V. Stabilitätsanalyse und implizite Verfahren

Ziel: - Stabilität, Stabilitätsgebiete, A-Stabilität
- Implizite RK-ESV
- Steife Probleme

Wozu: Steife Probleme treten oft in der Praxis auf (Schaltungen, Molekular-Dynamik, Zeitintegration von im Ort diskretisierten partielle Diff. Gl., z.B. Maxwell)

02.05.22 Bsp.: (0) Steifes Problem ⇝ Slides

V.1 Stabilitätsgebiete und A-Stabilität

Betrachten wir (wieder einmal) das einfache AWP

$$\dot{y}(t) = \lambda y(t)$$

$$y(0) = y_0$$

mit $\lambda \in \mathbb{C}$. Im Kontext der Stabilität ist diese DGL auch bekannt als Dahlquist Test-Gleichung bzw. - Test-AWP.

Die Lösung ist einfach

$$y(t) = y_0 \cdot e^{\lambda t}$$

Wenden wir das (explizite) Euler-Verfahren auf obiges AWP an

$$\begin{aligned} y_{j+1} &= y_j + h \cdot f(t_j, y_j) \\ &= y_j + h\lambda y_j \\ &= (1 + h\lambda) y_j \qquad \searrow\ y_j = (1+h\lambda) y_{j-1} \\ &= (1 + h\lambda)^2 y_{j-1} \qquad \searrow\ y_{j-1} = (1+h\lambda) y_{j-2} \\ &\vdots \\ &= (1 + h\lambda)^{j+1} y_0 \end{aligned}$$

Nun wollen wir den qualitativen Verlauf der Lösung mit der Näherung vergleichen:

(i) $\lambda > 0$: $y(t)$ nimmt zu

y_j nimmt zu ✓

(ii) $\lambda < 0$: $y(t)$ nimmt ab

y_j ? Hängt von der Schrittweite ab !

(oszillierende)

Dies erklärt das "explodieren" (präziser: das numerisch instabile Verhalten) des Euler-Verfahrens in Aufgabe 5, Serie 10.

Wenden wir nun das implizite Euler-Verfahren auf obiges AWP an:

$$y_{j+1} = y_j + h \cdot f(t_{j+1}, y_{j+1})$$

$$= y_j + h\lambda y_{j+1}$$ (auflösen nach y_{j+1} IMPLIZIT !)

$$\rightsquigarrow \quad y_{j+1} = \frac{1}{1-h\lambda} y_j$$

$$\vdots \qquad y_j = \frac{1}{1-h\lambda} y_{j-1}$$

$$= \left(\frac{1}{1-h\lambda}\right)^{j+1} y_0$$

Wie sieht es hier aus bei $\lambda < 0$?

$y(t)$ nimmt ab

y_j ? nimmt ab. Also unabhängig von der Schrittweite!

Dies erklärt das Verhalten der impliziten Euler-Verfahren in Aufgabe 5, Serie 10.

Def.: ESV angewendet auf das Dahlquist AWP kann man in folgender Form schreiben

$$y_{j+1} = g(z)\, y_j$$

wobei $z = h\lambda$ und $g(z)$ heisst Stabilitätsfunktion (SF).

Also: - expliziter Euler $\quad g(z) = 1 + z$

- impliziter Euler $\quad g(z) = \dfrac{1}{1-z}$

Die SF der bereits kennengelernten RK Verfahren sind

- verb. Euler-Verfahren $\quad g(z) = 1 + z + \frac{1}{2} z^2$
- Heun-Verfahren $\quad g(z) = 1 + z + \frac{1}{2} z^2$
- klassisches RK $\quad g(z) = 1 + z + \frac{1}{2} z^2 + \frac{1}{6} z^3 + \frac{1}{24} z^4$

(→ Übung!)

↑ Was fällt auf?

(Beachte: exakte Lösung ist $y(t) = y_0 \cdot e^{\lambda t}$)

Die SF sehen aus wie die (abgeschnittene) Taylor-Entwicklung.

Man ist natürlich daran interessiert, dass die Näherungslösung den selben qualitativen Verlauf hat.

Für den Fall $\lambda < 0$ verlangen wir, dass die Lösung betragsmässig abnimmt

$$|y_{j+1}| < |y_j| \qquad \text{(Absolute Stabilität)}$$

Also

$$|y_{j+1}| = |g(z)\, y_j| = |g(z)||y_j| < |y_j|$$

führt auf

$$|g(z)| < 1$$

Dies motiviert folgende Definition

Def.: Geg. ein ESV und zugehörige SF $g(z)$. Das Gebiet

$$SG = \{ z = h\lambda \in \mathbb{C} \mid |g(z)| < 1 \}$$

heisst **Stabilitätsgebiet** (SG) des Verfahrens. Für $\lambda \in \mathbb{R}$ spricht man analog vom **Stabilitäts-Intervall** (SI) des Verfahrens

$$SI = \{ x = h\lambda \in \mathbb{R} \mid |g(x)| < 1 \}$$

Bsp.: (1) SG von Euler
verbesserter Euler
Heun
klassisches RK

⇝ Slides

(2) Sei $\lambda = -200$ und wir verwenden das Euler-Verfahren. Wie müssen wir h wählen um absolut stabil zu sein?

$$g(z) = |1 + z| < 1$$

$$-1 < 1 + h\lambda < 1 \quad | -1$$

$$-2 < h\lambda < 0$$

$$-2 < -200h < 0 \quad | \times -\frac{1}{200}$$

$$\frac{2}{200} > h > 0$$

⇝ $0 < h < \frac{1}{100}$

Wie sieht es beim impliziten Euler Verfahren aus?

$$g(z) = \frac{1}{1-z}$$

$\rightsquigarrow$ SG auf Slides

Also keine Einschränkung des Zeitschritts aus Stabilitäts-Gründen!

Def.: Ein Verfahren heisst A-stabil, falls die gesamte linke komplexe Halbebene im SG enthalten ist

$$\{ z \in \mathbb{C} \mid Re(z) < 0 \} \subset SG$$

Also: das $\left\{ \begin{matrix} \text{explizite} \\ \text{implizite} \end{matrix} \right\}$ Euler Verfahren

$\left\{ \begin{matrix} \text{ist nicht} \\ \text{ist} \end{matrix} \right\}$ A-Stabil

Bsp.: (3) Studieren wir das SG der impliziten Mittelpunkts-Methode (IM) (Bsp. (1S) aus Kap. II):

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

auflösen ~> IMPLIZIT

$$k_1 = \frac{\lambda}{1 - h\lambda/2}\, \lambda y_j$$

$$y_{j+1} = y_j + h k_1$$

$$= y_j + \frac{h\lambda}{1 - h\lambda/2}\, y_j$$

$$= \left(1 + \frac{h\lambda}{1 - h\lambda/2}\right) y_j$$

$$\frac{1 - h\lambda/2}{1 - h\lambda/2}$$

$$= \frac{1 + h\lambda/2}{1 - h\lambda/2}\, y_j$$

$$\rightsquigarrow \quad g(z) = \frac{1 + z/2}{1 - z/2} \qquad \text{SF}$$

Frage: Ist die IM-Methode A-stabil?

~> Slides

Bem.: Für die exakte Lösung des Dahlquist Test-AWP gilt

$$\lim_{t \to \infty} y(t) = 0$$

Es wäre also wünschenswert, dass dies auf für ein Verfahren gilt. Dann hat man für die SF

$$\lim_{z \to -\infty} g(z) = 0$$

Man nennt ein Verfahren welches A-stabil ist und obige Bedingung erfüllt L-stabil.

z.B. das implizite Euler Verfahren ist L-stabil. Die implizite Mittelpunkts-Methode jedoch nicht (s. Bsp. 3).

V.2 Implizite Runge-Kutta Verfahren

Ein allgemeines RK ESV mit s Stufen ist gegeben durch folgendes Butcher Tableau:

c_1	a_{11}	a_{12}	$\cdots$	$a_{1,s-1}$	a_{1s}
c_2	a_{21}	a_{22}	$\cdots$	$a_{2,s-1}$	a_{2s}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
c_s	a_{s1}	a_{s2}	$\cdots$	$a_{s-1,s-1}$	a_{ss}
	b_1	b_2	$\cdots$	b_{s-1}	b_s

$\vec{c}$	A
	$\vec{b}^T$

Wenn A eine untere Dreiecksmatrix mit Nullen auf der Diagonalen ist, dann ist das RK Verfahren **explizit**.

Sonst ist es **implizit** $\rightsquigarrow$ i.A. muss ein nichtlineares Gleichungssystem gelöst werden!

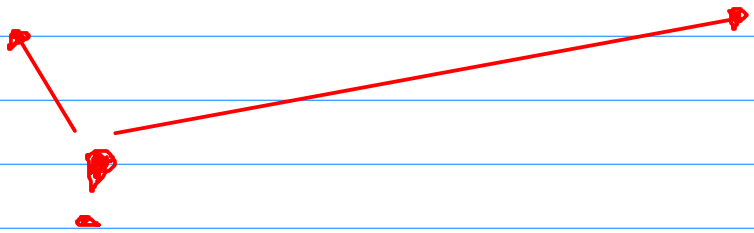
Ausgeschrieben

$$k_1 = f\left(t_j + c_1 \cdot h,\ y_j + h \cdot \left(a_{11} \cdot k_1 + a_{12} \cdot k_2 + \ldots + a_{1s} \cdot k_s\right)\right)$$

$$k_2 = f\left(t_j + c_2 \cdot h,\ y_j + h \cdot \left(a_{21} \cdot k_1 + a_{22} \cdot k_2 + \ldots + a_{2s} \cdot k_s\right)\right)$$

$$\vdots$$

$$k_s = f\left(t_j + c_s \cdot h,\ y_j + h \cdot \left(a_{s1} \cdot k_1 + a_{s2} \cdot k_2 + \ldots + a_{ss} \cdot k_s\right)\right)$$



Für skalare DGL sind dies s i.A. nichtlineare Gleichungen für s Unbekannte $(k_1, k_2, \ldots, k_s)$.
Für ein System von n DGLen sind dies ? $s \cdot n$ i.A. nicht lineare Gleichungen für ? $s \cdot n$ Unbekannte $(\vec{k}_1, \vec{k}_2, \ldots, \vec{k}_s)$.

Dies ist natürlich sehr aufwendig und deshalb nutzt man implizite Verfahren nur wenn es sich lohnt!

↳ Steife Probleme

Bsp.: (4) Impliziter Euler

1	1
	1

(5) Implizite Mittelpunkts-Methode (KO $p=2$)

$1/2$	$1/2$
	1

(6) Implizite Trapez-Methode (KO $p=2$)

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

Ausgeschrieben

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

$$k_2 = f\left(t_j + h, y_j + \frac{h}{2}(k_1 + k_2)\right)$$

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

Oft wird sie geschrieben als

$$y_{j+1} = y_j + \frac{h}{2}\left(f(t_j, y_j) + f(t_{j+1}, y_{j+1})\right)$$

(7) RK-Gauss Verfahren (KO $p=4$)

	$1/2 - \sqrt{3}/6$	$1/4$	$1/4 - \sqrt{3}/6$
Knoten →	$1/2 + \sqrt{3}/6$	$1/4 + \sqrt{3}/6$	$1/4$
		$1/2$	$1/2$
Gauss-Legendre Gewichte →			

(8) SDIRK (KO $p=3$)

Singly Diagonal Implicit RK

γ	γ	
$1-\gamma$	$1-2\gamma$	γ
	$1/2$	$1/2$

$$\gamma = \frac{3 \pm \sqrt{3}}{6}$$

Hier muss man auch nichtlineare Gleichungen lösen... Aber was ist ein Vorteil von SDIRK Methoden?

Bem.: Nicht alle impliziten RK ESV sind A-stabil ($\rightsquigarrow$ Übung)

Konvergenz expliziter RK-ESV

Bei expliziten RK-ESV lässt sich folgende maximale Konsistenzordnung $\tilde{p}$ zeigen:

s Anzahl Stufen

$$\tilde{p} = \begin{cases} s & \text{für } s \leq 4 \\ s-1 & \text{für } 4 < s \leq 7 \\ s-2 & \text{für } 8 \leq s \end{cases}$$

sehr oft genutzt

← "relativ" kompliziert

Falls die rechte Seite Funktion $\vec{f}(t, \vec{y})$ Lipschitz-stetig ist, so ist es auch die Verfahrensfunktion Φ.

Für $\vec{f}$ glatt genug erhalten wir von Satz II.3, dass das Verfahren mit der vollen Konsistenzordnung konvergiert.

(D.h. Konvergenz = Konsistenz-Ordnung)

Konvergenz impliziter RK-ESV

Bei impliziten RK-ESV lässt sich Folgende maximale Konsistenzordnung $\tilde{p}$ zeigen: (s Anzahl Stufen)

$$\tilde{p} = 2s$$

Diese wird durch sog. RK-Gauss ESV erreicht (wie in Bsp. (7)).

Aber Frage: Funktionieren diese Verfahren überhaupt?

Schreiben wir die RK-ESV Stufen so: (A: RK-Matrix)

$$\vec{v} = \vec{y}_j \vec{1} + h A \vec{F}(t_j, \vec{v})$$

$$\begin{pmatrix} \vec{v}^{(1)} \\ \vdots \\ \vec{v}^{(s)} \end{pmatrix} \qquad \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \qquad \begin{pmatrix} \vec{f}(t_j + c_1 h, \vec{v}^{(1)}) \\ \vdots \\ \vec{f}(t_j + c_s h, \vec{v}^{(s)}) \end{pmatrix}$$

Damit

$$\vec{y}_{j+1} = \vec{y}_j + h \vec{b}^T \vec{F}(t_j, \vec{v})$$

Man muss also in jedem Schritt ein i.A. nicht-lineares s·n System lösen

(s: Anzahl Stufen, n: Lösungskomponenten)

$$\vec{G}(\vec{v}) = \vec{v} - \vec{y}_j \vec{\lambda} - h A \vec{F}(t_j, \vec{v}) \doteq 0$$

Die naheliegende Idee eine Fixpunktiteration zu verwenden ist nicht zweckmässig, da bei den Problemen wo man implizite RK-ESV verwendet diese oft nur für sehr kleine Schrittweiten h konvergieren. (Steife Probleme ~ V.4)

In der Praxis verwendet man deshalb Newton-Methoden. Die Jacobi-Matrix hat die Form

$$J(\vec{v}) = \left(\delta_{il} - h\, a_{ik} \frac{\partial \vec{f}}{\partial \vec{y}}(t_j + c_i h, \vec{v}^{(i)}) \right)_{i,k=1}^{s}$$

Dies ist immer noch aufwendig!

(Die Jacobi-Matrix muss in jeder Iteration des Newton Verfahrens berechnet und "invertiert" werden)

Deshalb verwendet man oft sog. "vereinfachte" Newton Verfahren welche die Jacobi-Matrix nur einmal auswerten $J(\vec{y}_j)$ und einfrieren (d.h. die LU/LR-Zerlegung von J muss man nur einmal machen).

Stabilitätsfunktionen von RK-ESV

Für explizite RK-ESV ist die SF $g(z)$ ein Polynom vom Grad höchstens s in $z = h\lambda$. (↝ s. Übung SM1A02)

Wenden wir ein explizites RK-ESV auf das Dahlquist-AWP an:

i-te Stufe → da explizit

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

$$= \lambda\left(y_j + h \sum_{\ell=1}^{i-1} a_{i\ell} k_\ell\right)$$

Wir behaupten, dass $k_i = p_i(\lambda h)\,\lambda y_j$ und $p_i(h\lambda)$ ein Polynom vom Grad $i-1$ ist. Für $i = 1$ (Polynom vom Grad 0)

$$k_1 = f(t_j, y_j) = 1 \cdot \lambda y_j$$

stimmt es.

Nun per Induktion

$$k_{i+1} = f\Big(t_j + c_{i+1} h,\ y_j + h \sum_{l=1}^{i} a_{i+1,l}\, k_l\Big)$$

$$= \lambda\Big(y_j + h \sum_{l=1}^{i} a_{i+1,l}\, k_l\Big)$$

Induktions-Annahme

$$= \lambda\Big(y_j + h \sum_{l=1}^{i} a_{i+1,l}\ p_l(h\lambda)\ \lambda y_j\Big)$$

Polynome von max. Grad $l-1$

$$= \underbrace{\Big(1 + h\lambda \sum_{l=1}^{i} a_{i+1,l}\, p_l(h\lambda)\Big)}_{\text{Summe von Polynomen von max. Grad } i}\ \lambda y_j$$

$$= p_{i+1}(h\lambda)\ \lambda y_j$$

Die Behauptung stimmt ✓.

Für den RK-Schritt erhalten wir

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

$$= y_j + h \sum_{i=1}^{s} b_i\, p_i(h\lambda)\, \lambda y_j$$

$$= \Big(1 + h\lambda \sum_{i=1}^{s} b_i\, p_i(h\lambda)\Big)\, y_j$$

$$= g(h\lambda)\ y_j$$

g Polynom von höchstens Grad s ✓.

Für implizite RK-ESV ist die SF eine rationale Funktion in $z = h\lambda$.

(Quotient von zwei Polynomen)

Wenden wir ein implizites RK-ESV auf das Dahlquist-AWP an:

$$k_1 = f\left(t_j + c_1 h,\ y_j + h\sum_{l=1}^{s} a_{1l} k_l\right) = \lambda\left(y_j + h\sum_{l=1}^{s} a_{1l} k_l\right)$$

$$\vdots$$

$$k_s = f\left(t_j + c_s h,\ y_j + h\sum_{l=1}^{s} a_{sl} k_l\right) = \lambda\left(y_j + h\sum_{l=1}^{s} a_{sl} k_l\right)$$

$$\vec{k} = \lambda y_j \vec{1} + h\lambda A \vec{k}$$

($\vec{1}$: Einheitsmatrix)

$$\left(I - h\lambda A\right)\vec{k} = \lambda y_j \vec{1}$$

$$\vec{k} = \left(I - h\lambda A\right)^{-1} \vec{1}\ \lambda y_j$$

Für den Schritt

$$y_{j+1} = y_j + h\,\vec{b}^T \vec{k}$$

$$= \underbrace{\left(1 + h\lambda\,\vec{b}^T \left(I - h\lambda A\right)^{-1} \vec{1}\right)}\ y_j$$

$$= g(h\lambda)\, y_j$$

(Cramersche Regel)

Man kann zeigen, dass $g(z = h\lambda)$ rational ist.

V.3 Rückwärtsdifferenzenmethoden

In den Übungen haben wir uns mit Mehrschrittmethoden von Adams-Bashforth und Adams-Moulten befasst. Diese Verfahren sind Teil einer Familie von Verfahren: der sog. linearen Mehrschrittmethoden von der Form:

$$\sum_{\ell=0}^{k} \alpha_\ell \cdot y_{j+\lambda-\ell} = h \cdot \sum_{\ell=0}^{k} \beta_\ell \cdot f_{j+\lambda-\ell}$$

wobei $f_{j+\lambda-\ell} = f(t_{j+\lambda-\ell}, y_{j+\lambda-\ell})$ und α_ℓ, β_ℓ Koeffizienten sind.

Spezialfälle beschreiben folgende Verfahren:

- Adams-Bashforth: $\alpha_0 = 1$, $\alpha_1 = -1$ und $\alpha_\ell = 0$ für $\ell > 1$
  ↳ Übungen
  $\beta_0 = 0$ ← explizit
- Adams-Moulten: $\alpha_0 = 1$, $\alpha_1 = -1$ und $\alpha_\ell = 0$ für $\ell > 1$
  ↳ Übungen
  $\beta_0 \neq 0$ ← implizit
- Rückwärtsdifferenzenmethoden (Backward Differencing Methods BDF): $\beta_0 \neq 0$ und $\beta_\ell = 0$ für $\ell \geq 1$
  ↖ implizit

Die Idee eines k-Schritt BDF Verfahren ist die rechte Seite funktion f nur am neuen Zeitschritt, (t_{j+1}, y_{j+1}), zu evaluieren. Dies wird gleichgesetzt mit einer Approximation der Ableitung zur Zeit t_{j+1} welche man mittels Interpolation von $y_{j+1}, y_j, \dots, y_{j+1-k}$ bestimmt.

Bsp.: (a) BDF1: $k=1$

Bestimme das Interpolationspolynom durch y_{j+1}, y_j:

$$p_1(t) = y_{j+1} \cdot \frac{t-t_j}{h} - y_j \cdot \frac{t-t_{j+1}}{h}$$

Die Ableitung zur Zeit t_{j+1} ist dann

$$\frac{d}{dt} p_1(t)\Big|_{t=t_{j+1}} = \frac{y_{j+1}-y_j}{h} \approx \dot{y}(t)$$

Und damit

$$\frac{y_{j+1}-y_j}{h} = f(t_{j+1}, y_{j+1})$$

Oder

$$y_{j+1} - y_j = h \cdot f(t_{j+1}, y_{j+1})$$

BDF1 entspricht dem impliziten Euler Verfahren

Also $\beta_0 = 1$, $\alpha_0 = 1$, $\alpha_1 = -1$.

(10) **BDF2: $k=2$**

Bestimme das Interpolationspolynom durch y_{j+1}, y_j, y_{j-1}:

$$p_2(t) = y_{j-1} \cdot \frac{1}{2h^2}(t-t_j)(t-t_{j+1}) - y_j \cdot \frac{1}{h^2}(t-t_{j-1})(t-t_{j+1}) + y_{j+1} \cdot \frac{1}{2h^2}(t-t_{j-1})(t-t_j)$$

Die Ableitung zur Zeit t_{j+1} ist

$$\frac{d}{dt} p_2(t)\Big|_{t=t_{j+1}} = y_{j-1} \cdot \frac{1}{2h} - y_j \cdot \frac{2}{h} + y_{j+1} \frac{3}{2h} \approx \dot{y}(t)$$

Und damit

$$\frac{1}{h}\left(\frac{1}{2} y_{j-1} - 2y_j + \frac{3}{2} y_{j+1}\right) = f(t_{j+1}, y_{j+1})$$

Oder

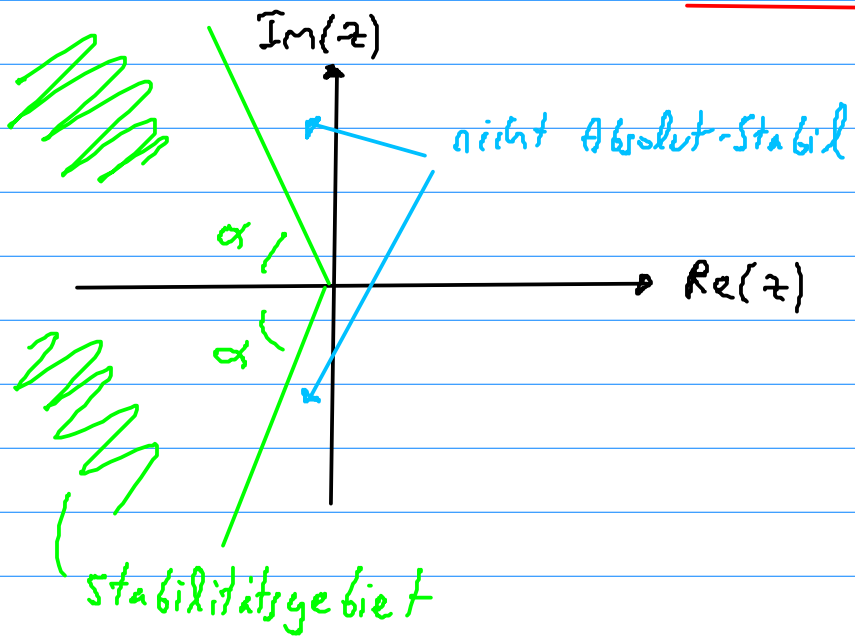
$$y_{j+1} - \frac{4}{3} y_j + \frac{1}{3} y_{j-1} = \frac{2}{3} h f(t_{j+1}, y_{j+1})$$

$\alpha_0 = 1$ $\quad$ $\alpha_1 = \frac{4}{3}$ $\quad$ $\alpha_2 = \frac{1}{3}$ $\quad$ $\beta_0 = \frac{2}{3}$

05.05.22

Bem.: (i) BDF1 ist das implizite Euler Verfahren

(ii) BDF Verfahren werden oft bei steifen Problemen verwendet ($\rightsquigarrow$ IV.4).
BDF1 und BDF2 sind A-stabil.
BDF3 bis BDF6 sind A(α)-stabil



(iii) Wie bei allen Mehrschrittverfahren muss man die ersten k Schritte z.B. mit einem geeigneten ESV berechnen

Konvergenz von linearen Mehrschrittverfahren

Die Konsistenzordnung wird wie bei ESV über den LDF definiert

$$e_{j+k} = y(t_{j+k}) - \frac{1}{\alpha_0}\left(-\sum_{l=1}^{k} \alpha_l \, y(t_{j+k-l}) + \sum_{l=0}^{k} \beta_l \, f(t_{j+k-l}, y(t_{j+k-l})) \right)$$

Also "einfach" um wieviel die exakte Lösung die diskrete bzw. approximierte Diff.-Gleichung nicht erfüllt. (Taylor-Entwicklung ...)

Der Konsistenzfehler ist

$$\tau_{j+k} = \frac{e_{j+k}}{h}$$

und die Konsistenzordnung $\tilde{p}$ wie bei ESV $\tau = \mathcal{O}(h^{\tilde{p}})$. (↝ s. Übung 509A04)

Aber: Im Gegensatz zu ESV impliziert Konsistenz noch nicht Konvergenz.
(s. z.B. Dahmen & Reusken (2006))

V.4 Steife Probleme

Steife Probleme begegnet man bei Systemen von Diff.-Gl. welche Prozesse mit stark unterschiedlichen Abklingzeiten modellieren. D.h. die Prozesse laufen auf sehr unterschiedlichen (sehr schnell / langsam) Zeitskalen ab.

Bsp.: (AA) Steifes lineares AWP ~~> Slides
(Übung Serie 12)

Ein lineares inhomogenes System

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

bezeichnet man als steif wenn für die Eigenwerte (EW) von A $(\lambda_1, \lambda_2, \dots, \lambda_n)$ gilt

$$Re(\lambda_i) < 0$$

und

$$S = \frac{\max\limits_{i=1,\dots,n} |Re(\lambda_i)|}{\min\limits_{i=1,\dots,n} |Re(\lambda_i)|}$$

(Steifigkeits-Parameter (Eng. stiffness ratio) parameter)

gross ist, d.h. $S \gg 1$

In Bsp. (11) ist $\lambda_1 = -1\text{h}$, $\lambda_2 = -15$, $\lambda_3 = -1000$:

$$S = ? \quad \frac{\max\limits_{i=1,\dots,3} |\lambda_i|}{\min\limits_{i=1,\dots,3} |\lambda_i|} = \frac{1000}{1\text{h}} \approx 1000$$

Steifigkeit tritt auch oft bei nichtlinearen DGLen auf

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

nichtlineare Vektorwertige Fkt.

Hier definiert man ein <u>lokales</u> Mass der Steifheit durch linearisieren an einem (interessanten) Punkt $t_n, \vec{y}_n$:

$$\vec{f}(t, \vec{y}(t)) = \vec{f}(t_n, \vec{y}_n) + \frac{\partial \vec{f}}{\partial t}(t_n, \vec{y}_n) \cdot (t - t_n) + J(t_n, \vec{y}_n) \cdot (\vec{y} - \vec{y}_n)$$

Jacobi-Matrix $\frac{\partial \vec{f}}{\partial \vec{y}}$

Durch rearrangieren der Terme, erhält man ein inhom. lin. System

$$\dot{\vec{y}}(t) = \underbrace{J(t_n, \vec{y}_n)}_{A} \vec{y}(t) + \underbrace{\left(\vec{f}(t_n, \vec{y}_n) + \frac{\partial \vec{f}}{\partial t}(t_n, y_n)(t - t_n) - J(t_n, y_n)\, \vec{y}_n \right)}_{\vec{c}}$$

Ist obige Linearisierung steif, so nennt man das nichtlineare System von DGLen lokal steif um den Punkt $(t_n, \vec{y}_n)$

Bsp.: (12) Steifes nichtlineares System
→ Slides
(13) Van der Pol Oszillator → Slides

Zur numerischen Behandlung steifer Probleme folgern wir aus Bsp. (11)–(14), dass explizite Verfahren ungeeignet sind. s. nächste Seite

D.h. ineffizient da die Schrittweite aus Stabilität- und NICHT Genauigkeits-Gründen gewählt werden muss

explizit	implizit
günstig pro Schritt	teuer pro Schritt
Schrittweite limitiert durch schnellste abfallende Komponente	Schrittweite nur durch gewünschte Genauigkeit limitiert

Bsp.: (14) Slides Die Wärmeleitungsgleichung (in ihrer einfachsten Form) in 2D für die Temperatur $u(x,y,t)$

$$\frac{\partial u}{\partial t} = \Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

wobei $(x,y) \in [0,1]^2$ und Zeit $t \geq 0$.

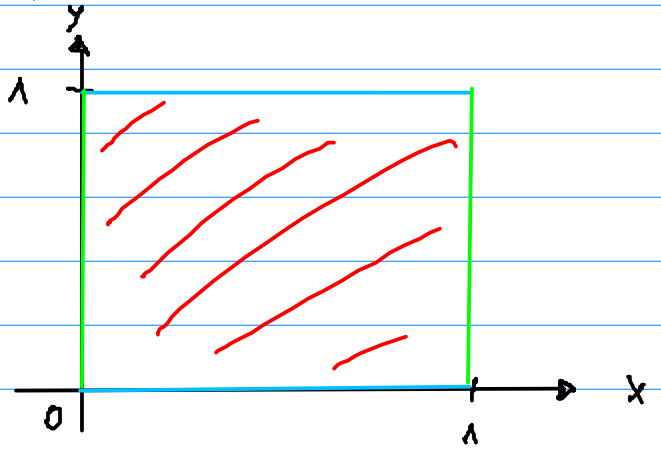
Die Anfangswerte seien

$$u(x,y,0) = \mathring{u}(x,y)$$

und die Randwerte

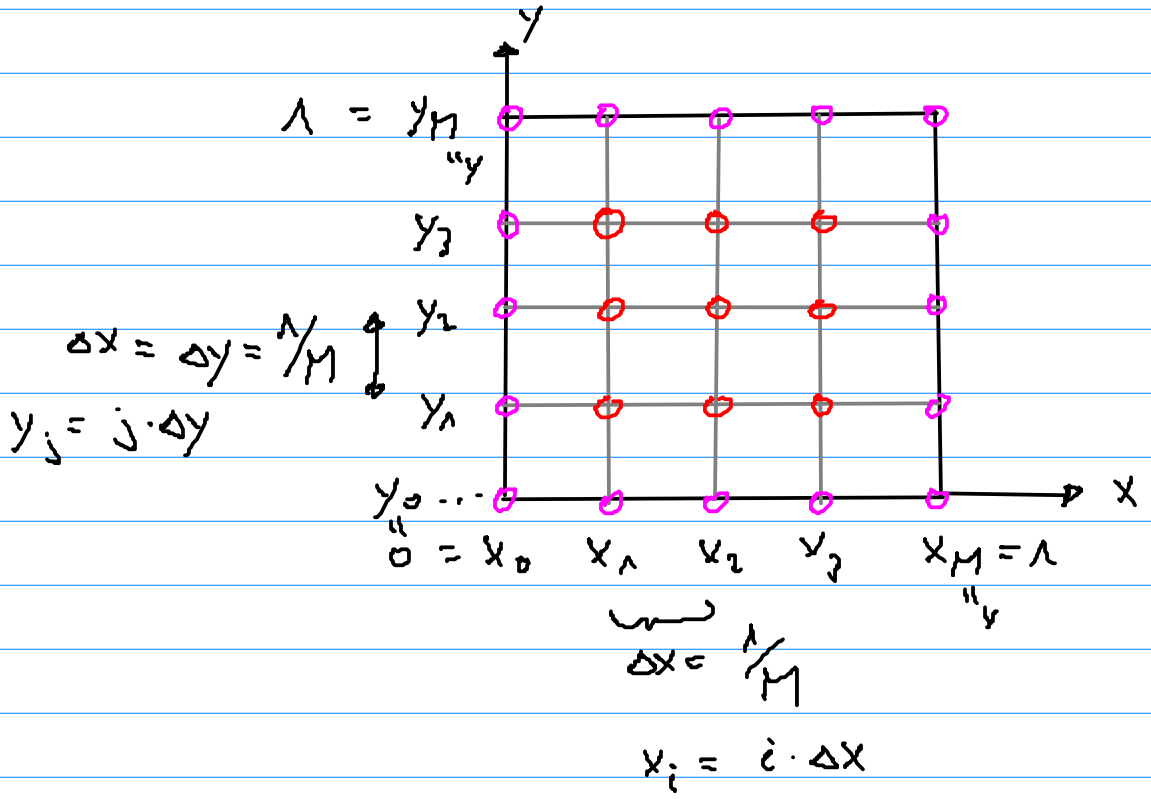
$$u(x,0/1,t) = u(0/1,y,t) = 0$$

Graphisch



Also ein Anfangs-Rand-Wert-Problem.

Verwenden wir die sog. Linien-Methode (method of lines) und diskretisieren das Problem zuerst in den Ortsvariablen. Dazu führen wir ein $(M+1)\times(M+1)$ Gitter (grid/mesh) ein:



Die Temperatur an den Gitter-Knoten wollen wir approximieren:

$$u(x_i, y_j, t) \approx u_{i,j}(t)$$

Dazu müssen wir die PDE approximieren:

$$\frac{\partial u}{\partial t} = \underbrace{\frac{\partial^2 u}{\partial x^2}}_{\text{Finite Differenzen}} + \underbrace{\frac{\partial^2 u}{\partial y^2}}_{\rightsquigarrow \text{Kap. } 1}$$

$$\approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2} = \frac{du_{i,j}}{dt}$$

für $1 \le i,j \le M-1$ (0-Knoten)

$u_{i,j} = 0$ am Rand (0-Knoten)

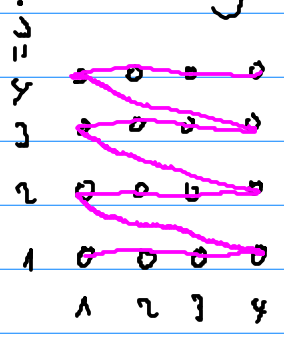
Dies ist ein lineares System von $(M-1) \times (M-1)$ DGlen mit AW

$$u_{i,j}(t=0) = \mathring{u}(x_i, y_j) \quad \text{für } 1 \le i,j \le M-1.$$

Die Temperaturen an den Gitter-Knoten packen wir in einen Lösungsvektor

$$\vec{y} = \begin{pmatrix} u_{1,1} \\ u_{2,1} \\ \vdots \\ u_{M-1,M-1} \end{pmatrix}$$

$(M-1)^2$ Komponenten



und schreiben das DGL System wie folgt

$$\dot{\vec{y}} = A\,\vec{y}$$

wobei die (Block-) Matrix

$$A = \begin{pmatrix} J & -I & & 0 \\ -I & J & \ddots & \\ & \ddots & \ddots & -I \\ 0 & & -I & J \end{pmatrix}$$

$(M-1)^2 \times (M-1)^2$ Matrix

und

$$J = \begin{pmatrix} 4 & -1 & & 0 \\ -1 & 4 & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 4 \end{pmatrix}$$

$(M-1) \times (M-1)$ Matrix

$$I = \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix}$$

$(M-1) \times (M-1)$ Einheitsmatrix

Die Matrix A ist dünn besetzt (sparse, viele Nulle!).

Für $n = 100$ ($\Delta x = 0.01$) ergibt sich bereits ein System von $99^2 = 9801$ DGLen!
(Man stellt sich leicht vor, dass in der Praxis viel grössere Systeme gibt.)

Nun müssen wir dieses AWP noch lösen. Um das lösen eines grossen LGS zu vermeiden ist man versucht einen expliziten Löser zu verwenden, z.B. das explizite Euler Verfahren

$$\vec{y}^{n+1} = \vec{y}^n + h A \vec{y}^n \qquad n = 0, 1, \ldots$$

Zeitindex

Aber man merkt schnell, dass dies eine schlechte Idee ist.
Das DGL System ist nämlich sehr steif. Man kann zeigen, dass die Eigenwerte von A mit $\frac{1}{\Delta x^2}$ skalieren.

Für $\Delta x = 0.01$ $(n = 100)$ ergeben sich so ungefähr 12'000 Zeit-Schritte bis zu $t = 0.1$.

16.05.22