System für jedes n!

(2) Linearimplizites Trapez-Verfahren

$$x_{n+1} = x_n + \frac{1}{2}\,k\,[(Ax_n + n(x_n)) + (Ax_{n+1} + n(x_n))]$$

$$\Rightarrow \qquad [2I - kA]x_{n+1} = [2I + kA]x_n + 2kn(x_n)$$

Solche Verfahren erhalten die Ordnung des ursprünglichen Verfahrens, sind stabiler als die expliziten Versionen und nicht so aufwendig als die vollimplizierten Versionen.

$\Rightarrow$ Die linearimplizite Crank-Nicolson-Methode für die obige nichtlineare PDGL lautet

$$V_{j+1} = V_j + \frac{1}{2}\,k\left[\frac{1}{h^2}\,TV_j + N(V_j)\right] + \frac{1}{2}\,k\left[\frac{1}{h^2}\,TV_{j+1} + N(V_j)\right]$$

$$\Rightarrow \quad \boxed{(2I - rT)V_{j+1} = (2I + rT)V_j + 2kN(V_j)}$$

Viel leichter als die Newton-Methode! Insbesondere, weil die Matrix $2I - rT$ hier eine Bandmatrix ist.

Kapitel 10

Differenzenmethoden in 2 räumlichen Dimensionen

10.1 Elliptische PDGLen: Poisson-Gleichung

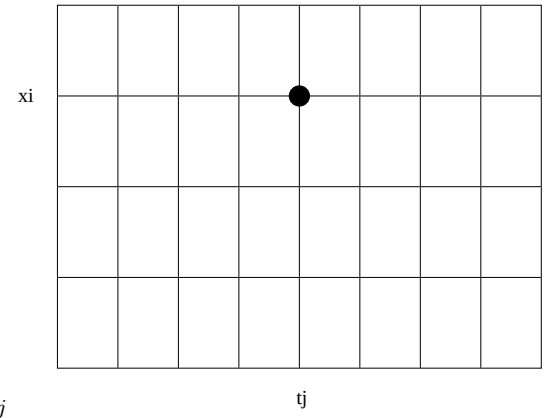
Betrachte die Randwertaufgabe der Poisson-Gleichung, eine typische elliptische PDGL:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f, \qquad 0 \le x \le a,\ 0 \le y \le b$$

mit Dirichlet-Randbedeingungen

$$\begin{cases} u(0,y) = u(a,y) = 0, & 0 \le y \le b \\ u(x,0) = u(x,b) = 0, & 0 \le x \le a \end{cases}$$

und das gleichmäßige Gitter $(x_i, y_j) = (i\Delta x, j\Delta y)$



$$i = 0, 1, \ldots, \; N := a/\Delta x, \quad j = 0, 1, \ldots, \; M := b/\Delta y$$

Die zentralisierten Differenzenquotienten für die obigen partiellen Ableitungen lauten

$$\frac{\partial^2 u}{\partial x^2}\,(x_i, y_j) = \frac{u(x_i + \Delta x, y_j) - 2u(x_i, y_j) + u(x_i - \Delta x, y_j)}{(\Delta x)^2} + O((\Delta x)^2)$$

$$\frac{\partial^2 u}{\partial y^2}\,(x_i, y_j) = \frac{u(x_i, y_j + \Delta y) - 2u(x_i, y_j) + u(x_i, y_j - \Delta y)}{(\Delta y)^2} + O((\Delta y)^2)$$

Schreibe $f_{i,j} = f(x_i, y_j)$, ersetze $u(x_i, y_j)$ durch $U_{i,j}$ und werfe den Fehlerterm weg $\Rightarrow$

$$\frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{(\Delta x)^2} + \frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{(\Delta y)^2} = f_{i,j}$$

Wir werden nur den Fall $\Delta x = \Delta y = h$ betrachten.

Dann haben wir die 5-Punkte-Relation

$$\boxed{U_{i+1,j} + U_{i-1,j} + U_{i,j+1} + U_{i,j-1} - 4U_{i,j} = h^2\, f_{i,j}}$$

mit den Randwerten

$$\begin{cases} U_{0,j} = U_{N,j} = 0, & 0 \le j \le M \\ U_{i,0} = U_{i,M} = 0, & 0 \le i \le N \end{cases}$$

Wir können diese Relationen als ein lineares System darstellen

$$\boxed{AU = f}$$

wobei U, $f \in \mathbb{R}^{N-1)(M-1)}$ definiert durch

$$U = \begin{pmatrix} U_{1,1} \\ U_{1,2} \\ \vdots \\ U_{1,M-1} \\ U_{2,1} \\ \vdots \\ U_{N-1,M-1} \end{pmatrix}, \qquad f = \begin{pmatrix} f_{1,1} \\ f_{1,2} \\ \vdots \\ f_{1,M-1} \\ f_{2,1} \\ \vdots \\ f_{N-1,M-1} \end{pmatrix}$$

sind und A die folgende $(N-1)(M-1) \times (N-1)(M-1)$ Matrix ist

$$A = \begin{bmatrix} -B & I & & & & & 0 \\ I & -B & I & & & & \\ & I & -B & \ddots & & & \\ & & I & \ddots & I & & \\ & & & \ddots & -B & I & \\ & & & & I & -B & I \\ 0 & & & & & I & -B \end{bmatrix}$$

mit der $(M-1) \times (M-1)$ Identitäts-Matrix I und der 3-bändige $(M-1) \times (M-1)$ Matrix

$$B = \begin{bmatrix} 4 & -1 & & & & & 0 \\ -1 & 4 & -1 & & & & \\ & -1 & 4 & \ddots & & & \\ & & -1 & \ddots & -1 & & \\ & & & \ddots & 4 & -1 & \\ & & & & -1 & 4 & -1 \\ 0 & & & & & -1 & 4 \end{bmatrix}$$

d.h., die Matrix A ist ist 5-bändig. Sie ist invertierbar, aber riesig, z.B. $(N-1)(M-1) = 10^3 \times 10^3 = 10^6 \quad \Rightarrow \quad$ eine $10^6 \times 10^6$ Matrix!

Gauß-Elimination ist unpraktisch $\Rightarrow$ wir müssen eine Iterationsmethode verwenden. z.B.

Jacobi-Iterationen

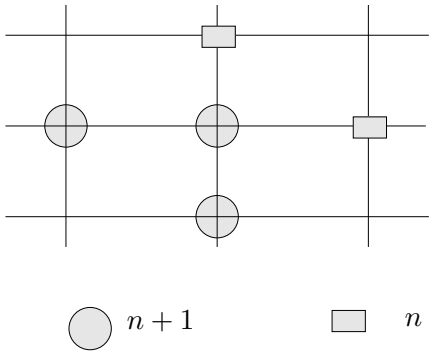
$$U_{i,j}^{(n+1)} = \frac{1}{4} \left[U_{i+1,j}^{(n)} + U_{i-1,j}^{(n)} + U_{i,j+1}^{(n)} + U_{i,j-1}^{(n)} - h^2 f_{i,j} \right]$$

Gauß-Seidel-Iterationen (hier Liebmann-Iterationen genannt).

$$U_{i,j}^{(n+1)} = \frac{1}{4} \left[U_{i-1,j}^{(n+1)} + U_{i,j-1}^{(n+1)} + U_{i+1,j}^{(n)} + U_{i,j+1}^{(n)} - h^2 f_{i,j} \right]$$

Oder eine relaxierte Methode, z.B. eine SOR-Methode, usw.

Wir fangen mit $(i,j) = (1,1)$ an!



Bemerkung Die obige 5-bändige Matrix A ist

- diagonaldominant
- symmetrisch
- positivdefinit

Sehr nett! Aber die Matrix ist extrem groß!

10.2 Die 2-dimensionale Wärme-Gleichung

Betrachte jetzt die Anfangsrandwertaufgabe der 2-dimensionalen Wärme-Gleichung

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}, \qquad 0 \le x \le a, \;\; 0 \le y \le b,$$

mit Dirichlet-Randbedingungen

$$u(0,y,t) = u(a,y,t) = 0 \qquad 0 \le y \le b, \; 0 \le t \le T$$

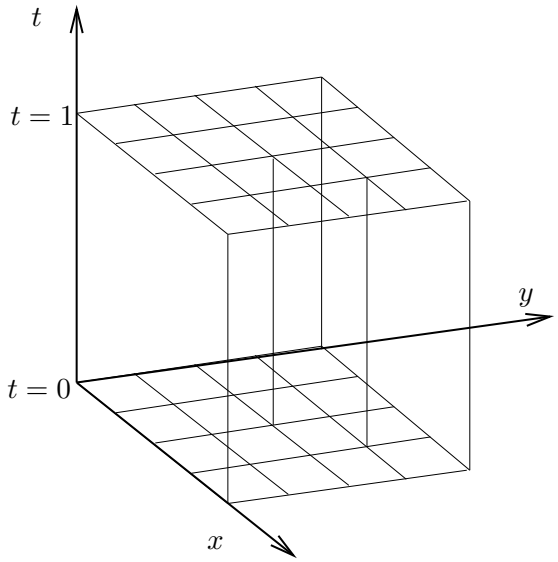
$$u(x,0,t) = u(x,b,t) = 0 \qquad 0 \le x \le a, \; 0 \le t \le T$$

und Anfangswert

$$u(x,y,0) = u_0(x,y), \qquad 0 \le x \le a, \; 0 \le y \le b.$$

Wir brauchen jetzt ein 3-dimensionales Gitter

$$(x_i, y_i, t_n) = (ih, jh, nk)$$



wobei

$$i = 0, 1, \ldots, N := a/h$$

$$j = 0, 1, \ldots, M := b/h$$

$$n = 0, 1, \ldots, K := T/k$$

Die volldiskretisierte Differenzenmethode hier lautet

$$U_{i,j,n+1} = U_{i,j,n} + r\left[U_{i+1,j,n} + U_{i-1,j,n} + U_{i,j+1,n} + U_{i,j-1,n} - 4U_{i,j,n}\right]$$

wobei $r = k/h^2$, $U_{i,j,n} \simeq u(x_i, x_i, t_n)$, usw.

Mit der Vektor-Matrix-Notation des obigen elliptischen Beispiels erhalten wir die vektorwertige lineare Differenzengleichung

$$U_{n+1} = [I + rA]\, U_n, \qquad n = 0, 1, \ldots$$

wobei

$$U_n = \begin{pmatrix} U_{1,1,n} \\ \vdots \\ U_{1,M-1,n} \\ U_{2,1,n} \\ \vdots \\ U_{N-1,M-1,n} \end{pmatrix} \in \mathbb{R}^{(N-1)(M-1)}$$

Für numerische Stabilität $\Rightarrow$ $|\lambda(I+rA)| \leq 1 \Rightarrow r \leq \frac{1}{2}$ nochmal.

Die Crank-Nicolson-Methode ist absolut stabil hier auch.

Jetzt fängt alles an!

Kapitel 11

Finite element and Galerkin methods

The finite element and Galerkin methods are alternatives to finite difference methods for discretising partial differential equations. They are both based on the idea of the generalised Fourier series expansions which we are often able to derive for special linear PDE, with the major difference being that for nonlinear PDE we have to determine the coefficients numerically. A major advantage of these methods in practice, especially for the finite element method, is that irregularly shaped domains can be easily handled.

11.1 The Galerkin method

Let us consider the Poisson equation on a bounded domain Ω in $\mathbb{R}^d$ with a smooth boundary, i.e.

$$\Delta u = g, \qquad u\Big|_{\partial\Omega} = 0, \tag{11.1}$$

where $g : \Omega \to \mathbb{R}$.

We will use the the function space $L_2(\Omega)$ of square-integrable functions $f : \Omega \to \mathbb{R}$, which contains the solutions of the PDE (11.1). This is a Hilbert space with the inner product and norm

$$< f, g >= \int_\Omega f(x)g(x)\,dx, \qquad \|f\| = \sqrt{< f, f >} = \sqrt{\int_\Omega f(x)^2\,dx},.$$

Let $\{\phi_1, \phi_2, \ldots, \phi_n, \ldots\}$ be an orthonormal basis of $L_2(\Omega)$. In particular,

$$< \phi_i, \phi_j >= \delta_{i,j} \qquad \text{(Kronecker delta)}$$

and every function $f \in L_2(\Omega)$ has the unique representation

$$f = \sum_{k=1}^{\infty} c_k^{(f)} \phi_k \qquad \text{with} \quad c_k^{(f)} =< f, \phi_k > .$$

Moreover, $f^{(N)} := \sum_{k=1}^{N} c_k^{(f)} \phi_k$ is the best ("least-squares") approximation for f in the finite dimensional subspace $\mathcal{X}_N := \text{span}\{\phi_1, \phi_2, \ldots, \phi_N\}$ of $L_2(\Omega)$ spanned by the first N elements of the orthonormal basis, i.e.

$$\|f^{(N)} - f\|^2 \leq \|g^{(N)} - f\|^2, \qquad \forall g^{(N)} : \mathcal{X}_N \to \mathcal{X}_N.$$

Let us write P_N for the projection of the space $L_2(\Omega)$ onto the subspace $\mathcal{X}_N$, so $P_N f = f^{(N)}$.

It is known that the normalised eigenfunctions and eigenvalues of the negative Laplacian operator $-\Delta$ on Ω with the Dirichlet boundary condition, i.e.

$$-\Delta \phi_k = \lambda_k \phi_k, \qquad k = 1, 2, 3, \ldots, \tag{11.2}$$

form an orthonormal basis of the function space $L_2(\Omega)$. It is a convention that the eigenvalues are positive, so we need $-\Delta$ rather than Δ since on integrating by parts

$$< -\Delta \phi_k, \phi_k > =< \nabla \phi_k, \nabla \phi_k > = \|\nabla \phi_k\|^2,$$

whereas from (11.2) we have

$$< -\Delta \phi_k, \phi_k > =< \lambda_k \phi_k, \phi_k > = \lambda_k < \phi_k, \phi_k > = \lambda_k$$

Let u be the solution of the Poisson equation (11.1). Then $u^{(N)} = P_N u$ is the solution of the projected problem

$$P_N \Delta u_N \; = \; P_N g, \qquad u_N \in \mathcal{X}_N, \tag{11.3}$$

in the finite dimensional subspace $\mathcal{X}_N$. This is due to the linearity of the problem. Note that (11.3) is equivalent to the following linear algebraic equation

$$A_N c_N = g_N, \qquad c_N \in \mathbb{R}^N, \tag{11.4}$$

where $g_N \in \mathbb{R}^N$ is defined componentwise by $g_{N,j} := < g, \phi_j >$ for $j = 1, 2,$

..., N, and

$$A_N = \begin{bmatrix} -\lambda_1 & 0 & & & & \bigcirc \\ 0 & -\lambda_2 & 0 & & & \\ & 0 & -\lambda_3 & \ddots & & \\ & & 0 & \ddots & 0 & \\ & & & \ddots & -\lambda_{N-1} & 0 \\ \bigcirc & & & & 0 & -\lambda_N \end{bmatrix} \quad N \times N \text{ diagonal matrix}$$

since

$$\begin{aligned} (A_N c_N)_j &= \langle P_N \Delta u_N \, , \, \phi_j \rangle \\ &= \left\langle P_N \Delta \sum_{k=1}^{N} c_{N,k} \phi_k \, , \, \phi_j \right\rangle \qquad \text{with} \quad u_N = \sum_{k=1}^{N} c_{N,k} \phi_k \\ &= \sum_{k=1}^{N} c_{N,k} \langle \Delta \phi_k \, , \, \phi_j \rangle \\ &= \sum_{k=1}^{N} c_{N,k} \langle -\lambda_k \phi_k \, , \, \phi_j \rangle \\ &= -\sum_{k=1}^{N} \lambda_k c_{N,k} \langle \phi_k \, , \, \phi_j \rangle = -\lambda_j c_{N,j} \end{aligned}$$

for $j = 1, 2, \ldots, N$, so the (i,j)th component of A_N is $-\lambda_i \delta_{i,j}$ (Kronecker delta).

It is not hard to see that the solution of (11.4) is given componentwise by $c_{N,j} := < c, \phi_j > = c_j^{(f)}$ for $j = 1, 2, \ldots, N$, i.e. the coefficients of the solution to the original problem (11.1).

We have not gained much from the above considerations for the linear problem (11.1), but serves as an introduction for nonlinear problems where the real benefits are to be found. Consider now the nonlinear PDE

$$\Delta u + f(u) = g, \qquad u\Big|_{\partial\Omega} = 0, \tag{11.5}$$

where $f : \mathbb{R} \to \mathbb{R}$ is nonlinear and $g : \Omega \to \mathbb{R}$. In this case the projection $P_n u$ onto $\mathcal{X}_N$ of the solution u is <u>not</u> a solution of the equations projected onto $\mathcal{X}_N$, i.e.

$$P_N \Delta v_N + P_N f(v_N) = P_N g, \qquad v_N \in \mathcal{X}_N, \tag{11.6}$$

which we can rewrite as the nonlinear algebraic equation

$$A_N c_N + F_N(c_N) = g_N, \qquad c_N \in \mathbb{R}^N, \tag{11.7}$$

where $F_N : \mathbb{R}^N \to \mathbb{R}^N$ is defined componentwise by

$$F_{N,j}(c_N) = \left\langle f\left(\sum_{k=1}^{N} c_{N,k}\phi_k\right), \phi_j \right\rangle, \qquad j = 1, 2, \ldots, N.$$

We can solve this equation using Newton's method.

Then $v_N := \sum_{j=1}^{N} c_{N,j}\phi_j$ is a solution of (11.6). In general, $v_N \neq P_N u$, the projection of the solution u of PDE (11.5) onto the subspace $\mathcal{X}_N$. Nevertheless, we can use v_N as an approximation for u. This method is known as the Galerkin method and v_N is called a Galerkin approximation of u.

The Galerkin method can also be applied to a PDE involving time derivatives such as the following nonlinear parabolic problem

$$\frac{\partial u}{\partial t} = \Delta u + f(u) + g, \qquad u\Big|_{\partial\Omega} = 0. \tag{11.8}$$

The solution coefficients now depend on time and the corresponding Galerkin equation is an N-dimensional ODE

$$\frac{dc_N}{dt} = A_N c_N + F_N(c_N) + g_N, \qquad c_N : [0, T] \to \mathbb{R}^N. \tag{11.9}$$

This ODE needs an implicit numerical method because the coefficients of the matrix A_N are very large for large N since $\lambda_N \to \infty$ as $N \to \infty$ (we say that the ODE is "stiff"). In fact, a linear-implicit method is ideal here since the stiffness is concentrated in the matrix A_N.

A major difficulty in using the Galerkin method is that it requires explicit knowledge of the eigenfunctions and eigenvalues of the Laplace operator on the given domain. This information is usually only available in very special cases with simple geometry.

11.2 The finite element method

The finite element method circumvents the problem of the Galerkin method in having to know explicitly the orthonormal basis of the eigenfunctions.

Instead it decomposes the domain Ω into a finite number of simpler regions or elements $\mathcal{D}_0$, ..., $\mathcal{D}_{N-1}$ such as intervals (in $\mathbb{R}^1$) or triangles (in $\mathbb{R}^2$) and uses a basis of functions which are only nonzero on all but a few adjacent subregions.

We will illustrate this in terms of of a one-dimensional domain $\Omega = [x_0, x_N]$, which we decompose into N subintervals $\mathcal{D}_j = [x_j, x_{j+1}]$, $j = 0, 1, \ldots, N-1$, where $x_0 < x_1 < \ldots < x_{j-1} < x_j < \ldots < x_N$, so

$$\Omega = \bigcup_{j=1}^{N-1} \mathcal{D}_j \qquad \Leftrightarrow \qquad [x_0, x_N] = \bigcup_{j=1}^{N-1} [x_j, x_{j+1}].$$

As the basis functions we will use the hat functions $\phi_j : [x_0, x_N] \to \mathbb{R}^1$ which are defined as follows:

• for the interior functions, i.e. with $1 \le j \le N-1$ set

$$\phi_j(x) := \begin{cases} \dfrac{x - x_{j-1}}{x_j - x_{j-1}} & \text{for } x_{j-1} \le x < x_j, \\ \dfrac{x_{j+1} - x}{x_{j+1} - x_j} & \text{for } x_j \le x < x_{j+1}, \\ 0 & \text{elsewhere} \end{cases}$$

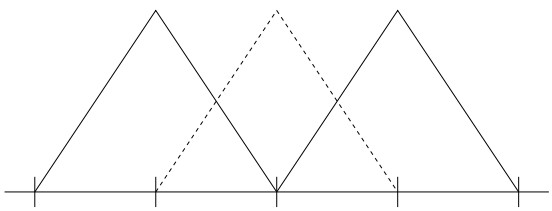
• for the boundary functions, with $j = 0$ and $j = N$ set

$$\phi_0(x) := \begin{cases} \dfrac{x_1 - x}{x_1 - x_0} & \text{for } x_0 \le x < x_1, \\ 0 & \text{elsewhere} \end{cases}$$

and

$$\phi_N(x) := \begin{cases} \dfrac{x - x_{N-1}}{x_N - x_{N-1}} & \text{for } x_{N-1} \le x < x_N \\ 0 & \text{elsewhere} \end{cases}$$

In future we shall write $h_j := x_{j+1} - x_j > 0$ for the length of the jth subinterval $[x_j, x_{j+1}]$.



The hat functions are thus continuous piecewise linear or polygonal functions. Note that they are differentiable except at a finite number of points with the derivatives taking values $\pm 1/h_j$ or 0. Their second derivatives are thus identically zero at the points where they exist. They thus appear to be useless as analogues of the eigenfunctions of the negative Laplacian operator in equation (11.2). However we never really said what we mean by the solution of a PDE such as the Poisson equation

$$\Delta u = g, \qquad u\Big|_{\partial\Omega} = 0. \tag{11.10}$$

The implication above was that we were using a classical solution, for which all the derviatives in the equation exist, i.e. the solution is at least two times continuously differentiable. Instead we shall use the concept of a weak solution.

First we multiply (11.10) by a continuously differentiable function $\psi : \Omega \to \mathbb{R}$ which vanishes on the boundary $\partial\Omega$. Then, after integrating over Ω and using integration by parts we obtain

$$\int_\Omega g(x)\psi(x)\,dx = \int_\Omega \Delta u(x)\psi(x)\,dx = -\int_\Omega \nabla u(x)^T \nabla\psi(x)\,dx,$$

that is,

$$\langle \nabla u, \nabla\psi\rangle + \langle g, \psi\rangle = 0. \tag{11.11}$$

Let us introduce the subspace $H_0^1(\Omega)$ of $L_2(\Omega)$ consisting of square-integrable functions $u : \Omega \to \mathbb{R}$ which vanish of the boundary $\partial\Omega$ and are differentiable almost every where with square-integrable first order partial derviatives.

Definition *A function $u \in H_0^1(\Omega)$ is called a weak solution of the Poisson equation (11.10) if it satisfies (11.11) for all $\psi \in H_0^1(\Omega)$.*

We will use the method of finite elements to find approximations for weak solutions of the Poisson equation and similar boundary value problems for elliptic PDE as well as initial boundary value problems for parabolic PDE.

11.2.1 Properties of hat functions

The hat functions belong to the the space $\mathcal{P}_{x_0,\ldots x_N}$ consisting of the piecewise linear continuous functions which are linear on each of the subintervals $[x_j, x_{j+1}]$ for $j = 0,\ 1,\ \ldots,\ N-1$, i.e. the continuous piecewise linear or

polygonal functions which can change direction only at the points x_j. We assume that N is an odd number, so $N+1$ is even.

Property 1 The hat functions $\{\phi_0, \phi_1, \ldots, \phi_N\}$ are a basis for $\mathcal{P}_{x_0,\ldots x_N}$.

Hence every function $f \in \mathcal{P}_{x_0,\ldots x_N}$ has the unique representation

$$f(x) = \sum_{k=0}^{N} c_k^{(f)} \phi_k(x).$$

Property 2 Only ϕ_j and ϕ_{j+1} are nonzero on $[x_j, x_{j+1}]$. Hence

$$\phi_i(x)\phi_j(x) = 0 \qquad \text{for} \quad |i-j| > 1.$$

Property 3 A simple approximation formula for the integral $\int_{x_0}^{x_N} g(x)\phi_j(x)\,dx$ can be derived as follows. Substitute g by $g_P \in \mathcal{P}_{x_0,\ldots x_N}$ defined by

$$g_P(x) := \sum_{i=0}^{N} g(x_i)\phi_i(x)$$

and note that

$$\begin{aligned} I_j := \int_{x_0}^{x_N} g_P(x)\phi_j(x)\,dx &= \int_{x_0}^{x_N} \sum_{i=0}^{N} g(x_i)\phi_i(x)\phi_j(x)\,dx \\ &= \sum_{i=0}^{N} g(x_i) \underbrace{\int_{x_0}^{x_N} \phi_i(x)\phi_j(x)\,dx}_{=:b_{i,j}} \end{aligned}$$

for $j = 0, 1, \ldots, N$. Then $B = [b_{i,j}]$ is a symmetric $(N+1)\times(N+1)$ matrix and

$$\mathbf{I} = B\,\mathbf{g}$$

where

$$\mathbf{I} = \begin{pmatrix} I_0 \\ I_1 \\ \vdots \\ I_N \end{pmatrix}, \qquad \mathbf{g} = \begin{pmatrix} g(x_0) \\ g(x_1) \\ \vdots \\ g(x_N) \end{pmatrix}.$$

The matrix B here is tri-diagonal with the contribution of the nonvanishing integration over the jth subinterval given by the 2×2 block

$$B_j = \int_{x_j}^{x_{j+1}} \begin{bmatrix} \phi_j(x)^2 & \phi_j(x)\phi_{j+1}(x) \\ \phi_{j+1}(x)\phi_j(x) & \phi_{j+1}(x)^2 \end{bmatrix} dx$$

$$= \frac{1}{(x_{j+1}-x_j)^2}\int_{x_j}^{x_{j+1}} \begin{bmatrix} (x_{j+1}-x)^2 & (x_{j+1}-x)(x-x_j) \\ (x-x_j)(x_{j+1}-x) & (x-x_j)^2 \end{bmatrix} dx$$

$$= \frac{h_j}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \qquad \text{where} \quad h_j = x_{j+1}-x_j.$$

The matrix B is formed by overlapping these blocks along the diagonal, in effect adding the diagonal terms, to obtain

$$B = \frac{1}{6}\begin{bmatrix} 2h_0 & h_0 & 0 & 0 & \dots \\ h_0 & 2h_0+2h_1 & h_1 & 0 & \dots \\ 0 & h_1 & 2h_1+2h_2 & h_2 & \dots \\ \ddots & \ddots & \ddots & \ddots & \end{bmatrix}$$

B is often called the element-mass matrix.

Property 4 Similarly we can evaluate the element-stiffness matrix A which is also an $(N+1)\times(N+1)$ tri-diagonal symmetric matrix formed in the same way as the matrix B in terms of the 2×2 overlapping blocks

$$A_j = \int_{x_j}^{x_{j+1}} \begin{bmatrix} \phi_j'(x)^2 & \phi_j'(x)\phi_{j+1}'(x) \\ \phi_{j+1}'(x)\phi_j'(x) & \phi_{j+1}'(x)^2 \end{bmatrix} dx$$

$$= \frac{1}{h_j^2}\int_{x_j}^{x_{j+1}} \begin{bmatrix} (-1)^2 & (-1)(1) \\ (1)(-1) & (1)^2 \end{bmatrix} dx$$

$$= \frac{1}{h_j}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \qquad \text{where} \quad h_j = x_{j+1}-x_j.$$

Here ϕ_j' is the derivative of ϕ_j.

Thus, when the points x_j are equi-spaced, i.e. $h_j \equiv h$ for $j = 0, 1, \ldots,$

$N-1$, we have

$$A = \frac{1}{h}\begin{bmatrix} 1 & -1 & & & & & 0 \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & \ddots & & & \\ & & -1 & \ddots & -1 & & \\ & & & \ddots & 2 & -1 & \\ & & & & -1 & 2 & -1 \\ 0 & & & & & -1 & 1 \end{bmatrix}, \quad B = \frac{h}{6}\begin{bmatrix} 2 & 1 & & & & & 0 \\ 1 & 4 & 1 & & & & \\ & 1 & 4 & \ddots & & & \\ & & 1 & \ddots & 1 & & \\ & & & \ddots & 4 & 1 & \\ & & & & 1 & 4 & 1 \\ 0 & & & & & 1 & 2 \end{bmatrix}$$

Note that the coefficients of the matrix A will be very large for a small spacing h between the points. This is the reason for the name "stiffnessänd can cause numerical problems when solving algebraic systems involving the matrix.

11.2.2 The one-dimensional Poisson problem

For a one-dimensional domain $\Omega = [x_0, x_N]$ the Poisson problem (11.10) reduces to a boundary value problem for an ODE, namely

$$u''(x) = g(x), \qquad u(x_0) = u(x_N) = 0. \tag{11.12}$$

We do not really need to solve this simple problem numerically because we can easily integrate to obtain an explicit solution: First integrate forwards from x_0 to y

$$u'(y) = u'(x_0) + \int_{x_0}^{y} g(s)\,ds$$

and then integrate backwards from x_N to x

$$\begin{aligned} u(x) &= u(x_N) + \int_{x_N}^{x} \left[u'(x_0) + \int_{x_0}^{y} g(s)\,ds \right] dy \\ &= -(x_N - x)u'(x_0) - \int_{x}^{x_N} \int_{x_0}^{y} g(s)\,ds\,dy, \end{aligned} \tag{11.13}$$

since $u(x_N) = 0$. Now $u(x_0) = 0$ too, so taking $x = x_0$ we get

$$0 = -(x_N - x_0)u'(x_0) - \int_{x_0}^{x_N} \int_{x_0}^{y} g(s)\,ds\,dy,$$

which we can solve for

$$u'(x_0) = \frac{-1}{x_N - x_0} \int_{x_0}^{x_N} \int_{x_0}^{y} g(s)\,ds\,dy,$$

which we substitue into (11.13) to obtain the solution in explicit form

$$u(x) = \frac{x_N - x}{x_N - x_0} \int_{x_0}^{x_N} \int_{x_0}^{y} g(s)\, ds\, dy - \int_{x}^{x_N} \int_{x_0}^{y} g(s)\, ds\, dy.$$

However, we return to the numerical approach because it gives insight into what happens in more complicated situations. To apply the finite element method we first write the BVP (11.12) in the weak solution form (11.11)

$$\int_{x_0}^{x_N} u'(x)\psi'(x)\, dx + \int_{x_0}^{x_N} g(x)\psi(x)\, dx = 0. \tag{11.14}$$

We approximate the known function g and the unknown solution u in $H_0^1([x_0, x_N])$ by functions g_N and u_N in $\mathcal{P}_{x_0,\ldots x_N}$ with the representations

$$g_N(x) = \sum_{i=0}^{N} c_i^{(g_N)} \phi_i(x), \qquad u_N(x) = \sum_{i=0}^{N} c_{N,i} \phi_i(x),$$

and use each of the hat functions as the test function ψ (which is OK as we could also represent ψ in terms of the hat functions). From (11.14) we obtain the $N+1$ linear equations

$$\sum_{i=0}^{N} c_{N,i} \underbrace{\int_{x_0}^{x_N} \phi_i'(x)\phi_j'(x)\, dx}_{=:a_{i,j}} + \sum_{i=0}^{N} c_i^{(g_N)} \underbrace{\int_{x_0}^{x_N} \phi_i(x)\phi_j(x)\, dx}_{=:b_{i,j}} = 0$$

for $j = 0, 1, \ldots, N$. Since the matrices A and B are symmetric we can rewrite this in the matrix-vector form

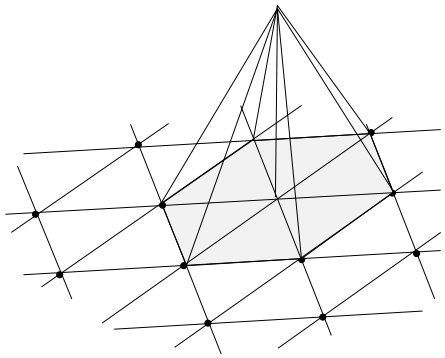
$$Ac_N + Bc^{(g_N)} = 0, \tag{11.15}$$

which has the explicit solution $c_N = -A^{-1}Bc^{(g_N)}$. However, it is better to solve the system of linear equations directly, which can be done very efficiently due to the tri-diagonal structure of the matrices — but one has to be careful with the small parameter h.

Remark: What has happened to the boundary condition in (11.12). These have been used in deriving in the weak form of the BVP (11.14) – the boundary terms in the integration by parts vanish. Since all of the hat functions except ϕ_0 and ϕ_N vanish at the boundary, the coefficients $c_{N,0}$ and $c_{N,N}$ of the latter in the hat function representation of the solution must be zero. Knowing this in advance, we could reduce the algebraic problem

(11.15) by two dimensions to $N-1$.

The counterpart of the hat functions in two dimensional domains are pyramidal in shape as in the figure.



11.2.3 The one-dimensional heat equation

We can also solve IBVP for the heat equation using finite elements. Consider the 1-dimensional heat equation with external forcing

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + g(x) \qquad x_0 \le x \le x_N, \tag{11.16}$$

with Dirichlet boundary condition using finite elements. A weak solution now satisfies the equation

$$\frac{d}{dt}\langle u(t), \psi\rangle + \langle \nabla u, \nabla \psi\rangle - \langle g, \psi\rangle = 0.$$

We now look for an approximation $u_N(t)$ in $\mathcal{P}_{x_0,\ldots,x_N}$ of the form

$$u_N(t,x) = \sum_{i=0}^{N} c_{N,i}(t)\phi_i(x).$$

From (11.16) we obtain the $(N+1)$-dimensional system of linear ordinary differential equations

$$\sum_{i=0}^{N} \frac{d}{dt} c_{N,i}(t) \underbrace{\int_{x_0}^{x_N} \phi_i(x)\phi_j(x)\,dx}_{=:b_{i,j}} + \sum_{i=0}^{N} c_{N,i}(t) \underbrace{\int_{x_0}^{x_N} \phi_i'(x)\phi_j'(x)\,dx}_{=:a_{i,j}}$$

$$- \sum_{i=0}^{N} c_i^{(g_N)} \underbrace{\int_{x_0}^{x_N} \phi_i(x)\phi_j(x)\,dx}_{=:b_{i,j}} = 0$$

for $j = 0, 1, \ldots, N$. The matrices A and B are symmetric, so we can rewrite this as the vector-valued algebraic-differential equation

$$B\frac{d}{dt}c_N(t) + Ac_N - Bc^{(g_N)} = 0, \tag{11.17}$$

which we can rewrite as the vector-valued ODE

$$\frac{d}{dt}c_N(t) + B^{-1}Ac_N - c^{(g_N)} = 0, \tag{11.18}$$

since the matrix B is invertible.

Kapitel 12

Free boundary value problems for PDE

Free boundary value problems occur in many applications, an important one being with American options which terminate before the end of an agreed contract period if a certain barrier is attained. The difficult here is that it is not known in advance if or when this may occur. We will briefly indicate how such problems can be formulated and solved numerically in the simpler setting of boundary problems with obstacles for ordinary differential equations.

12.1 Obstacle problems

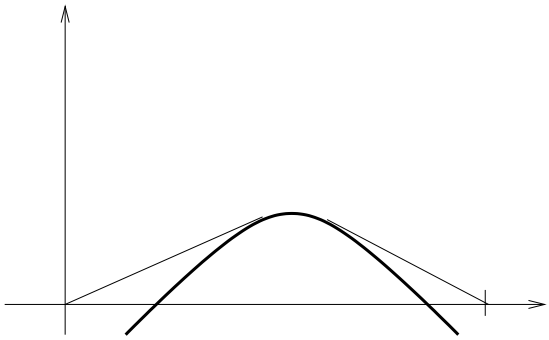
We consider a spatial domain $[-1, 1]$ and a two times continuously differentiable function $f : [-1, 1] \to \mathbb{R}$ with

- $f(-1) < 0$ and $f(1) < 0$
- $f''(x) < 0$ for $x \in [-1, 1]$
- $f(\bar{x}) > 0$ for some $\bar{x} \in (-1, 1)$.

The function f bounds from above a convex set which has a unique positive maximum in $(-1, 1)$. Moreover, its graph bounds from above a convex set which we will consider as an "obstacle" in the following situation. Let $u(x)$ denote a position on a curve which is fixed at the end points, i.e. with $u(-1) = u(1) = 0$, and lies above the obstacle defined by f. In particular, we want to find the curve of shortest length joining these end points that lies above

the obstacle. We can think of $u(x)$ as the displacement of a rubber band which is fixed the end points and passes over the obstacle.

It is fairly clear that the solution curve is given by straight lines from the endpoints which touch the obstacle tangentially together with the part of the obstacle between the points where the tangents lines touch it. That is, a straight lines joining $(-1,0)$ to $(\alpha, f(\alpha))$ and $(\beta, f(\beta))$ to $(1,0)$ in the plane together with the curve formed by the points $(x, f(x))$ for $x \in [\alpha, \beta]$, where $-1 < \alpha < \beta < 1$. However, the tangential points α and β, which represent the "free boundary points" that have to be determined.



We can formulate this problem mathematically as follows: Find a function $u \in C^2([-1,1], \mathbb{R})$ with $u(-1) = u(1) = 0$, and points α and β in $(-1,1)$ such that

$$\begin{cases} \text{for} \quad -1 < x < \alpha: & u''(x) = 0 & \text{hence} \quad u(x) > f(x) \\ \text{for} \quad \alpha < x < \beta: & u(x) = f(x) & \text{hence} \quad u''(x) = f''(x) < 0 \\ \text{for} \quad \beta < x < 1: & u''(x) = 0 & \text{hence} \quad u(x) > f(x) \end{cases}$$

which we can write in the complementary form

$$\begin{cases} \text{if} \quad u(x) > f(x), \quad \text{then} \quad u''(x) = 0 \\ \text{if} \quad u(x) = f(x), \quad \text{then} \quad u''(x) < 0 \end{cases}$$

Yet another way of writing this is: Find a function $u \in C^2([-1,1], \mathbb{R})$ with $u(-1) = u(1) = 0$ such that

$$u''(u - f) = 0, \quad -u'' \geq 0, \quad u - f \geq 0, \tag{12.1}$$

which is known as a linear complementarity problem. (We write $-u'' \geq 0$ here instead of the more obvious $u'' \leq 0$ for later convenience). This formulation does not mention the free boundary conditions at $x = \alpha$ and $x =$

β explicitly. These points can be read off from the solution when we have found one.

Actually we interpret the problem in term of a weak or variational solution which enables us to use a discertization method more easily to approximate it. Consider the following class of test or competing functions

$$\begin{aligned}\mathcal{K} \;:=\; & \{v \in C^0([-1,1],\mathbb{R}) \,:\, v(-1) = v(1) = 0, \\ & v(x) \geq u(x) \text{for} -1 \leq x \leq 1, v \,\text{piecewise in} C^1([-1,1],\mathbb{R})\}\end{aligned}$$

The conditions on u imply that $u \in \mathcal{K}$. Hence for $v \in \mathcal{K}$, we have $v - f \geq 0$ as well as $-u''(v-f) \geq 0$ (since $-u'' \geq 0$). Hence for all $v \in \mathcal{K}$the inequality

$$\int_{-1}^{1} -u''(v-f)\,dx \geq 0.$$

But from (12.1) we obtain

$$\int_{-1}^{1} -u''(u-f)\,dx \geq 0,$$

so subtracting gives

$$\int_{-1}^{1} -u''(v-u)\,dx \geq 0 \qquad \forall\, v \in \mathcal{K}.$$

Finally integrating by parts, using the vanishing boundary conditions of u and v we obtain

$$\int_{-1}^{1} u'(v-u)'\,dx \geq 0 \qquad \forall\, v \in \mathcal{K}. \tag{12.2}$$

This is called a variational inequality. We use (12.2) to define a weak solution $u \in \mathcal{K}$ of the obstacle problem.

12.2 Discretization of the obstacle problems

We consider a grid in $[-1,1]$ consisting of equally spaced points $x_i = -1 + i\Delta$, $i = 0, 1, \ldots, N$ with $\Delta = 2/N$. We define $f_i := f(x_i)$ and approximate $u''(x)$ by the central difference quotient

$$u''(x_i) \approx \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{\Delta^2}.$$

Thus with $w_i \approx u(x_i)$ we have a discretized version of the linear complementarity problem (12.1): Find $(w_0, w_1, \ldots, w_{N-1}, w_N)$ with $w_0 = w_N = 0$ such that

$$\left\{ \begin{array}{l} (w_{i-1} - 2w_i + w_{i+1})\,(w_i - f_i) = 0 \\ \\ w_{i-1} - 2w_i + w_{i+1} \geq 0, \quad w_i \geq f_i \end{array} \right\} \qquad i = 1, \ldots, N.$$

Let us introduce the $(N-1) \times (N-1)$ tridiagonal matrix

$$B = \begin{bmatrix} 2 & -1 & & & & \bigcirc \\ -1 & 2 & -1 & & & \\ & -1 & 2 & \ddots & & \\ & & -1 & \ddots & -1 & \\ & & & \ddots & 2 & -1 \\ \bigcirc & & & & -1 & 2 \end{bmatrix}$$

and the $(N-1)$-dimensional vectors

$$w = \begin{pmatrix} w_1 \\ \vdots \\ w_{N-1} \end{pmatrix}, \qquad f = \begin{pmatrix} f_1 \\ \vdots \\ f_{N-1} \end{pmatrix}.$$

Then the discretized linear complementarity problem for the weak solution (12.2) has the compact matrix-vector form

$$\left\{ \begin{array}{l} (w - f)^T\, Bw = 0 \\ \\ Bw \geq 0, \quad w \geq f \end{array} \right\} \tag{12.3}$$

where vector inequalities are interpreted componentwise.

To find a solution of (12.3) we solve $Bw = 0$ under the side condition $w \geq f$.

12.2.1 Cryer's SOR method

The SOR method is an iterative method for solving linear systems such as $Ax = \hat{b}$. Cryer introduced a projected version of the SOR method which takes into account side conditions. In particular, he wanted to solve the constrained linear problem

Cryer's problem

Find vectors x and y such that

$$Ax - y = \hat{b}, \quad x \geq 0, \quad y \geq 0, \quad x^T y = 0.$$

This problem is equivalent to the minimisation problem

$$\min_{x \geq 0} G(x), \qquad \text{where} \quad G(x) := \frac{1}{2}\left(x^T A x\right) - \hat{b}^T x.$$

The function G is strictly convex and the problem has a unique solution. See Lemma 4.11 in the english version of Seydel's book for a proof.

Now the SOR method for $Ax - y = \hat{b}$ $(= b - Af)$ in our free boundary value problem) is written componentwise as

$$r_i^{(k)} := \hat{b}_i - \sum_{j=1}^{i-1} a_{i,j} x_j^{(k)} - a_{i,i} x_i^{(k)} - \sum_{j=i}^{N-1} a_{i,j} x_j^{(k-1)} \tag{12.4}$$

with

$$x_i^{(k)} = x_i^{(k-1)} + \omega_R \frac{r_i^{(k)}}{a_{i,i}}, \tag{12.5}$$

where ω_R is the relaxation parameter which is chosen to accelerate convergence.

So far this does not take into account the positivity constraints. Cryer's projected SOR method starts with a vector $x^{(0)} \geq 0$ and modifies (12.5) to ensures that the successive $x^{(k)}$ satisfy $x^{(k)} \geq 0$.

Cryer's projected SOR method

outer loop: $k = 1, 2, \ldots$

inner loop: $i = 1, 2, \ldots, N-1$

$$r_i^{(k)} \quad \text{as in (12.4)}$$

$$x_i^{(k)} = \max\left\{0, x_i^{(k-1)} + \omega_R \frac{r_i^{(k)}}{a_{i,i}}, \right\}$$

$$y_i^{(k)} = -r_i^{(k)} + a_{i,i}\left(x_i^{(k)} - x_i^{(k-1)}\right)$$

To apply this to the discretized linear complementarity problem (12.3), we need to take

$$x = w - f, \qquad A = B, \qquad b = 0.$$

Kapitel 13

Stochastic Numerics

Suppose that an ODE

$$\frac{dx}{dt} = f(x)$$

represents an averaged effect and that the actual solutions are affected by noisy, that is we really have a noisy ODE

$$\frac{dx}{dt} = f(x) + \xi_t,$$

where ξ_t is a noise process and represents a random perturbation of the tangent direction.

More generally, the noise can have an intensity factor depending on the solution. Then, the noisy ODE has the form

$$\frac{dx}{dt} = f(x) + g(x)\xi_t.$$

In the physics and engineering literature ξ_t is meant to be Gaussian white noise

i.e. $\xi \sim N(0,1)$ (standard normally) distributed with ξ_s and ξ_t independent for $s \neq t$.

What does this actually mean ? We need the concepts of Random Variables and Stochastic Processes

13.1 Random variables and stochastic processes

A very fundamental example of a **random variable** is an $N(\mu, \sigma^2)$-distributed (Gaussian) random variable, which has the probability density

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2}$$

and the distribution function

$$F(a, b) = \int_a^b p(x)dx$$

is the probability that this RV takes values in the interval $[a, b]$.

The expected value and variance are basic properties of a random variable X.

Expected Value

$$\mathbb{E}(X) := \int_{-\infty}^{\infty} xp(x)dx$$

Variance

$$\mathrm{Var}(X) = \mathbb{E}\left(X - \mathbb{E}(X)\right)^2 = \mathbb{E}(X^2) - \left(\mathbb{E}(X)\right)^2)$$

e.g., for an $N(\mu, \sigma^2)$-distributed RV: $\mathbb{E}(X) = \mu$ and $\mathrm{Var}(X) = \sigma^2$.

We need the concept of a probability space to go further. This is a triple $(\Omega, \mathcal{A}, \mathbb{P})$ consisting of a sample space Ω of basic outcomes, σ-algebra $\mathcal{A}$ of (admissible) events and probability measure $\mathbb{P} : \mathcal{A} \to [0, 1]$.

A random variable is an $\mathcal{A}$-measurable function $X : \Omega \to \mathbb{R}$, i.e such that

$$\{\omega \in \Omega \,:\, X(\omega) - \leq a\} \in \mathcal{A}$$

for all $a \in \mathbb{R}$.

This says that the values taken by X are compatible with the information in the σ-algebra of admissible events.

We call the numerical value $X(\omega) \in \mathbb{R}$ a realisation of the RV X.

A **stochastic process** is essentially a family of random variables which is parameterized by time. Consider a time set $\mathbb{T} \subset \mathbb{R}$ (continuous time) or $\mathbb{Z}$ (discrete time).

A function $X : \mathbb{T} \times \Omega \to \mathbb{R}$ such that $X(t, \cdot)$ is a random variable for each $t \in \mathbb{T}$ is called a stochastic process.

We usually write $X_t(\cdot)$ for $X(t, \cdot)$ and we call a function $t \mapsto X_t(\omega)$ for a fixed ω a sample path. Thus a stochastic process is either a family of random variables $\{X_t(\cdot), t \in \mathbb{T}\}$ or a family of sample paths $\{X(\omega), \omega \in \Omega\}$.

A Wiener process or Brownian motion is one of the most important stochastic processes. Its time set is $\mathbb{T} = [0, \infty)$. The Wiener process $\{W_t : t \in [0, \infty)\}$ has the defining properties:

(1) $W_0 = 0$ with probability one

(2) $W_t \sim N(0, t)$, i.e. Gaussian distributed with

$$\mathbb{E}(W_t) = 0, \qquad \mathrm{Var}(W_t) = \mathbb{E}(W_t^2) = t$$

(3) independant increments: $W_{t_2} - W_{t_1}$ and $W_{t_4} - W_{t_3}$ are independent RVs for all $0 \leq t_1 < t_2 \leq t_3 < t_4$

Gaussian white noise is pathwise the derivative of a Wiener process

$$\xi_t(\omega) = \frac{d}{dt} W_t(\omega),$$

so we can write our noisy ODE as

$$\frac{dx}{dt} = f(x) + g(x)\frac{dW_t}{dt}$$

or as:

$$\frac{d}{dr} X_t = f(X_t) + g(xX_t)\frac{dW_t}{dt}$$

since the solution is a stochastic process.

But we have a BIG PROBLEM.

From the properties defining a Wiener process, one can show that the sample paths are continuous, but are nowhere differentiable. This says that Gaussian white noise does not exist! (at least as a well defined function - we need the theory of distributions).

Nondifferentiability is easy to see in the mean-square sense. Note that

$$W_t - W_s \sim N(0, t-s)$$

so

$$\mathbb{E}\left(\frac{W_{t+\Delta} - W_t}{\Delta}\right)^2 = \frac{\mathbb{E}(W_{t+\Delta} - W_t)^2)}{\Delta^2} = \frac{t+\Delta-t}{\Delta^2} = \frac{1}{\Delta},$$

and the limit as $\Delta \to 0$ does not exist.

13.2 Stochastic differential equations

We write a noisy ODE or stochastic differential equation SDE as

$$dX_t = f(X_t)dt + g(X_t)\, dW_t,$$

but this is only symbolic for a stochastic integral equation

$$X_t = X_0 + \int_0^t f(X_s)\, ds + \int_0^t g(X_s)\, dW_s$$

where

$$\int_0^t f(X_s(\omega))\, ds \qquad \text{pathwise a Riemann integral}$$

$$\int_0^t g(X_s)\, dW_s \qquad \text{Ito stochastic integral}$$

The Ito stochastic integral looks like a pathwise Riemann-Stieltjes integral, but this is not possible since the paths of a Wiener process are very irregular (though continuous) and not of bounded variation on any finite interval (which isa requirement for the existence of Riemann-Stieltjes integrals).

An Ito-Integral

$$\int_0^T h(t)\, dW_t$$

of a (possibly random) function $h : [0,T](\times\Omega) \to \mathbb{R}^1$ is defined as the mean-square limit of the finite sums

$$\sum_{n=0}^{N_T-1} h(t_n)\,(W_{t_{n+1}} - W_{t_n})$$

for all partitions $0 < t_1 < \ldots < t_n < \ldots < t_{N_T} = T$ as the maximum step size $\Delta = \max\{t_{n+1} - t_n\} \to 0$, i.e.,

$$\int_0^T h(t)dW_t = \text{ms} - \lim \sum_{n=0}^{N_{T-1}} h(t_n)\,(W_{t_{n+1}} - W_{t_n}).$$

Important point: the integrand function is always evaluated at the start t_n of each subinterval $[t_n, t_{n+1}]$ in contrast with a Riemann or Riemann-Stieltjes integral for which the evaluation point is chosen arbitrarily in $[t_n, t_{n+1}]$.

If h is a random function, then we require it to be non-anticipative, i.e. for any $\Delta > 0$

$$h(t,\cdot) \quad \text{and} \quad W_{t+\Delta} - W_t \qquad \text{are independent}$$

(i.e. $h(t,\cdot)$ is independent of the future noise) and mean-square integrable, i.e.

$$\int_0^T \mathbb{E}\,(h(t,\cdot))^2\,dt < \infty.$$

The Ito-integral has the following basic properties

$$\mathbb{E}\left(\int_0^T h(t)\,dW_t\right) = 0,$$

$$\mathbb{E}\left(\int_0^T h(t)\,dW_t\right)^2 = \int_0^T \mathbb{E}\,(h(t))^2\,dt \quad \text{(Ito isometry)}$$

where the last integral is a Riemann integral.

There are also some strange consequences of the definition, e.g.

$$\int_0^T W_t dW_t = \frac{1}{2}W_T^2 - \frac{1}{2}T.$$

In fact, the chain rule for Ito stochastic calculus is different from the deterministic chain rule.

13.3 The Euler-Maruyama Scheme

Let us consider an initial value problem for an Ito SDE on an interval $[0, T]$, i.e.

$$\begin{aligned} dX_t &= f(t, X_t)\, dt + g(t, X_t)\, dW_t \\ X_0 &= \bar{x}_0 \qquad \text{which could be random} \end{aligned}$$

Existence and uniqueness holds if f and g are globally Lipschitz and $\mathbb{E}\left(\bar{x}_0^2\right) < \infty$.

Consider a partition of $[0, T]$, i.e.

$$0 = t_0 < t < \ldots t_n < t_{+1} < \ldots < t_N = T$$

The SDE in integral form on the subinterval $[t_n, t_{n+1}]$ reads

$$\begin{aligned} X_{t_{n+1}} &= X_{t_n} + \int_{t_n}^{t_{n+1}} f(t, X_t)\, dt + \int_{t_n}^{t_{n+1}} g(t, X_t)\, dW_t \\ &\cong X_{t_n} + \int_{t_n}^{t_{n+1}} f(t_n, X_{t_n})\, dt + g(t_n, X_{t_n})\, dW_t \\ &= X_{t_n} + f(t_n, X_{t_n}) \int_{t_n}^{t_{n+1}} 1 \cdot ds + g(t_n, X_{t_n}) \int_{t_n}^{t_{n+1}} 1 \cdot dW_s \end{aligned}$$

i.e. we freeze the integrand functions to their values at the starting time t_n, which is compatible with the definition of the Ito stochastic integral.

Let us replace X_{t_n} by Y_n and write the above approximation equation as an equality. This gives us the Euler-Maruyama scheme.

$$Y_{n+1} = Y_n + f(t_n, Y_n)\, \Delta_n + g(t_n, Y_n)\, \Delta W_n$$

where

$$\begin{aligned} \Delta_n &= \int_{t_n}^{t_{n+1}} dt = t_{n+1} - t_n \\ \Delta W_n &= \int_{t_n}^{t_{n+1}} dW_t = W_{t_{n+1}} - W_{t_n} \end{aligned}$$

Note that $\Delta W_n \sim N(0, \Delta_n)$, i.e. ΔW_n is Gaussian distributed with

$$\mathbb{E}\left(\Delta W_n\right) = 0, \qquad \mathbb{E}\left(\left(\Delta W_n\right)^2\right) = \Delta_n.$$

The Euler-Maruyama scheme generates a discrete time stochastic process $\{Y_n, n = 0, 1, \ldots N_T\}$, where each Y_n is a random variable and so too is ΔW_n. However, in a computer we can only compute individual sample paths or realisation

$$\begin{array}{ccccccc} \Delta W_0(\omega) & & \Delta W_1(\omega) & & \Delta W_2(\omega) & & \\ & \searrow & & \searrow & & \searrow & \\ \bar{x}_0(\omega) = Y_0(\omega) & \longrightarrow & Y_1(\omega) & \longrightarrow & Y_2(\omega) & \longrightarrow & Y_3(\omega) \end{array}$$

For this we need to generate the realisations (for the same ω) of ΔW_n. We do this in 2 stages using the pseudo-random number generator RAN in a computer, e.g.

$$Z_{n+1} = aZ_n + b \mathrm{mod} odc$$

where, for example,

$$a = 2^{16} + 3, \qquad b = 0, \qquad c = 2^{31}$$

Then $U_n := Z_n/c$ are üniformly distributedïn $[0, 1]$

We call RAN twice and get 2 realisations of 2 independent uniformly distributed random variables U_1 and U_2. Then we use the Box-Muller method

$$\begin{aligned} N_1 &= \sqrt{-2\ln(U_1)}\cos(2\pi U_2) \\ N_2 &= \sqrt{-2\ln(U_1)}\sin(2\pi U_2) \end{aligned}$$

Then N_1, N_2 are realisations of 2 independent $N(0, 1)$ distributed Gaussian random variables and

$$\begin{aligned} \Delta W_{n_1}(\omega) &= N_1\sqrt{\Delta_n} \\ \Delta W_{n_2}(\omega) &= N_2\sqrt{\Delta_n} \end{aligned}$$

are two realisations of independent $N(0, \Delta_n)$ distributed random variables.

In this way we can implement the Euler-Maruyama scheme.

How good is the approximation $Y_n \simeq X_{t_n}$?

13.4 Convergence

We say that a numerical approximation Y_n^Δ with step size Δ has strong order $\gamma > 0$ to the solution X_t of an SDE on an interval $[0, T]$ if

$$\max_{n=0,1,\ldots,N_T} \mathbb{E}\left|Y_n^\Delta - X_{t_n}\right| \leq K_T \cdot \Delta^\gamma$$

where K_T depends on the length of interval T and on the coefficients of the SDE, etc.

For example, the Euler- Maruyama scheme has strong order

$$\gamma = \frac{1}{2}$$

for SDE with globally Lipschitz coefficients. Note that the order is the worst case for all such functions - we will see later that it can be better in special cases.

The approximation Y_n^0 has weak order $\beta > 0$ if

$$\left|\mathbb{E}\left(P(Y_n^\Delta)\right) - \mathbb{E}\left(P(X_{t_n})\right)\right| \leq K_{g,T} \cdot \Delta^\beta$$

for all polynomials P. Weak convergences essentially says all the moments convergence.

The Euler-Maruyama scheme has weak order

$$\beta = 1.$$

We note here that the strong and weak convergence orders of the Euler-Maruyama scheme are not the same

$$\gamma = \frac{1}{2} \neq 1 = \beta.$$

This is typical in stochastic numerics - in fact we construct different schemes for strong and weak convergence.

The orders of the Euler-Maruyama schemes are not very high. How can we construct schemes with a higher order ?

We should not use adaptations of the schemes for deterministic ODE since they either do not converge at all or converge only with a low order.

Example Consider the IVP for the SDE

$$dX_t = 2X_t\,dW_t, \qquad X_0 = 1$$

for which $f(x) \equiv 0$ and $g(x) \equiv 1$.

Integrating the SDE we get

$$X_t = 1 + 2\int_0^t X_s\,dW_s,$$

so

$$\mathbb{E}(X_t) = 1 + 2\mathbb{E}\left(\int_0^t X_s\,dW_s\right) = 1 + 2\cdot 0 = 1$$

by a property of the Ito integral, i.e.

$$\mathbb{E}(X_t) \equiv 1.$$

The Heun scheme here reads

$$\begin{aligned} Y_{n+1} &= Y_n + \frac{1}{2}\left[2Y_n + 2(Y_n + Y_n)\,\Delta W_n\right]\,\Delta W_n \\ &= Y_n\left[1 + 2\Delta W_n + 2(\Delta W_n)^2\right] \end{aligned}$$

We now write $Y_n^{(\Delta)}$ instead of Y_n to emphasize the dependence on the step size Δ. Since the RVs here in the product are independent, we have

$$\begin{aligned} \mathbb{E}\left(Y_n^{(\Delta)}\right) &= \mathbb{E}\left(\prod_{j=0}^{n-1}\left(1 + 2\Delta W_j + 2(\Delta W_j)^2\right)\right) \\ &= \prod_{j=0}^{n-1}\mathbb{E}\left(1 + 2\Delta W_j + 2(\Delta W_j)^2\right) \\ &= \prod_{j=0}^{n-1}\left[1 + 2\mathbb{E}\left(\Delta W_j\right) + 2\mathbb{E}\left(\Delta W_j\right)^2)\right] \\ &= \prod_{j=0}^{n-1}[1 + 2\Delta] \\ &= (1 + 2\Delta)^n = e^{2n\Delta} + 0(\Delta) \end{aligned}$$

Thus for time T we have

$$\begin{aligned} \mathbb{E}\left(Y_{N_T}^{(\Delta)}\right) &= e^{2T} + 0(\Delta), \qquad \text{since } N_T\Delta = T, \\ &\to e^{2T} \end{aligned}$$

BUT

$$\mathbb{E}\left(X_T\right) \equiv 1 < e^{2T} \qquad \Longrightarrow \quad \text{i.e., no weak convergence}$$

13.5 Stochastic Taylor expansions

To derive schemes of higher order we use the stochastic Taylor expansion, which is based on the stochastic chain rule or the Ito formula.

Let X_t be a solution of the Ito SDE

$$dX_t = f(X_t)\,dt + g(X_t)\,dW_t$$

and let

$$Y_t = U(t, X_t),$$

where $U : [0,T]\times\mathbb{R} \to \mathbb{R}$ is a smooth function. Then Y_t has the stochastic differential

$$dY_t = L^0U(t, X_t)\,dt + L^1U(t, X_t)\,dW_t,$$

where

$$\begin{cases} L^0U = & \dfrac{\partial U}{\partial t} + f\dfrac{\partial U}{\partial x} + \dfrac{1}{2}g^2\dfrac{\partial^2 U}{\partial x^2} \\ L^1U = & g\dfrac{\partial U}{\partial x} \end{cases}$$

i.e. in integral form

$$\boxed{U(t, X_t) = U(0, X_0) + \int_0^t L^0U(s, X_s)\,ds + \int_0^t L^1U(s, X_s)\,dW_s}$$

This is called the Ito-formula

13.5.1 An application of the Ito formula

We will prove that

$$d\left(W_t^2\right) = dt + 2W_t\, dW_t$$

i.e. in integral form that

$$W_t^2 - t = 2\int_0^t W_t\, dW_t$$

We apply the Ito formula to $U = x^2$, where $X_t \equiv W_t$, i.e. so X_t satisfies the Ito SDE

$$dX_t = 0\, dt + 1\, dW_t$$

with coefficients $f(x) \equiv 0$ and $g(x) \equiv 1$

Now for $U(x) = x^2$ we obtain $L^0U = 1$ and $L^1U = 2x$, so the Ito formula reads

$$d(X_t)^2 = 1\, dt + 2X_t\, dW_t$$

But $X_t = W_t$, so

$$d\left(W_t^2\right) = 1\, dt + 2W_t\, dW_t.$$

Let us integrate this from t_n to t_{n+1}. Then

$$2\int_{t_n}^{t_{n+1}} W_t\, dW_t = \int_{t_n}^{t_{n+1}} d(W_t^2) - \int_{t_n}^{t_{n+1}} dt = \left[W_{t_{n+1}}^2 - W_{t_n}^2\right] - [t_{n+1} - t_n]$$

Now consider the double integral

$$\begin{aligned}
\int_{t_n}^{t_{n+1}} \int_{t_n}^{s} dW_\tau\, dW_s &= \int_{t_n}^{t_{n+1}} (W_s - W_{t_n})\, dW_s \\
&= \int_{t_n}^{t_{n+1}} W_s\, dW_s - W_{t_n}\int_{t_n}^{t_{n+1}} dW_s \\
&= \int_{t_n}^{t_{n+1}} W_s\, dW_s - W_{t_n}\left[W_{t_{n+1}} - W_{t_n}\right] \\
&= \frac{1}{2}\left(W_{t_{n-1}} - W_{t_n}\right)\left(W_{t_{n+1}} + W_{t_n}\right) - \frac{1}{2}\Delta_n - W_{t_n}\left(W_{t_{n+1}} - W_{t_n}\right) \\
&= \frac{1}{2}\left(W_{t_{n+1}} - W_{t_n}\right)^2 - \frac{1}{2}\Delta_n
\end{aligned}$$

$$= \frac{1}{2}\Delta W_n^2 - \frac{1}{2}\Delta_n$$

We use this in Milstein Scheme in the next subsection.

13.5.2 Examples of stochastic Taylor expansions

Note if $U(t,x) = x$, then $L^0U = f$ and $L^1U = g$ and the Ito formula gives the original SDE in integral form

$$X_t = X_0 + \int_0^t f(X_s)\,ds + \int_0^t g(X_s)\,dW_s.$$

Now let us now apply the Ito formula to the integrand functions here. Then

$$\begin{aligned} X_t &= X_0 + \int_0^t \left[f(X_0) + \int_0^s L^0 f(X_\tau)\,d\tau + \int_0^s L^1 f(X_\tau)\,dW_\tau \right] ds \\ &\quad + \int_0^t \left[g(X_0) + \int_0^s L^0 g(X_\tau) d\tau + \int_0^s L^1 g(X_\tau)\,dW_\tau \right] dW_s \\ &= X_0 + f(X_0)\int_0^t ds + g(X_0)\int_0^t dW_s \qquad \underline{\text{Taylor approximation}} \\ &\quad + \int_0^t \int_0^s L^0 f(X_\tau)\,d\tau\,ds + \int_0^t \int_0^s L^1 f(X_\tau)\,dW_\tau\,ds \\ &\quad + \int_0^t \int_0^s L^0 g(X_\tau)\,d\tau\,dW_s + \int_0^t \int_0^s L^1 g(X_\tau)\,dW_\tau\,dW_s \end{aligned}$$

If we apply this on the interval $[t_n, t_{n+1}]$, and discard the remainder we get

$$X_{t_{n+1}} \simeq X_{t_n} + f(X_{t_n})\int_{t_n}^{t_{n+1}} dt + g(X_{t_n})\int_{t_n}^{t_{n+1}} dW_t$$

Then, if we replace X_{t_n} by Y_n and make this into an equation, we get the Euler-Maruyama scheme

$$Y_{n+1} = Y_n + f(Y_n)\underbrace{\int_{t_n}^{t_{n+1}} dt}_{\Delta_n} + g(Y_n)\underbrace{\int_{t_n}^{t_{n+1}} dW_t}_{\Delta W_n}.$$

The Euler-Maruyama scheme is thus the simplest nontrivial stochastic Taylor scheme.

Let us now apply the Ito formula to the integrand function $L^1 g(X_t)$ in the above remainder. Then we obtain next stochastic Taylor approximation (among many!)

$$X_t = X_0 + f(X_0)\int_0^t ds + g(X_0)\int_0^t dW_s + L^1 g(X_0)\int_0^t \int_0^s dW_\tau\, dW_s$$

$$\text{(with remainder)} \begin{cases} + & \int_0^t \int_0^s L^0 f(X_\tau)\, d\tau\, ds + \int_0^t \int_0^s L^1 f(X_\tau))\, dW_\tau\, ds \\ + & \int_0^t \int_0^s L^0 g(X_\tau)\, d\tau\, dW_s + \int_0^t \int_0^s \int_0^\tau L^0 L^1 g(X_\rho)\, d\rho\, dW_\tau\, dW_s \\ + & \int_0^t \int_0^s \int_0^\tau L^1 L^1 g(X_\rho)\, dW_\rho\, dW_\tau\, dW_s \end{cases}$$

13.6 Milstein scheme

Applying the final stochastic Taylor expansion on $[t_n, t_{n+1}]$, deleting the remainder and replacing X_{t_n} by Y_n yields the Milstein scheme

$$\begin{aligned} Y_{n+1} &= Y_n + f(Y_n)\int_{t_n}^{t_{n+1}} dt + g(Y_n)\int_{t_n}^{t_{n+1}} dW_t \\ &\quad + L^1 g(Y_n)\int_{t_n}^{t_{n+1}} \int_{t_n}^{s} dW_\tau dW_s \end{aligned}$$

Here

$$L^1 g(x) = g(x)\frac{\partial g}{\partial x}(x)$$

$$\Delta_n = \int_{t_n}^{t_{n+1}} dt = t_{n+1} - t_n$$

$$\Delta W_n = \int_{t_n}^{t_{n+1}} dW_t = W_{t_{n+1}} - W_{t_n} \sim N(0, \Delta_n)$$

and

$$\int_{t_n}^{t_{n+1}} \int_{t_n}^{s} dW_\tau\, dW_s = \frac{1}{2}(\Delta W_n)^2 - \frac{1}{2}\Delta_n.$$

Hence the Milstein scheme reads

$$Y_{n+1} = Y_n + f(Y_n)\Delta_n + g(Y_n)\Delta W_n + \frac{1}{2} g(Y_n)\frac{\partial g}{\partial x}(Y_n)\{(\Delta W_n)^2 - \Delta_n\}$$

The Milstein scheme has strong order $\gamma = 1$, but weak order $\beta = 1$. It was an improvement on the Euler-Maruyama scheme for strong convergence, but

is not better for weak convergence (and requires more work).

Note that for additive noise, i.e.

$$\frac{\partial g}{\partial x}(x) \equiv 0,$$

the last term vanishes and the Milstein scheme reduces to the Euler-Maruyama scheme.

Thus the Euler-Maruyama scheme has strong order $\gamma = 1$ for additive noise SDE (it is in fact the Milstein scheme!).

Literaturverzeichnis

[1] B. Aulbach, *Gewöhnliche Differentialgleichungen*, Spektrum

[2] P. Deuflhard und V. Bornemann, *Numerische Mathematik II. Integration gewöhnlicher Differentialgleichungen.* Göttingen: de Gruyter, 1994.

[3] M. Hanke-Bourgeois, *Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens.* Stuttgart: Teubner 2002.

[4] A. Iserles *A First Course in the Numerical Analysis of Differential Equations,* Cambridge: Cambridge University Press 1996.

[5] P.E. Kloeden *Skript zur Vorlesung: Einführung in die Numerische Mathematik,* Universität Frankfurt WS2009/10

[6] R.Plato *Numerische Mathematik kompakt,* Wiesbaden: Vieweg 2000.

[7] H.R. Schwarz, *Numerische Mathematik,* Stuttgart: Teubner 1997.

[8] G.D. Smith, *Numerical Solution of Partial Differential Equations,* Oxford: Oxford University Press 1969.

[9] K. Strehmel und R. Weiner, *Numerik gewöhnlicher Differentialgleichungen,* Teubner, Stuttgart, 1995.

[10] A. Stuart und R. Humphries, *Dynamical Systems and Numerical Analysis,* Cambridge University Press, Cambridge, 1996.

[11] F. Stummel und K. Hainer, *Praktische Mathematik,* Stuttgart: Teubner 1982.