

# 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)

Bsp.: (0) Steifes Problem  $\rightsquigarrow$  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

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

$$= y_j + h\lambda y_j$$

$$= (1 + h\lambda) y_j$$

$$\left\{ \begin{array}{l} y_j = (1 + h\lambda) y_{j-1} \end{array} \right.$$

$$= (1 + h\lambda)^2 y_{j-1}$$

$$\left\{ \begin{array}{l} y_{j-1} = (1 + h\lambda) y_{j-2} \end{array} \right.$$

⋮

$$= (1 + h\lambda)^{j+1} y_0$$

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

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

$y_j$  nimmt zu  $\checkmark$

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

$y_j$  ? hängt von der Schrittweite  $h$  ab!

(oszillierende)

3

Dies erklärt das "explodieren" (präziser: das numerisch instabile Verhalten) des Euler-Verfahrens in Aufgabe 4, Serie 10.

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

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

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

$$\begin{aligned} \rightsquigarrow y_{j+1} &= \frac{1}{1-h\lambda} y_j, & y_j &= \frac{1}{1-h\lambda} y_{j-1} \\ &\vdots & & \\ &= \left( \frac{1}{1-h\lambda} \right)^{j+1} y_0 \end{aligned}$$

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

$y(t)$  nimmt ab

$y_j$  ? nimmt immer ab, UNABHÄNGIG von Schrittweite  $h$

Dies erklärt das Verhalten des impliziten Euler-Verfahrens 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  $g(z) = 1 + z$
- impliziter Euler  $g(z) = \frac{1}{1 - z}$

Die SF der bereits kennengelernten RK Verfahren sind

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

(→ Übung!)

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

$g(z)$  sieht bei diesen Verfahren wie die ersten Glieder der Taylor-Entw. aus.

Man ist natürlich daran interessiert, dass die Nherungslosung den selben qualitativen Verlauf hat.

Fur den Fall  $\lambda < 0$  verlangen wir, dass die Losung betragsmassig abnimmt

$$|y_{j+1}| < |y_j| \quad (\text{Absolute Stabilitat})$$

Also

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

fuhrt auf

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

Dies motiviert folgende Definition

Def.: Geg. ein ESV und zugehorige SF  $g(z)$ .  
Das Gebiet

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

heisst Stabilitatsgebiet ( $S_G$ ) des Verfahrens.  
Fur  $\lambda \in \mathbb{R}$  spricht man analog vom Stabilitats-Intervall ( $S_I$ ) des Verfahrens

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

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?

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

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

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

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

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



$$\rightarrow 0 < h < \frac{1}{100}$$

Wie sieht es beim impliziten Euler Verfahren aus?

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

→ 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

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

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

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

Bsp.: (3) Studieren wir das SG der impliziten  
Mittelpunkts-Methode (IM) (Bsp. (15) aus Kap. II):

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

auflöser  $\leadsto$   
IMPLIZIT!

$$k_n = \frac{\lambda}{\lambda - h\lambda/2} \lambda y_j$$

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

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

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

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

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

$$\leadsto g(z) = \frac{\lambda + z/2}{\lambda - z/2} \quad \text{SF}$$

Frage: Ist die IM-Methode A-stabil?

$\leadsto$  slides



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

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

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

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

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 ESU mit  $s$  Stufen ist gegeben durch folgendes Butcher Tableau:

$c_1$	$a_{11}$	$a_{12}$	$\dots$	$a_{1,s-1}$	$a_{1s}$	$\vec{c}$	$A$	
$c_2$	$a_{21}$	$a_{22}$	$\dots$	$a_{2,s-1}$	$a_{2s}$			$\vec{b}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$			
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$			
$c_s$	$a_{s1}$	$a_{s2}$	$\dots$	$a_{s,s-1}$	$a_{ss}$			
	$b_1$	$b_2$	$\dots$	$b_{s-1}$	$b_s$			

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!

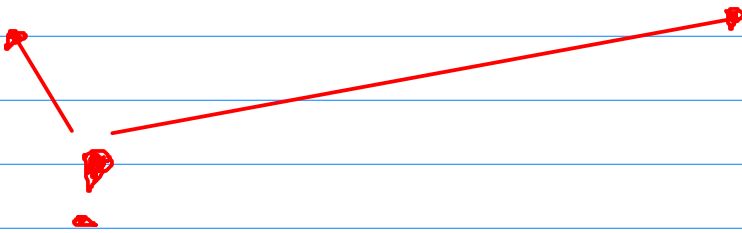
Ausgeschrieben

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

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

⋮

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



Für skalare DGL sind dies  $s$  i.A. nichtlineare Gleichungen für  $s$  Unbekannte  $(k_1, k_2, \dots, k_s)$ .

Für ein System von  $n$  DGLen sind dies  $? \cdot s \cdot n$  i.A. nicht lineare Gleichungen für  $? \cdot s \cdot n$  Unbekannte  $(\vec{k}_1, \vec{k}_2, \dots, \vec{k}_s)$ .

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

↳ Steife Probleme

Bsp.: (4) Impliziter Euler  $\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$

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

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

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

Ausgeschrieben

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

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

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

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

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  $p=4$ )

$$\begin{array}{c|cc} & 1/2 - \sqrt{3}/6 & 1/4 & 1/4 - \sqrt{3}/6 \\ \text{Knoten} & 1/2 + \sqrt{3}/6 & 1/4 + \sqrt{3}/6 & 1/4 \\ \hline \text{Gauss-Legendre Gewichte} & & 1/2 & 1/2 \end{array}$$

(8) SDIRK (KO  $\rho=3$ )

Singly Diagonal  
Implicit RK

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

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

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

Bem.: Nicht alle impliziten RK ESV sind A-stabil (siehe Übung)

## Konvergenz expliziter RK-ESV

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

$$\tilde{p} = \begin{cases} s & \text{für } s \leq 4 \\ s-1 & \text{für } 4 < s \leq 7 \\ s-2 & \text{für } 8 \leq s \end{cases} \left. \begin{array}{l} \text{sehr} \\ \text{oft} \\ \text{genutzt} \end{array} \right\}$$

↓  
s Anzahl Stufen

← "relativ" kompliziert

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

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:

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

s Anzahl Stufen  
↓

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

Aber Frage: Funktionieren diese Verfahren überhaupt?

Schreiben wir die RK-ESV Stufen so:

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

RK-Matrix

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

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  
Anzahl Stufen      Lösungskomponenten

$$\vec{G}(\vec{v}) = \vec{v} - \vec{y}_j \vec{1} - h A \vec{F}(t_j, \vec{v}) \stackrel{!}{=} 0$$

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  $\approx$  II.4)

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

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

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 SMA02)

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

$$\begin{aligned}
 \text{i-te Stufe} &\rightarrow k_i = f\left(t_j + c_i h, y_j + h \sum_{l=1}^{i-1} a_{il} k_l\right) \\
 &= \lambda \left( y_j + h \sum_{l=1}^{i-1} a_{il} k_l \right)
 \end{aligned}$$

da explizit

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

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

stimmt es.

Nun per Induktion

$$\begin{aligned}
 k_{i+1} &= f(t_j + c_{i+1} h, y_j + h \sum_{l=1}^i a_{i+1,l} k_l) \\
 &= \lambda \left( y_j + h \sum_{l=1}^i a_{i+1,l} k_l \right) \\
 &= \lambda \left( y_j + h \sum_{l=1}^i a_{i+1,l} \underbrace{p_l(h\lambda)}_{\substack{\text{Induktions-Annahme} \\ \text{Polynome von max.} \\ \text{Grad } l-1}} \lambda y_j \right) \\
 &= \underbrace{\left( 1 + h\lambda \sum_{l=1}^i a_{i+1,l} p_l(h\lambda) \right)}_{\substack{\downarrow \text{Summe von Polynomen von max. Grad } i}} \lambda y_j \\
 &= p_{i+1}(h\lambda) \lambda y_j
 \end{aligned}$$

Die Behauptung stimmt ✓

Für den RK-Schritt erhalten wir

$$\begin{aligned}
 y_{j+1} &= y_j + h \sum_{i=1}^s b_i k_i \\
 &= y_j + h \sum_{i=1}^s b_i p_i(h\lambda) \lambda y_j \\
 &= \left( 1 + h\lambda \sum_{i=1}^s b_i p_i(h\lambda) \right) y_j \\
 &= \underbrace{g(h\lambda)}_z y_j
 \end{aligned}$$

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:

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

$$\vdots$$

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

Einheitsmatrix

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

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

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

Für den Schritt

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

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

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

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

(Cramersche Regel)

## V.3 Rückwärtsdifferenzenmethoden

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

$$\sum_{l=0}^k \alpha_l \cdot y_{j+l-1} = h \cdot \sum_{l=0}^k \beta_l \cdot f_{j+l-1}$$

wobei  $f_{j+l-1} = f(t_{j+l-1}, y_{j+l-1})$  und  $\alpha_l, \beta_l$  Koeffizienten sind.

Spezialfälle beschreiben folgende Verfahren:

- Adams-Bashforth:  $\alpha_0 = 1, \alpha_1 = -1$  und  $\alpha_l = 0$  für  $l > 1$   
↳ Übungen

-  $\beta_0 = 0$  ← explizit

- Adams-Moulton:  $\alpha_0 = 1, \alpha_1 = -1$  und  $\alpha_l = 0$  für  $l > 1$   
↳ Übungen

-  $\beta_0 \neq 0$  ← implizit

- Rückwärtsdifferenzenmethoden (Backward Differencing Methods BDF):  $\beta_0 \neq 0$  und  $\beta_l = 0$  für  $l \geq 1$   
← implizit

Die Idee eines  $k$ -Schritt BDF Verfahrens 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

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

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

$y_{j+1}$ ,  $y_j$ ,  $y_{j-1}$ :

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

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

$$\begin{aligned} \left. \frac{d}{dt} p_2(t) \right|_{t=t_{j+1}} &= y_{j-1} \cdot \frac{1}{2h} - y_j \cdot \frac{2}{h} + y_{j+1} \cdot \frac{3}{2h} \\ &\approx \dot{y}(t) \end{aligned}$$

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

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

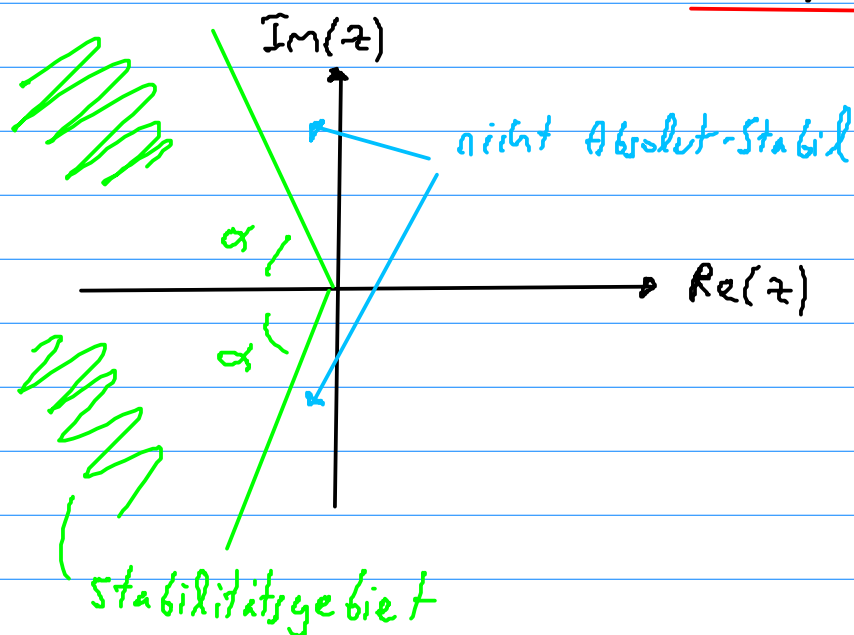
$\alpha_0 = 1$        $\alpha_1$        $\alpha_2$        $\beta_0$

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

(ii) BDF Verfahren werden oft bei steifen Problemen verwendet ( $\approx$  V. 4).

BDF1 und BDF2 sind A-stabil.

BDF3 bis BDF6 sind A( $\alpha$ )-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

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

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

Der Konsistenzfehler ist

$$\tau_{j+1} = \frac{e_{j+1}}{h}$$

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 Dif.-Gl. welche Prozesse mit stark unterschiedlichen Abklingzeiten modellieren.

D.h. die Prozesse laufen auf sehr unterschiedlichen (sehr schnell/langsam) Zeitskalen ab.

15.05.23

Bsp.: (11) Steifes lineares AWP no 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

$$\operatorname{Re}(\lambda_i) < 0$$

und

$$S = \frac{\max_{i=1, \dots, n} |\operatorname{Re}(\lambda_i)|}{\min_{i=1, \dots, n} |\operatorname{Re}(\lambda_i)|} \quad \left( \text{Steifigkeits-Parameter} \right)$$

(eng. stiffness ratio / parameter)

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

In Bsp. (11) ist  $\lambda_1 = -112$ ,  $\lambda_2 = -15$ ,  $\lambda_3 = -1000$ :

$$S = ?$$

Steifigkeit tritt auch oft bei nichtlinearen DGLen auf

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

↑ nicht lineare Vektorwertige Fkt.

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

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

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

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

$$\dot{\vec{y}}(t) = \underbrace{\mathcal{J}(t_n, \vec{y}_n)}_A \vec{y}(t) + \underbrace{\left( \vec{F}(t_n, \vec{y}_n) + \frac{\partial \vec{F}}{\partial t}(t_n, \vec{y}_n) (t - t_n) - \mathcal{J}(t_n, \vec{y}_n) \vec{y}_n \right)}_b$$

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) Die Wärmeleitungsgleichung (in ihrer einfachsten Form) in 2D für die Temperatur  $u(x, y, t)$

Slides

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

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

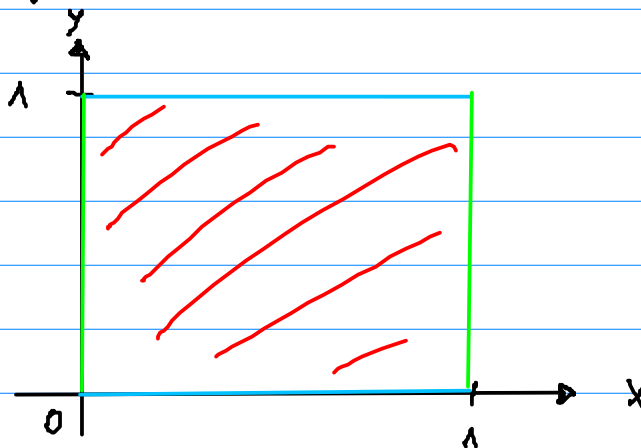
Die Anfangswerte seien

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

und die Randwerte

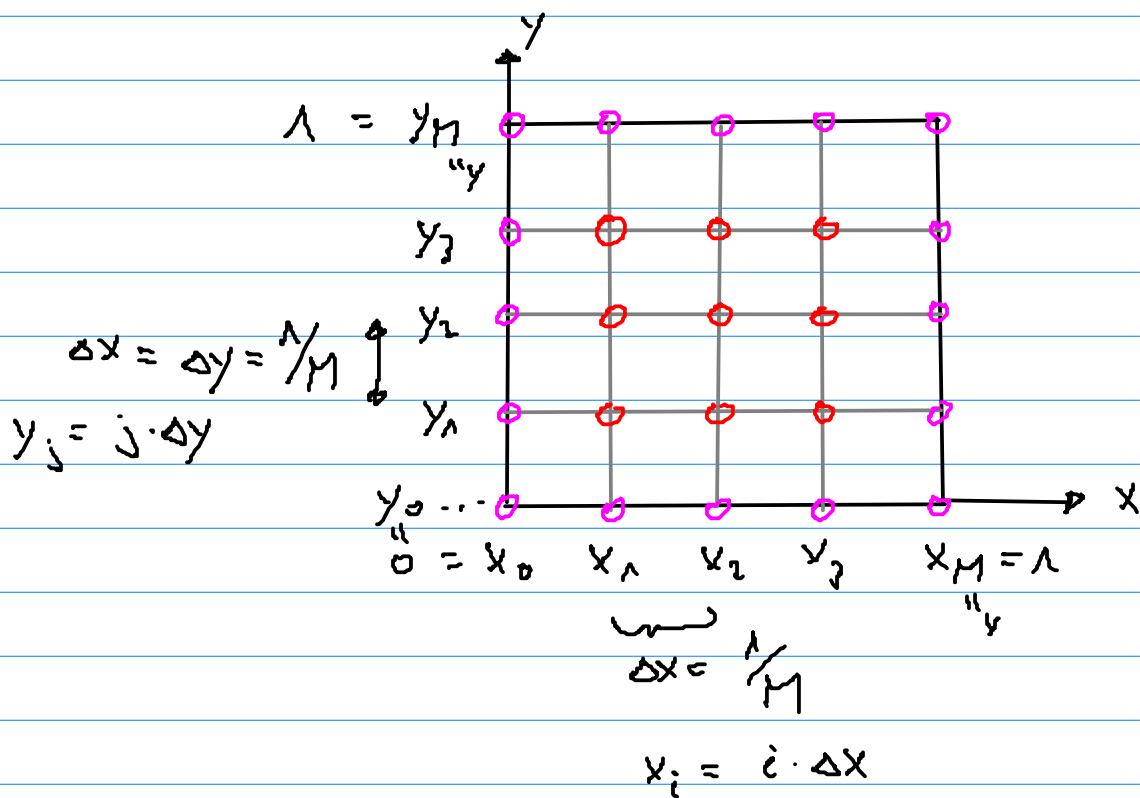
$$\underline{u(x, 0/1, t)} = \underline{u(0/1, y, t)} = 0$$

Graphisch



Also ein Anfangs-Randwert-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:

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

Finite Differenzen  
s. 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  $1 \leq i, j \leq M-1$  (0-Knoten)

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

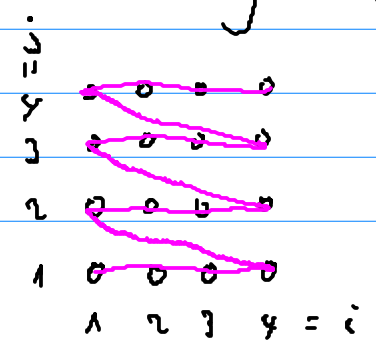
Dies ist ein lineares System von  $(M-1) \times (M-1)$  DGLen mit AW

$$u_{i,j}(t=0) = \overset{\circ}{u}(x_i, y_j) \text{ für } 1 \leq i, j \leq M-1.$$

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

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

$(M-1)^2$  Komponenten



und schreiben das DGL System wie folgt

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

wobei die (Block-) Matrix

$$A = \begin{pmatrix} \gamma - I & & & & 0 \\ -I & \gamma & & & \\ & & \ddots & & \\ & & & \ddots & \\ 0 & & & & \gamma - I \\ & & & & -I & \gamma \end{pmatrix} \quad \begin{matrix} (\pi-1)^2 \times (\pi-1)^2 \\ \text{Matrix} \end{matrix}$$

und

$$\gamma = \begin{pmatrix} \gamma - \lambda & & & & 0 \\ -\lambda & \gamma & & & \\ & & \ddots & & \\ & & & \ddots & \\ 0 & & & & \gamma - \lambda \\ & & & & -\lambda & \gamma \end{pmatrix} \quad \begin{matrix} (\pi-1) \times (\pi-1) \\ \text{Matrix} \end{matrix}$$

$$I = \begin{pmatrix} \lambda & & & & 0 \\ & \ddots & & & \\ & & \ddots & & \\ 0 & & & & \lambda \end{pmatrix} \quad \begin{matrix} (\pi-1) \times (\pi-1) \\ \text{Einheitsmatrix} \end{matrix}$$

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



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

$$\vec{y}^{n+1} = \vec{y}^n + h A \vec{y}^n \quad n=0,1,\dots$$

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

22.05.23