

# Inhaltsverzeichnis

<b>1. Parameterabhängige Anfangswertprobleme</b>	<b>3</b>
Differentiation der Lösung des AWP's . . . . .	5
Berechnung der Ableitungen: . . . . .	6
Variationsdifferentialgleichungen für Richtungsableitungen . . . . .	8
Ableitungsmatrizen bei abschnittsweise definierter rechter Seite . . . . .	10
Adjungierte Differentialgleichung . . . . .	11
Satz 1.12 . . . . .	11
Bemerkung 1.13 . . . . .	13
Ableitung von Linearkombinationen von $\frac{\partial y}{\partial y_0}$ . . . . .	13
Anwendung: . . . . .	14
Zusammenfassung: . . . . .	14
Simulation und Optimierungsprobleme bei Differentialgleichungen . . . . .	14
<b>2. Formulierung von Parameterschätzproblemen</b>	<b>17</b>
Problemformulierung . . . . .	18
Prozessmodell . . . . .	18
Modell für Beobachtungen . . . . .	18
Lösungsmethoden . . . . .	23
Parametrisierung der Lösung des AWP durch Single Shooting, Multiple Shooting oder Kollokation . . . . .	23
Löse (2.15) mit verallgemeinertem Gauß-Newton-Verfahren . . . . .	23
<b>3. Shooting-Verfahren und Kollokation</b>	<b>25</b>
3.1 Single-Shooting: Einzelschießverfahren . . . . .	25
Algorithmus 3.2: Verallgemeinertes Gauß-Newton-Verfahren . . . . .	26
Algorithmus 3.3: Single-Shooting Gauß-Newton . . . . .	26
Bemerkung 3.4 . . . . .	27
3.2 Multiple Shooting, die Mehrzielmethode . . . . .	28
Bemerkung 3.5 . . . . .	30
3.3 Kollokation . . . . .	30
Kollokations-Diskretisierung . . . . .	30
Wahl der Polynombasis und der Kollokationspunkte . . . . .	31
Konsistenzfehler: . . . . .	32
Beispiel Impliziter Euler, $k=1$ . . . . .	32
Kollokation für beschränkte Parameterschätzprobleme . . . . .	33
3.4 Ansätze zur Optimierung von DGL-Modellen . . . . .	34

3.5 Relaxierte Formulierung von DAEs . . . . .	35
<b>4. Verallgemeinerte Gauß-Newton-Verfahren</b>	<b>37</b>
Algorithmus 4.1 (Verallgemeinertes Gauß-Newton-Verfahren) . . . . .	37
Bemerkung 4.2 . . . . .	38
Annahmen 4.3: Regularitätsannahmen . . . . .	38
Lemma 4.4 . . . . .	38
Lösung der linearen Ausgleichsprobleme . . . . .	38
1. Unbeschränkter Fall . . . . .	38
Bemerkung 4.10 . . . . .	39
2. Beschränkter Fall . . . . .	39
Lemma 4.7 . . . . .	40
Numerische Lösung . . . . .	42
Lemma 4.11 (Berechnung der adjungierten . . . . .	45
3. Variante: Stoer 1979 . . . . .	45
Anwendung auf die Mehrzielmethode . . . . .	45
Algorithmus 4.12 (Eliminationsalgorithmus) . . . . .	46
Algorithmus 4.13: Lösen . . . . .	47
<b>5. Lokale Konvergenz von Newton-Typ-Verfahren</b>	<b>51</b>
Algorithmus 5.1 (Newton-Typ-Verfahren) . . . . .	51
Bemerkung 5.2 . . . . .	52
Satz 5.3: Lokaler Kontraktionssatz (Bock 1987) . . . . .	52
Korollar 5.4 . . . . .	54
Bemerkung 5.5: Quasi-Newton-Verfahren . . . . .	55
Satz 5.6 (Dennis-Moré . . . . .	55
Varianten von Quasi-Newton-Verfahren . . . . .	56
Bemerkung 5.7 (Bedeutung von $\omega$ ) . . . . .	56
Anwendung auf (verallgemeinerte) Gauß-Newton-Verfahren: . . . . .	56
Bemerkung 5.8: Bedeutung von $\kappa$ . . . . .	57

# 1. Parameterabhängige Anfangswertprobleme

*Parameterabhängige gewöhnliche Differentialgleichung* mit Parametervektor  $p$  und parameterabhängiger Anfangsbedingung:

$$\dot{y}(t) = f(t, y(t), p) \quad (1.1)$$

$$y(t_0) = y_0(p) \quad (1.2)$$

wobei

- $t \in [t_0, t_{end}]$  „Zeit“
- $y: [t_0, t_{end}] \rightarrow \mathbb{R}^{n_y}$  „Zustände“
- $p \in \mathbb{R}^{n_p}$  „Parameter“
- $f: [t_0, t_{end}] \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_y}$  sei hinreichend oft in  $t, y$  (stückweise) stetig differenzierbar, damit numerische Integrationsverfahren mit Fehlerkontrolle funktionieren, außerdem einmal in  $p$  stetig differenzierbar.
- $y_0: \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_y}$  einmal stetig differenzierbar.

*Variante:* Differentiell-algebraische Gleichungssysteme (DAE):

$$\begin{aligned} f: [t_0, t_{end}] \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_z} \times \mathbb{R}^{n_p} &\rightarrow \mathbb{R}^{n_y} \\ g: [t_0, t_{end}] \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_z} \times \mathbb{R}^{n_p} &\rightarrow \mathbb{R}^{n_z} \\ \dot{y} &= f(t, y, z, p) \\ 0 &= g(t, y, z, p) \\ y: [t_0, t_{end}] &\rightarrow \mathbb{R}^{n_y} && \text{differentielle Zustände} \\ z: [t_0, t_{end}] &\rightarrow \mathbb{R}^{n_z} && \text{algebraische Zustände} \end{aligned} \quad (1.3)$$

Das DAE benötigt nur Anfangswerte für  $y$ :

$$y(t_0) = y_0(p) \quad (1.4)$$

Die Anfangswerte der  $z$  sind durch die *Konsistenzbedingung* gegeben:

$$g(t_0, y(t_0), z(t_0), p) = 0 \quad (1.5)$$

Wir betrachten in dieser Vorlesung nur den *Index-1-Fall*, d. h.

$$\frac{\partial g}{\partial z} \quad \text{hat den Rang } n_z \quad (1.6)$$

Daraus folgt, dass die algebraischen Gleichungen lokal eindeutig nach  $\dot{z}$  auflösbar sind:

$$\begin{aligned} 0 &= g(t, y, z, p) \\ 0 &= \frac{\partial}{\partial t} g(t, y, z, p) \\ &= \frac{\partial g}{\partial t} + \frac{\partial g}{\partial y} \dot{y} + \frac{\partial g}{\partial z} \dot{z} \\ \dot{z} &= - \left( \frac{\partial g}{\partial z} \right)^{-1} \left( \frac{\partial g}{\partial t} + \frac{\partial g}{\partial y} \dot{y} \right) \end{aligned}$$

Man erhält zusammen mit  $\dot{y} = f$  eine ODE für  $y$  und  $z$ , deren Anfangswerte aber die Konsistenzbedingung (1.5) erfüllen müssen. Diese ODE- bzw. DAE-Anfangswertprobleme können auch aus im Ort diskretisierten Anfangs-(Rand-)Wertproblemen von instationären partiellen Differentialgleichungen kommen. In dieser Vorlesung behandeln wir nicht die Diskretisierungsmethoden für PDEs.

*Satz 1.1:* Lokaler Stabilitätssatz

Seien die beiden Anfangswertprobleme

$$\dot{u}(t) = f(t, u, p_1), \quad u(t_0) = u_0 \quad (1.7)$$

$$\dot{v}(t) = f(t, v, p_1), \quad v(t_0) = v_0 \quad (1.8)$$

auf  $[t_0, t_{\text{end}}]$  gegeben. Die Funktion  $f(t, y, p)$  sei stetig in  $(t, y)$  und genüge einer Lipschitz-Bedingung in  $y$  mit Konstante  $L < \infty$ . Dann gilt für die Lösungen  $u$  und  $v$  von (1.7) und (1.8):

$$\|u(t) - v(t)\| \leq e^{L(t-t_0)} \left( \|u_0 - v_0\| + \int_{t_0}^t \sup_{t,y} \|f(\tau, y(\tau), p_1) - f(\tau, y(\tau), p_2)\| d\tau \right)$$

Der Beweis wurde bereits in Numerik 1 gegeben, er benutzt das Gronwall-Lemma.

*Korollar 1.2:* Trompetenabschätzung

Sei  $f$  wie in Satz 1.1 mit  $L < \infty$ . Für die Abweichung  $\delta y := v - y$  der Lösungen  $y$  und  $v$  von

$$\begin{aligned}\dot{y}(t) &= f(t, y, p), & y(t_0) &= y_0(p) \\ \dot{v}(t) &= f(t, v, p_1), & v(t_0) &= y_0(p + \delta p)\end{aligned}$$

gilt:

$$\|\delta y(t)\| \leq \varepsilon_1 e^{L(t-t_0)} + \varepsilon_2 e^{L(t-t_0)}(t - t_0) \quad (1.10)$$

wobei

$$\begin{aligned}\|\delta y_0\| &:= \left\| \frac{\partial y_0}{\partial p}(p) \delta p \right\| \leq \varepsilon_1 \\ \|\delta f\| &:= \left\| \frac{\partial f}{\partial p}(t, y, p) \delta p \right\| \leq \varepsilon_2\end{aligned}$$

*Beweis:*

Siehe Numerik 1, folgt aus Satz 1.1.

Für Änderungen von  $p$  können die Lösungen der Anfangswertprobleme exponentiell auseinanderlaufen. Kleine Störungen von  $p$  in Anfangsbedingungen und Rechter-Seite-Funktion können sehr große Unterschiede in  $y(t_{end})$  zur Folge haben.

## Differentiation der Lösung des AWP

*Schreibweise:*

Betrachte das Anfangswertproblem

$$\dot{y} = f(t, y, p) \quad y(0) = y_0 \quad (1.11)$$

Die Lösung  $y$  hängt von  $t$ ,  $t_0$ ,  $y_0$  und  $p$  ab. Wir schreiben:

$$y(t) = y(t; t_0, y_0, p)$$

*Satz 1.3:*

Sei  $f \in \mathbb{C}^m$ ,  $m \geq 1$ . Dann ist  $y(t; t_0, y_0, p)$

- $(m+1)$  mal stetig differenzierbar in  $t$
- $m$  mal stetig differenzierbar in  $t_0, y_0, p$

*Beweis:*

Integraldarstellung der Lösung:

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

*Definition 1.4:* Ableitungen der Lösung des AWP

- Ableitung nach Anfangswerten:  $G(t; t_0, y_0, p) := \frac{\partial}{\partial y_0} y(t; t_0, y_0, p)$
- Ableitung nach Anfangszeitpunkt:  $G_{t_0}(t; t_0, y_0, p) := \frac{\partial}{\partial t_0} y(t; t_0, y_0, p)$
- Ableitung nach Parametern:  $G_p(t; t_0, y_0, p) := \frac{\partial}{\partial p} y(t; t_0, y_0, p)$

### Berechnung der Ableitungen:

Integralform des AWP:

$$y(t) = y_0 + \int_{t_0}^t f(\tau, y(\tau), p) \, d\tau \quad (1.12)$$

#### a) Differenziere nach Anfangswerten:

$$\begin{aligned} \frac{\partial}{\partial y_0} y(t) &= \frac{\partial y_0}{\partial y_0} + \int_{t_0}^t \frac{\partial}{\partial y} f(\tau, y(\tau), p) \frac{\partial y}{\partial y_0}(\tau) \, d\tau \\ G(t) &= I + \int_{t_0}^t \frac{\partial}{\partial y} f(\tau, y(\tau), p) G(\tau) \, d\tau \end{aligned} \quad (1.13)$$

Dies ist äquivalent zum Anfangswertproblem

$$\begin{aligned} \dot{G}(t) &= \frac{\partial f}{\partial y}(t, y(t), p) G(t) \\ G(t_0) &= I \end{aligned} \quad (1.14)$$

zur sogenannten Variationsdifferentialgleichung (VDE) nach Anfangswerten.

**Differenziere nach dem Anfangszeitpunkt:**

$$\begin{aligned}\frac{\partial}{\partial t_0} y(t) &= \frac{\partial y_0}{\partial t_0} + \frac{\partial}{\partial t_0} \int_{t_0}^t f(\tau, y(\tau), p) d\tau \\ G_{t_0}(t) &= 0 - f(t_0, y(t_0), p) + \int_{t_0}^t \frac{\partial f}{\partial y} G_{t_0}(\tau) d\tau\end{aligned}\quad (1.15)$$

Dies ist äquivalent zum Anfangswertproblem

$$\begin{aligned}\dot{G}(t_0)(t) &= \frac{\partial f}{\partial y} G_{t_0}(t) \\ G_{t_0}(t_0) &= -f(t_0, y(t_0), p)\end{aligned}\quad (1.16)$$

zur VDE nach dem Anfangszeitpunkt.

**c) Differenziere nach den Parametern:**

$$\begin{aligned}\frac{\partial}{\partial p} y(t) &= \frac{\partial y_0}{\partial p} + \int_{t_0}^t \frac{\partial f}{\partial y}(\tau, y(\tau), p) \frac{\partial y}{\partial p}(\tau) + \frac{\partial f}{\partial p}(\tau, y(\tau), p) d\tau \\ G_p(t) &= \frac{\partial y_0}{\partial p} + \int_{t_0}^t \frac{\partial f}{\partial y} G_p(\tau) + \frac{\partial f}{\partial p} d\tau\end{aligned}\quad (1.17)$$

Dies ist äquivalent zum Anfangswertproblem

$$\begin{aligned}\dot{G}_p(t) &= \frac{\partial f}{\partial y}(t, y(t), p) G_p(t) + \frac{\partial f}{\partial p}(t, y(t), p) \\ G_p(t_0) &= \frac{\partial y_0}{\partial p}\end{aligned}\quad (1.18)$$

zur VDE nach den Parametern.

*Bemerkung 1.5:*

Die Anfangswertprobleme der VDEs (1.14), (1.16) und (1.18) hängen von der Lösung  $y(t)$  des „nominalen“ Anfangswertproblems (1.11) ab. Sie müssen also jeweils mit (1.11) in einem gemeinsamen System gelöst werden.

*Satz 1.6*

Es gilt:

$$G_{t_0}(t; t_0, y_0, p) = -G(t; t_0, y_0, p)f(t_0, y_0, p) \quad (1.19)$$

*Beweis:*

Multipliziere  $\dot{G} = \frac{\partial f}{\partial y}G$ ,  $G(t_0) = I$  von rechts mit  $-f(t_0, y_0, p)$ :

$$-\dot{G}f(t_0, y_0, p) = -\frac{\partial f}{\partial y}Gf(t_0, y_0, p)$$

$$\text{und } -G(t_0)f(t_0, y_0, p) = -f(t_0, y_0, p)$$

Also erfüllt  $y: = -Gf(t_0, y_0, p)$  die Differentialgleichung  $\dot{y} = \frac{\partial f}{\partial y}y$  und die Anfangsbedingung  $y(t_0) = -f(t_0, y_0, p)$ . Das ist die VDE für  $G_{t_0}$ , also ist  $y = -Gf(t_0, y_0, p) = G_{t_0}$

### Variationsdifferentialgleichungen für Richtungsableitungen

Gegeben sei eine Richtung  $\Delta y_0 \in \mathbb{R}^{n_y}$ . Die Richtungsableitung von  $y$  nach  $y_0$  in der Richtung  $\Delta y_0$  ist gegeben durch

$$\begin{aligned} \frac{\partial y}{\partial y_0}(t; t_0, y_0, p)\Delta y_0 &:= \left. \frac{\partial}{\partial h} y(t; t_0, y_0 + h\Delta y_0, p) \right|_{h=0} \\ &= \lim_{h \rightarrow 0} \frac{y(t; t_0, y_0 + h\Delta y_0, p) - y(t; t_0, y_0, p)}{h} \in \mathbb{R}^{n_y} \end{aligned} \quad (1.19a)$$

Die Richtungsableitung erfüllt

$$\begin{aligned} \frac{\partial}{\partial t}(G(t; t_0, y_0, p)\Delta y_0) &= \frac{\partial f}{\partial y}(t, y, p)(G(t; t_0, y_0, p)\Delta y_0) \\ G(t; t_0, y_0, p)\Delta y_0 &= I\Delta y_0 = \Delta y_0 \end{aligned} \quad (1.20)$$

Dies ist ein ( $n_y$ -dimensionales) VDE-Anfangswertproblem für jede Richtung  $\Delta y_0$ . hat man mehrere Richtungen, kann man diese spaltenweise in einer Richtungsmatrix  $S$  zusammenfassen:

$$S = (\Delta y_{0,1}, \dots, \Delta y_{0,n_s})$$

Zur Berechnung aller zugehörigen Richtungsableitungen muss man also lösen:

$$(\dot{G}S) = \frac{\partial f}{\partial y}(GS), \quad (GS)(t_0) = S \quad (1.21)$$

mit dem  $n_s$ -fachen Aufwand wie für eine Richtung. Für die gesamte Ableitung  $\frac{\partial y}{\partial y_0} = \frac{\partial y}{\partial y_0} I_{n_y \times n_y}$  braucht man  $n_y$  Richtungen, hat also den  $n_y$ -fachen Aufwand. Deshalb berechnet man Richtungsableitungen nicht durch Berechnen von  $\frac{\partial y}{\partial y_0}$  und anschließende Multiplikation mit den Richtungen, sondern durch Lösen von Richtungs-VDE.

Für Richtungsableitungen nach  $p$  analog:



$$\begin{aligned}\frac{\partial}{\partial t}(G_p(t; t_0, y_0, p)\Delta p) &= \frac{\partial f}{\partial y}(t, y, p)G_p(t; t_0, y_0, p)\Delta p + \frac{\partial f}{\partial p}(t, y, p) \\ G_p(t_0; t_0, y_0, p)\Delta p &= \frac{\partial y_0}{\partial p}\Delta p\end{aligned}\quad (1.22)$$

### Eigenschaften der Ableitungsmatrizen

Füge an der Stelle  $t_0 < s < t_{end}$  einen „Haltepunkt“ ein:

$$\dot{y} = f(t, y, p), \quad t \in [s, t_{end}], \quad y(t_0) = y_0$$

und weiter

$$\dot{y} = f(t, y, p), \quad t \in [s, t_{end}], \quad y(s) = y(s; t_0, y_0, p)$$

Nach dem Satz von Picard-Lindelöf ist die abschnittsweise Lösung die selbe wie für

$$\dot{y} = f(t, y, t), \quad t \in [t_0, t_{end}], \quad y(t_0) = y_0$$

*Definition 1.7: Wronski-Matrix*

$$W(t, s) := G(t; s, y_s, p)$$

*Satz 1.8: Eigenschaften der Wronski-Matrizen*

- $W(t, s)$  erfüllt  $\partial_t W(t, s) = \partial_y f(t, y, p)W(t, s)$ ,  $W(s, s) = I$
- $W(t, s)W(s, r) = W(t, r)$  (1.23)
- Für alle  $t, s, r \in [t_0, t_{end}]$  ist  $W(t, s)$  invertierbar und es gilt  $W(t, s)^{-1} = W(s, t)$  (1.24)
- Für beliebige  $t_1, \dots, t_n$  ist  $W(t_1, t_2)W(t_2, t_3) \cdots W(t_{n-1}, t_n) = W(t_1, t_n)$

*Wronski-Matrizen für Ableitungen nach Parametern:*

*Definition 1.9:*

$$W_p(t, s) := G_p(t; s, y_s, p)$$

*Satz 1.10*

$$W_p(t_{end}, t_0) = W(t_{end}, s)W_p(s, t_0) + W_p(t_{end}, s) \quad (1.25)$$

*Beweis:*

$$\begin{aligned}
 y(t_{end}) &= y(t_{end}; t_0, y_0, p) = y(t_{end}; s, y_s, p) \\
 y_s &= y(s) = y(s; t_0, y_0, p) \\
 W_p(t_{end}, t_0) &= \frac{\partial y}{\partial y_s}(t_{end}; s, y_s, p) \cdot \frac{\partial y}{\partial p}(s; t_0, y_0, p) + \frac{\partial y}{\partial p}(t_{end}; s, y_s, p) \\
 &= W(t_{end}, s)W_p(s, t_0) + w_p(t_{end}, s)
 \end{aligned}$$

*Anwendung:*

Das System ODE+VDE wird abschnittsweise auf den Teilintervallen  $[t_0, s]$ ,  $[s; t_{end}]$  gelöst, liefert

$$y(s), \quad W(s, t_0), \quad W_p(s, t_0)$$

und

$$y(t_{end}), \quad W(t_{end}, s), \quad W_p(t_{end}, s)$$

Dann kann man zusammensetzen:

$$\begin{aligned}
 W(t_{end}, t_0) &= W(t_{end}, s)W(s, t_0) \\
 W_p(t_{end}, t_0) &= W(t_{end}, s)W_p(s, t_0) + W_p(t_{end}, s)
 \end{aligned}$$

### Ableitungsmatrizen bei abschnittsweise definierter rechter Seite

$$\dot{y}(t) = \begin{cases} f_1(t, y, p) & \text{für } t < t_s \\ f_2(t, y, p) & \text{für } t > t_s \end{cases} \quad (1.26) y(t_0) = y_0$$

Wobei der Umschaltzeitpunkt  $t_0 < t_s < t_{end}$  als eindeutige einfache Nullstelle der Schaltbedingung

$$Q(t_s, y(t_s), p) = 0$$

gegeben sei.

*Satz 1.11:*

Dann gilt:

$$\begin{aligned}
 \frac{\partial y}{\partial y_0}(t_{end}; t_0, y_0, p) &= W(t_{end}, t_s) [I - (f_1(t_s, y(t_s), p) - f_2(t_s, y(t_s), p))] \\
 &\quad \left( \frac{\partial Q}{\partial t}(t_s, y(t_s)) \right)^{-1} \frac{\partial Q}{\partial y}(t_s, y(t_s)) \Big] W(t_s, t_0) \quad (1.27)
 \end{aligned}$$

*Beweis:*

$$\begin{aligned}
 y(t_{end}) &= y(t_{end}; t_s, y_s, p) \\
 y(t_s) &= y(t_s; t_0, y_0, p) \\
 \frac{\partial y}{\partial y_0} &= \frac{\partial y}{\partial t_s}(t_{end}; t_s, y_s, p) \frac{\partial t_s}{\partial y_s} \frac{\partial y}{\partial y_0}(t_s; t_0, y_0, p) \\
 &\quad + \frac{\partial y}{\partial y_s}(t_{end}) \left( \frac{\partial y(t_s)}{\partial y_0} + \left( \frac{\partial y(t_s)}{\partial t_s} \right) \frac{\partial t_s}{\partial y_s} \frac{\partial y}{\partial y_0}(t_s) \right)
 \end{aligned}$$

$t_s$  ist Endzeitpunkt des ersten und Anfangszeitpunkt des zweiten Intervalls.

$$\begin{aligned}
 0 &= Q(t_s, y(t_s), p) \\
 \Rightarrow 0 &= \frac{\partial Q}{\partial t} \frac{\partial t}{\partial y} + \frac{\partial Q}{\partial y} \Big|_{y=y_s} \\
 \Rightarrow \frac{\partial t_s}{\partial y_s} &= - \left( \frac{\partial Q}{\partial t} \right)^{-1} \frac{\partial Q}{\partial y_s} \Big|_{t=t_s, y=y_s} \quad (\text{Satz für implizite Funktionen})
 \end{aligned}$$

$$\frac{\partial y}{\partial t_s}(t) = -W(t, t-s) f_2(t_s, y_s, p) \quad (\text{Satz 1.6})$$

$$\begin{aligned}
 \frac{\partial y}{\partial y_0}(t_{end}) &= W(t_{end}, t_s) [I - (f_1(t_s, y_s, p) \\
 &\quad - f_2(t_s, y_s, p)) \left( \frac{\partial Q}{\partial t} \right)^{-1} \frac{\partial Q}{\partial y} \Big|_{t=t_s, y=y_s}] W(t_s, t_0)
 \end{aligned}$$

## Adjungierte Differentialgleichung

Will man Matrix-Matrix-Produkte von links an  $\frac{\partial y}{\partial y_0}$  oder  $\frac{\partial y}{\partial p}$  berechnen:

$$\begin{aligned}
 u^T \frac{\partial y}{\partial y_0} &\quad \text{mit } u \in \mathbb{R}^{n_y} \text{ oder} \\
 U^T \frac{\partial y}{\partial y_0} &\quad \text{mit } U \in \mathbb{R}^{n_y \times n}
 \end{aligned}$$

Berechnet man ebenfalls nicht zuerst  $\frac{\partial y}{\partial y_0}$  und multipliziert dann, sondern löst die adjungierte Differentialgleichung.

### Satz 1.12

Gegeben sei das ODE-AWP

$$\dot{y} = f(t, y, p), \quad y(t_0) = y_0 \quad (1.28)$$

Dann gilt: Integriert man die adjungierte Differentialgleichung (ADE)

$$\dot{\Lambda}(t)^T = -\Lambda(t)^T \frac{\partial f}{\partial y}(t, y, p) \quad (1.29)$$

rückwärts, d. h. ausgehend von  $T > t_0$  mit dem Anfangswert  $\Lambda(T) = I$ , dann gilt:

$$\frac{\partial y}{\partial y_0}(T) = \Lambda(t_0)^T \quad (1.30)$$

$$\frac{\partial y}{\partial p}(T) = \int_{t_0}^T \Lambda(t)^T \frac{\partial f}{\partial p}(t, y, p) dt \quad (1.31)$$

*Beweis:*

Es gilt:

$$\begin{aligned} \dot{y} - f(t, y, p) &= 0 \\ \Rightarrow \int_{t_0}^T \Lambda(t)^T (\dot{y} - f(t, y, p)) dt &= 0 \end{aligned} \quad (1.32)$$

Leite (1.32) nach  $y_0$  ab:

$$\begin{aligned} 0 &= \frac{\partial}{\partial y_0} \int_{t_0}^T \Lambda(t)^T (\dot{y} - f(t, y, p)) dt \\ &= \int_{t_0}^T \Lambda(t)^T \left( \frac{\partial \dot{y}}{\partial y_0} - \frac{\partial f}{\partial y} \frac{\partial y}{\partial y_0} \right) dt \end{aligned}$$

Partielle Integration:

$$\int_{t_0}^T \Lambda(t)^T \frac{\partial \dot{y}}{\partial y_0} dt = \left[ \Lambda(t)^T \frac{\partial y}{\partial y_0} \right]_{t_0}^T - \int_{t_0}^T \dot{\Lambda}(t)^T \frac{\partial y}{\partial y_0} dt$$

Daraus folgt:

$$\begin{aligned}
 0 &= \int_{t_0}^T \Lambda^T \left( \frac{\partial \dot{y}}{\partial y_0} - \frac{\partial f}{\partial y_0} \frac{\partial y}{\partial y_0} \right) dt \\
 &= \int_{t_0}^T \left( -\dot{\Lambda} - \Lambda^T \frac{\partial f}{\partial y} \frac{\partial y}{\partial y_0} \right) dt + \left[ \Lambda^T \frac{\partial y}{\partial y_0} \right]_{t_0}^T \\
 &= \int_{t_0}^T \underbrace{\left( -\dot{\Lambda}^T - \Lambda^T \frac{\partial f}{\partial y} \right)}_{=0 \text{ adj. DGL}} \frac{\partial y}{\partial y_0} dt + \underbrace{\Lambda(T)^T}_{=I \text{ (AW)}} \frac{\partial y}{\partial y_0}(T) - \Lambda(t_0)^T \underbrace{\frac{\partial y}{\partial y_0}}_{=I} \\
 &= \frac{\partial y}{\partial y_0}(T) - \Lambda(t_0)^T \\
 \Rightarrow \Lambda^T(t_0) &= \frac{\partial y}{\partial y_0}(T)
 \end{aligned}$$

Leite (1.32) nach  $p$  ab mit analoger partieller Integration:

$$\begin{aligned}
 0 &= \int_{t_0}^T \Lambda^T \left( \frac{\partial \dot{y}}{\partial p} - \frac{\partial f}{\partial y} \frac{\partial y}{\partial p} - \frac{\partial f}{\partial p} \right) dt \\
 &= \int_{t_0}^T \underbrace{\left( -\dot{\Lambda}^T - \Lambda^T \frac{\partial f}{\partial y} \right)}_{=0} \frac{\partial y}{\partial p} dt - \int_{t_0}^T \Lambda^T \frac{\partial f}{\partial p} dt + \underbrace{\Lambda^T}_{=I} \frac{\partial y}{\partial p}(T) - \Lambda^T(t_0) \underbrace{\frac{\partial y}{\partial p}(t_0)}_{=0} \\
 &= - \int_{t_0}^T \Lambda^T \frac{\partial f}{\partial p} dt + \frac{\partial y}{\partial p}(T) \\
 \Rightarrow \frac{\partial y}{\partial p}(T) &= \int_{t_0}^T \Lambda^T \frac{\partial f}{\partial p} dt
 \end{aligned}$$

### Bemerkung 1.13

Zur Berechnung von  $\Lambda^T$  muss man erst vorwärts (1.28) lösen und  $y$  berechnen, die Werte von  $y$  dabei zwischenspeichern und dann rückwärts (1.29) lösen, um  $\Lambda^T$  zu berechnen.

### Ableitung von Linearkombinationen von $\frac{\partial y}{\partial y_0}$

Gegeben:  $u \in \mathbb{R}^{n_y}$  (adjungierte Richtung). Die Linearkombinations-ADE

$$\partial_t(u^T \Lambda(t)^T) = -(u^T \Lambda(t)^T) \frac{\partial f}{\partial y}(t, y, p) \quad (1.23)$$

## Inhaltsverzeichnis

mit dem Anfangswert

$$(u^T \Lambda(T)^T) = u^T \quad (1.34)$$

hat die Lösung

$$u^T \Lambda(t_0)^T = u^T \frac{\partial y}{\partial y_0}(T) \quad (1.35)$$

$$\text{und } \int_{t_0}^T (u^T \Lambda(t)^T) \frac{\partial f}{\partial p}(t, y, p) dt = u^T \frac{\partial y}{\partial p}(T) \quad (1.36)$$

D. h. pro Linearkombination der Zeilen von  $\frac{\partial y}{\partial y_0}$  und  $\frac{\partial y}{\partial p}$  muss nur eine Rückwärts-AWP gelöst werden.

### Anwendung:

$$\begin{aligned} \Phi(y(T; t_0, y_0, p)) & \quad \text{„Zielfunktion“} \\ \Phi : \mathbb{R}^{n_y} & \rightarrow \mathbb{R} \end{aligned}$$

Gradient:

$$\nabla_p \Phi(y(T; t_0, y_0, p)) = \frac{\partial \Phi}{\partial y} \frac{\partial y}{\partial p}(T; t_0, y_0, p)$$

benötigt nur eine adjungierte Richtung  $\frac{\partial \Phi}{\partial y} \in \mathbb{R}^{n_y}$

### Zusammenfassung:

- Variationsdifferentialgleichung
  - Vorwärtsdifferentiation
  - Richtungsableitungen
- Adjungierte Differentialgleichung
  - Rückwärtsdifferentiation
  - Linearkombinationen

## Simulation und Optimierungsprobleme bei Differentialgleichungen

- *Simulation:* Löse die Mathematischen Modellgleichungen, in dieser Vorlesung: Integration von ODE/DAE-AWPn

- *Optimierungsprobleme:*
  - *Parameterschätzung:* Bestimme die Modellparameter  $p$  so, dass Modell und Realität möglichst gut übereinstimmen.
  - *Modelldiskriminierung:* Bestimme durch Experimente, welche Modellvariante die Realität besser beschreibt.
  - *Optimale Versuchsplanung:* Bestimme Experimente, aus denen die Parameter möglichst signifikant geschätzt werden können.
  - *Optimales Design:* Berechne, wie ein System, Gerät etc. nach einem bestimmten Ziel optimal gebaut werden soll.
  - *Optimale Steuerung:* Berechne, wie ein Prozess nach einem bestimmten Ziel optimal durchgeführt werden soll, typischerweise: Minimale Kosten oder maximale Ausbeute.
  - *Optimale modellbasierte Regelung:* Wie ändert sich die optimale Steuerung bei Störungen des Systems?





## 2. Formulierung von Parameterschätzproblemen

englisch: Parameter Estimation

Beispiel: Chemische Reaktion („Bimolekulare Katalyse“), zwei Stoffe reagieren zu einem dritten:



Es gibt zwei sogenannte Reaktionspfade, einen mit Katalysator und einen ohne Katalysator, beide Pfade laufen parallel ab und die Konzentrationsänderungen durch die beiden Pfade werden addiert. Wir nennen den Reaktionsgeschwindigkeitskoeffizienten  $k_1$  für die Reaktion ohne Katalysator bzw.  $k_2$  für die Reaktion mit Katalysator.

$$k_1 = f_1 \exp\left(-\frac{E_1}{RT}\right)$$

Die Wirkung des Katalysators nimmt mit der Zeit ab, das wird durch eine Exponentialfunktion beschrieben:

$$k_2 = c_{kat} \exp(-\lambda t) f_2 \exp\left(-\frac{E_2}{RT}\right)$$

Reaktionsgeschwindigkeit:  $r = k_1 c_1 c_2 + k_2 c_1 c_2$

Daraus ergibt sich ein DGL-System mit Anfangsbedingungen:

$$\begin{aligned}\dot{n}_1 &= -V(k_1 + k_2) \frac{n_1}{V} \frac{n_2}{V} \\ \dot{n}_2 &= -V(k_1 + k_2) \frac{n_1}{V} \frac{n_2}{V} \\ \dot{n}_3 &= V(k_1 + k_2) \frac{n_1}{V} \frac{n_2}{V} \\ n_1(0) &= n_{1,0} \\ n_2(0) &= n_{2,0} \\ n_3(0) &= 0\end{aligned}$$

Wobei  $n_1$  die Anzahl Moleküle des ersten Stoffes beschreibt und  $V$  das Gesamt-Volumen aller an dem Experiment beteiligter Stoffe.

Der Prozessverlauf wird bestimmt von Größen, die von dem Experimentator eingestellt werden:  $T, V, n_{1,0}, n_{2,0}$ . Diese nennen wir „Steuergrößen“. Außerdem hängt der

Verlauf auch von den Größen  $f_1, E_1, f_2, E_2, \lambda$  ab. Diese Größen sind durch Naturgesetze, Stoffeigenschaften etc. bestimmte Größen, diese nennen wir "Parameter", sie sind nicht zeitabhängig.

*Experiment:* Wähle Steuerungen, führe Prozess durch, erhebe Messdaten und Messfehler. Die Messdaten hier liefert eine Apparatur zur Messung von C.

*Parameterschätzung:* Bestimme die Werte der unbekannten Modellparameter so, dass die Simulation die Messwerte möglichst gut beschreibt.

## Problemformulierung

### Prozessmodell

$$\dot{y} = f(t, y(t), \bar{p}) \quad y(t_0) = y_0(\bar{p}) \quad (2.1)$$

### Modell für Beobachtungen

- Nicht notwendigerweise verschiedene Messzeitpunkte  $t_i, i = 1, \dots, M$
- Messwerte  $\eta_i \in \mathbb{R}$  und zugehörige Modellantworten  $h_i(t_i, y(t_i), \bar{p}) \in \mathbb{R}$
- Messfehler  $\varepsilon_i \in \mathbb{R}$  mit Standardabweichung  $\sigma_i \in \mathbb{R}_+$

$\bar{p}$  seien die wahren, aber unbekannten Werte der Parameter.

*Annahme:* Die Messfehler seien unabhängige, additive und normalverteilte Zufallsgrößen mit bekannter Standardabweichung:

$$\eta_i = h_i(t_i, y(t_i), \bar{p}) + \varepsilon_i \quad (2.2)$$

$$\varepsilon_i \sim \mathcal{N}(0, \sigma_i^2) \quad (2.3)$$

Erwartungswert 0 bedeutet, dass es keinen systematischen Messfehler gibt.

*Bemerkung:* Da die Messzeitpunkte nicht notwendig verschieden sind, kann es an einem Zeitpunkt  $t$  auch mehrere Messungen geben. Auch die Messfunktionen  $h_i$  sind nicht notwendig verschieden.

Beispiele für die Standardabweichung  $\sigma$  :

- absolute Messfehler:  $\sigma_i \equiv \sigma$
- relative Messfehler:  $\sigma_i = \frac{x}{100} |h_i(t_i, y(t_i), \bar{p})| \sim \frac{x}{100} |\eta_i|$

Wichtig: Zur Beschreibung von Messdaten benötigt man Messwert  $\eta$  und „Genauigkeit“ (Standardabweichung)  $\sigma$ . Wenn man die wahren Parameter  $\bar{p}$  nicht kennt kann man trotzdem immernoch die Residuen  $\eta_i - h_i(t_i, y(t_i), \bar{p})$ ,  $i = 1, \dots, M$  oder die gewichteten Residuen  $\sigma_i^{-1}(\eta_i - h_i(t_i, y(t_i), \bar{p}))$ ,  $i = 1, \dots, M$  (2.4) betrachten.

*Ziel der Parameterschätzung:* passe Modell an die Daten an.

*Möglicher Ansatz:* Maximum Likelihood, d. h. suche die Parameter, unter denen die beobachteten Daten die höchste Wahrscheinlichkeit haben. Das ist der ML-Schätzer  $\hat{p}$ :

$$\hat{p} = \operatorname{argmax}_p L(p)$$

Mit der Likelihood-Funktion  $L(p)$ , die die bedingte Wahrscheinlichkeit der Daten in Abhängigkeit von den Parametern angibt. Oft betrachtet man die sogenannte log-Likelihood-Funktion  $\log L$ . Für unabhängige, normalverteilte Messfehler gilt:

$$\begin{aligned} \hat{p} &= \operatorname{argmax}_p \log L(p) \\ &= \operatorname{argmax}_p \log \prod_{i=1}^M \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp \left( -\frac{1}{2} \left( \frac{\eta_i - h_i(t_i, y(t_i), p)}{\sigma_i} \right)^2 \right) \\ &= \operatorname{argmax}_p \sum_{i=1}^M \log \left[ \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp \left( -\frac{1}{2} \left( \frac{\eta_i - h_i(t_i, y(t_i), p)}{\sigma_i} \right)^2 \right) \right] \\ &= \operatorname{argmax}_p \sum_{i=1}^M -\log \sqrt{2\pi\sigma_i^2} + \log \exp \left( -\frac{1}{2} \left( \frac{\eta_i - h_i(t_i, y(t_i), p)}{\sigma_i} \right)^2 \right) \\ &= \operatorname{argmax}_p \sum_{i=1}^M -\frac{1}{2} \left( \frac{\eta_i - h_i(t_i, y(t_i), p)}{\sigma_i} \right)^2 - \sum_{i=1}^M \log \sqrt{2\pi\sigma_i^2} \\ &= \operatorname{argmax}_p -\frac{1}{2} \sum_{i=1}^M \left( \frac{\eta_i - h_i(t_i, y(t_i), p)}{\sigma_i} \right)^2 \\ &= \operatorname{argmin}_p \frac{1}{2} \sum_{i=1}^M \left( \frac{\eta_i - h_i(t_i, y(t_i), p)}{\sigma_i} \right)^2 \quad (2.5) \end{aligned}$$

Also minimiert  $\hat{p}$  die  $\|\cdot\|_2$ -Norm des Residuen-Vektors. Gleichzeitig müssen  $y$  und  $p$  das AWP (2.1) erfüllen.

*Parameterschätzproblem:*

$$\min_p \frac{1}{2} \sum_{i=1}^M \left( \frac{\eta_i - h_i(t_i, y(t_i), p)}{\sigma_i} \right)^2$$

So dass gilt:  $\dot{y} = f(t, y(t), p)$ ,  $y(t_0) = y_0(p)$  (2.6).

Wichtig: Es werden immer Gewichte verwendet.

- Wenn man keine hat, macht man die Annahme, dass alle  $\sigma_1 \equiv 1$
- Ein gemeinsamer (positiver) Faktor an den verwendeten Gewichten ändert das Problem nicht.

## Inhaltsverzeichnis

- Das Problem (2.6) und seine Lösung hängen sehr von den Gewichten ab.
- Jede Wahl der Gewichte macht eine spezielle Annahme über die Statistik der Messfehler:  $N(0, \sigma_i^2)$ -verteilt.

*Schreibweisen:*

$$\begin{aligned}\eta &= \begin{pmatrix} \eta_1 \\ \vdots \\ \eta_M \end{pmatrix} \\ h = h(y, p) &= \begin{pmatrix} h_1(t_1, y(t_1), p) \\ \vdots \\ h_M(t_M, y(t_M), p) \end{pmatrix} \\ \varepsilon &= \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_M \end{pmatrix} \\ S &= \begin{pmatrix} \sigma_1^2 & & 0 \\ & \ddots & \\ 0 & & \sigma_M^2 \end{pmatrix}\end{aligned}$$

Dann gilt:  $\varepsilon \sim N(0, S) \Rightarrow S^{-1/2}\varepsilon \sim N(0, I)$

Das Zielfunktional von (2.6) minimiert

$$\frac{1}{2} \|S^{-\frac{1}{2}}(\eta - h)\|_2^2 = \frac{1}{2} (\eta - h)^T S^{-1} (\eta - h) \quad (2.7)$$

$S$  heißt Kovarianzmatrix der Messfehler

$$S = E(\varepsilon \varepsilon^T) \quad \text{„Erwartungswert von } \varepsilon \varepsilon^T \text{“}$$

Für unabhängige normalverteilte Messfehler ist  $S$  diagonal und positiv definit. Allgemeiner für unabhängige (korrelierte) multinomialverteilte Messfehler ist  $S$  symmetrisch und positiv definit.

$$S = \begin{pmatrix} \sigma_1^2 & s_{ij} \\ s_{ji} & \sigma_M^2 \end{pmatrix}$$

mit Varianzen  $\sigma_i^2$  und Kovarianzen  $s_{ij}$ . Zusätzlich definieren wir Korrelationskoeffizienten:

$$r_{ij} = \frac{s_{ij}}{\sigma_i \sigma_j} \quad \text{Korrelation } -1 \leq r_{ij} \leq 1$$

$$R = \begin{pmatrix} \sigma_1^{-1} & & 0 \\ & \ddots & \\ 0 & & \sigma_M^{-1} \end{pmatrix} S \begin{pmatrix} \sigma_1^{-1} & & 0 \\ & \ddots & \\ 0 & & \sigma_M^{-1} \end{pmatrix}$$

$$= \begin{pmatrix} 1 & & r_{ij} \\ & \ddots & \\ r_{ji} & & 1 \end{pmatrix}$$

R heißt Korrelationsmatrix. Für nicht-diagonales  $S$  ist die ML-Schätzung gegeben durch

$$\min(\eta - h(y, p))^T S^{-1}(\eta - h(y, p)) \quad (2.8)$$

Transformation: Zerlege  $S = LDL^T$  und minimiere folgendes:

$$\begin{aligned} \frac{1}{2}(\eta - h)^T S^{-1}(\eta - h) &= \frac{1}{2}(\eta - h)L^{-T}D^{-\frac{1}{2}}D^{-\frac{1}{2}}L^{-1}(\eta - h) \\ &= \frac{1}{2}\|D^{-\frac{1}{2}}L^{-1}(\eta - h)\|_2^2 \quad (2.9) \end{aligned}$$

$$D^{-\frac{1}{2}}L^{-1}(\eta - h(y, \bar{p})) \sim \mathcal{N}(0, I)$$

Andere Messverteilungen:

Dichte:

$$\exp\left(-\left(\frac{|\varepsilon|}{\sigma}\right)^q\right) \quad (2.10)$$

Schätzer:

$$\min \|S^{-\frac{1}{2}}(\eta - h(y, p))\|_q \quad (2.11)$$

Beispiele:

„Robuste Parameterschätzung“

$$q = 1$$

$$\min \sum_{i=1}^M \left| \frac{\eta_i - h_i(y, p)}{\sigma_i} \right| \quad (2.12)$$

„Worst-Case-Parameterschätzung“

$$q = \infty$$

$$\min \max_{i=1, \dots, M} \left\{ \left| \frac{\eta_i - h_i(y, p)}{\sigma_i} \right| \right\} \quad (2.13)$$

Nebenbedingung des Parameterschätzproblems:

- $y, p$  müssen die ODE-Modellgleichungen erfüllen:  $\dot{y} = f(t, y, p)$   
Variante:  $y, z, p$  müssen ein DAE erfüllen  $\dot{y} = f(t, y, z, p)$ ,  $0 = g(t, y, z, p)$
- Anfangsbedingung: Konsistenzbedingung  $y(t_0) = y_0(p)$ , im DAE-Fall zusätzlich  $0 = g(t_0, y_0, z(t_0), p)$
- Randbedingungen  $r(y(t_0), y(t_{end}), p) = 0$  oder auch  $r(y(t_0), y(t_1), \dots, y(t_{K-1}), y(t_{end}), p) = 0$

Beispiele und Spezialfälle:

- *Periodizität*:  $y(t_0) = y(t_{end})$ .
- *Innere-Punkt-Bedingung*:  $r(y(t_i), p) = 0$ .
- *Linear gekoppelte Randbedingungen*:  $r(y(t_0), p) + r(y(t_1), p) + \dots + r(y(t_{end}), p) = 0$ .
- *Ungleichsbedingungen*:  $s(y(t_0), \dots, y(t_k), p) \geq 0$ , z. B. Vorzeichenbedingung  $p_i \geq 0$  für einige  $i$ , Schranken  $a_i \leq p_i \leq b_i$ , Zustandsbeschränkungen  $\bar{y}_i \geq y(t_i) \geq \underline{y}_i$

In einem effizienten numerischen Code sollten solche speziellen Strukturen berücksichtigt werden.

Bemerkung zu Ungleichsbedingungen: Wir betrachten PS-Probleme, bei denen man ein physikalisches Modell an experimentelle Daten fitten will. Typischerweise wird das Modell durch Gleichungen beschrieben: DGL-System, Anfangs- und Rand-Bedingungen. Ungleichungen werden dagegen vom Modellierer formuliert mit irgendwie willkürlichen Grenzen. Wenn wir ein PS-Problem mit Ungleichsbedingungen gelöst haben kann folgendes passieren:

- Im Lösungspunkt sind die Ungleichungen inaktiv, d. h. der Lösungspunkt erfüllt die echte Ungleichung  $s(y(t_0), \dots, y(t_k), p) > 0$ . Dann können wir die Ungleichung weglassen.
- Im Lösungspunkt sind Ungleichungen aktiv, aber die zugehörigen Lagrange-Multiplikatoren sind Null. Dann können wir die Ungleichung ebenfalls weglassen.
- Eine Ungleichung ist aktiv mit Lagrange-Multiplikator ungleich Null, d. h. wenn man die Ungleichung weglässt wird die Zielfunktion besser, d. h. die Daten werden besser gefittet. Der Schätzer wird dann nicht nur durch die Daten sondern auch durch die vom Modellierer festgelegten Grenzen bestimmt. Das ist im allgemeinen physikalisch nicht sinnvoll.

Es kann sinnvoll sein, während der Algorithmus noch nicht terminiert ist, Grenzen zu fordern, um Auswertbarkeit des Modells zu gewährleisten. Im Lösungspunkt sollten diese Grenzen nicht aktiv sein. Wir behandeln bis auf weiteres Gleichungsbeschränkte PS-Probleme.

Allgemeine Problemformulierung:

$$\min_{p,y} \frac{1}{2} \sum \left( \frac{\eta_i - h_i(t_i, y(t_i), p)}{\sigma_i^2} \right)^2 \quad (2.14)$$

so dass

$$\begin{aligned} \dot{y}(t) &= f(t, y(t), p), & y(t_0) &= y_0(p) \\ 0 &= r(y(t_0), \dots, y(t_k), p) \end{aligned}$$

oder mit DAE

$$\begin{aligned} \dot{y}(t) &= f(t, y, z, p), & y(t_0) &= y_0(p) \\ 0 &= g(t, y, z, p) \end{aligned}$$

## Lösungsmethoden

### Parametrisierung der Lösung des AWP durch Single Shooting, Multiple Shooting oder Kollokation

Dadurch wird (2.14) endlichdimensional:

$$\min \frac{1}{2} \|F_1(x)\|_2^2 \quad (2.15)$$

so dass  $F_2(x) = 0$  mit  $x \in \mathbb{R}^n$  geeignet.

### Löse (2.15) mit verallgemeinertem Gauß-Newton-Verfahren





## 3. Shooting-Verfahren und Kollokation

### 3.1 Single-Shooting: Einfeldschießverfahren

Vorgehensweise:

- Wähle Werte für die Parameter  $p$  („Initial Guess“)
- Wir lösen das AWP  $\dot{y} = f(t, y, p)$ ,  $y(t_0) = y_0(p)$  (3.1) mit einem numerischen Verfahren und erhalten eine Darstellung der Lösung  $y(t; t_0, y_0, p)$
- Setze die Lösung an den Messzeitpunkten in die Modellantwortsfunktionen ein und berechne  $F_{1,i}(p) = \sigma_i^{-1}(\eta_i - h_i(t_i, y(t_i; t_0, y_0, p), p))$ ,  $i = 1, \dots, M$  (3.2). Setze die Lösung außerdem an den Randbedingungspunkten in die Randbedingung ein:  $F_2(p) = r(y(t_0; t_0, y_0, p), \dots, y(t_k; t_0, y_0, p), p)$  (3.3). Halte dazu den Integrator an den Punkten  $t_i$  an oder benutze die fehlerkontrollierte kontinuierliche Ausgabe.
- Das ergibt ein endlichdimensionales nichtlineares Ausgleichsproblem, nämlich  $\min \frac{1}{2} \|F_1(p)\|_2^2$  so dass  $F_2(p) = 0$  (3.4).
- Löse diese mit einer geeigneten Methode. Kriterien: (3.5)
  - iterativ, da das Problem nichtlinear ist
  - sollte unzulässige Iterierte erlauben, d. h. Zwischenwerte, bei denen  $F_2(p) \neq 0$
  - sollte für Least-Squares-Zielfunktion geeignet sein

Wir benutzen daher das verallgemeinerte Gauß-Newton-Verfahren. Dieses benötigt folgende Ableitungen:

$$J_1(p) := \frac{\partial}{\partial p} F_1(p)$$
$$J_2(p) := \frac{\partial}{\partial p} F_2(p)$$

In jeder Iteration des Gauß-Newton-Verfahrens muss also das AWP gelöst und  $F_1$  und  $F_2$  ausgewertet werden.

*Bemerkung 3.1* berechnung von  $J_1$  und  $J_2$ .

$$\begin{aligned}
 J_1(p) &:= \frac{\partial F_1}{\partial p}(p) \\
 &= \frac{\partial}{\partial p} \left( \frac{\eta_i - h_i(\dots)}{\sigma_i} \right)_{i=1, \dots, M} \\
 &= \left( -\frac{1}{\sigma_i} \left( \frac{\partial h_i}{\partial y}(\dots) \frac{\partial y}{\partial p}(t_i; t_0, y_0, p) + \frac{\partial h_i}{\partial p}(t_i; t_0, y_0, p) \right) \right)_{i=1, \dots, M} \quad (3.6)
 \end{aligned}$$

$$\begin{aligned}
 J_2(p) &= \frac{\partial F_2}{\partial p}(p) \\
 &= \sum_{i=0}^k \frac{\partial r}{\partial y_i} \frac{\partial y}{\partial p}(t_i; t_0, y_0, p) + \frac{\partial r}{\partial p} \quad (3.7)
 \end{aligned}$$

Dazu berechnet man  $\frac{\partial y}{\partial p} =: G_p$  als Lösung der VDE

$$\dot{G}_p = \frac{\partial f}{\partial y} G_p + \frac{\partial f}{\partial p} \quad (3.8)$$

sowie die Ableitungen

$$\frac{\partial f}{\partial y}, \frac{\partial f}{\partial p}, \frac{\partial h_i}{\partial y}, \frac{\partial h_i}{\partial p}, \frac{\partial r}{\partial y_i}, \frac{\partial r}{\partial p}$$

der Modellfunktion per Hand, durch numerische Differentiation, oder durch automatische Differentiation.

### Algorithmus 3.2: Verallgemeinertes Gauß-Newton-Verfahren

Zur Lösung von  $\min_p \frac{1}{2} \|F_1(p)\|_2^2$  s. t.  $F_2(p) = 0$

- Start mit einer Startschätzung  $p^0$ ,  $k = 0$  („Initial Guess“)
- Solange ein Abbruchkriterium verletzt ist:
  - Berechne  $\delta p^k$  durch Lösung des linearisierten Ausgleichsproblems  $\min \frac{1}{2} \|F_1(p^k) + J_1(p^k) \delta p\|_2^2$  s. t.  $F_2(p^k) + J_2(p^k) \delta p = 0$  (3.9)
  - Bestimme eine Schrittweite  $\alpha^k$ , z. B. durch Linesearch
  - Iteriere  $p^{k+1} = p^k + \alpha^k \delta p^k$  (3.10)

Mögliches Abbruchkriterium:  $\|\delta p^k\| \leq \varepsilon$ . Mehr zu Gauß-Newton-Verfahren in Kapitel 4.

### Algorithmus 3.3: Single-Shooting Gauß-Newton

- Start mit einer Startschätzung  $p^0$ ,  $k = 0$
- Solange ein Abbruchkriterium verletzt ist:

- Integriere das AWP (3.1) zusammen mit der VDE (3.8) für  $p = p^k$
- Halte an den Punkten  $t_i$  an und werte  $F_1(p^k)$  und  $F_2(p^k)$  gemäß (3.2) und (3.3) aus. Berechne  $J_1(p^k)$  und  $J_2(p^k)$  gemäß (3.6) und (3.7)
- Berechne  $\delta p^k$  durch Lösen des linearen Ausgleichsproblems (3.9).
- Berechne eine Schrittweite  $\delta p^k$  und iteriere gemäß (3.10)

Implementierung: Praktische Aufgabe 1

Benötigte Bestandteile für die Implementierung:

- Integrator für AWP/VDE: entweder durch Integrator mit Interner Numerischer Integration („IND“), siehe Kapitel 5. Oder durch Integrieren des Systems

$$\begin{pmatrix} \dot{y} \\ \dot{G}_p \end{pmatrix} = \begin{pmatrix} f \\ \frac{\partial f}{\partial y} G_p + \frac{\partial f}{\partial p} \end{pmatrix}, \quad y(t_0) = \begin{pmatrix} y_0(p) \\ \frac{\partial y_0}{\partial p}(p) \end{pmatrix}$$

- Löser für lineare Ausgleichsprobleme  $\min \frac{1}{2} \|F_1 + J_1 \delta x\|_2^2$  s. t.  $F_2 + J_2 \delta x = 0$ . KKT-Bedingung:

$$\exists \lambda: \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix} \begin{pmatrix} \delta x \\ \lambda \end{pmatrix} = - \begin{pmatrix} J_1^T F_1 \\ F_2 \end{pmatrix}$$

- Globalisierungs-Strategie: BT-Linesearch oder  $\alpha^j = 1$

### Bemerkung 3.4

Wenn wir keine Randbedingung  $r(\dots) = 0$  haben und die Differentialgleichung eine ODE und keine DAE ist dann lösen wir in jeder Iteration des Gauß-Newton-Verfahrens („GN-Verfahren“) für die aktuellen Parameter das komplette Simulationsproblem. Das Ausgleichsproblem ist dann unbeschränkt:  $\min_p \frac{1}{2} \|F_1(p)\|_2^2$ . Das linearisierte Ausgleichsproblem  $\min_{\Delta p} \frac{1}{2} \|F_1 + \partial_1 \Delta p\|_2^2$  hat die Lösung  $\Delta p = -(J_1^T J_1)^{-1} J_1^T F_1$ . Berechnung über QR-Zerlegung von  $J_1$ .

*Schwierigkeiten beim Single-Shooting:*

- In Satz 1.2 (Trompetenabschätzung) haben wir gesehen, dass kleine Störungen der Parameter sehr große Änderungen der Lösung  $y$  des AWP und damit auch große Änderungen des Zielfunktions-Wertes und Nebenbedingungen des Parameter-Schätz-Problems zur Folge haben können. Wir sind dann „weit weg“ von der Lösung und das GN-Verfahren konvergiert nicht, siehe auch Kapitel 4.
- Wenn die Ableitungen  $\frac{\partial f}{\partial y}$  nicht auf ganz  $[t_0, t_{end}] \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_p}$  beschränkt bleiben kann es sein, dass für Parameter-Werte weit weg von den wahren Parametern die Lösung des AWP nicht auf dem ganzen Intervall  $[t_0, t_{end}]$  existiert. Dann ist das PS-Problem nicht auswertbar.
- Die Vorinformation über die Trajektorien, die durch die Messwerte gegeben ist wird nicht genutzt.

## 3.2 Multiple Shooting, die Mehrzielmethode

Zerlege das Integrationsintervall in Teilintervalle  $t_0 = \tau_0 < \tau_1 < \dots < \tau_m = t_{end}$ . Die  $\tau_i$  nennt man Mehrzielknoten. Löse das AWP nur auf diesen Intervallen  $i = 0, \dots, m-1$ :  $\dot{y} = f(t, y, p)$ ,  $t \in [\tau_i, \tau_{i+1}]$  (3.10) mit den Anfangsbedingungen  $y(\tau_i) = s_i$  (3.11). Erhalte die Lösung  $y(t; \tau_i, s_i, p)$ ,  $t \in [\tau_i, \tau_{i+1}]$  (3.12). Setze diese  $y$  in die Messfunktion und Randbedingungs-Funktion an den entsprechenden Zeitpunkten  $t_i$  ein. Die  $s_i$ ,  $i = 0, \dots, m-1$  sind zusätzliche Variablen des Problems. An den Mehrzielknoten  $\tau_i$  formulieren wir zusätzliche Stetigkeitsbedingungen, sog. „Anschlussbedingungen“:

$$s_{i+1} + y(\tau_{i+1}; \tau_i, s_i, p) = 0 \quad i = 0, \dots, m-2 \quad (3.13)$$

Für DAEs ist  $s_i = (s_i^y, s_i^z)$  und im Punkt  $\tau_i$  müssen die Konsistenzbedingungen  $g(\tau_i, s_i^y, s_i^z, p) = 0$ ,  $i = 0, \dots, m-1$  erfüllt sein (3.14). Anschlussbedingungen braucht man dann nur für die  $s_i^y$ . Das ergibt insgesamt das beschränkte, nichtlineare Ausgleichsproblem

$$\min_x \frac{1}{2} \|F_1(x)\|_2^2 \quad (3.15)$$

so dass  $F_2(x) = 0$

- mit den Variablen  $x = (s_0, \dots, s_{m-1}, p)$  (3.16)
- den Zielfunktionstermen

$$F_{1,j}(x) = \frac{1}{\sigma_j} (\eta_j - h_j(t_j, y(t_j; \tau_{i_j}, s_{i_j}, p), p)) \quad (3.17)$$

mit  $t_j \in [\tau_{i_j}, \tau_{i_j+1}]$ ,  $j = 1, \dots, M$

- den Nebenbedingungen  $F_2(x)$ , die bestehen aus:
  - den Randbedingungen:  $r(y(t_0; \tau_0, s_0, p), \dots, y(t_k; \tau_{i_k}, s_{i_k}, p), p) = 0$  (3.18) mit  $t_j \in [\tau_{i_j}, \tau_{i_j+1}]$ ,  $i = 1, \dots, k$
  - den Anschlussbedingungen  $s_{i+1} + y(\tau_{i+1}; \tau_i, s_i, p) = 0$ ,  $i = 0, \dots, m-2$  (3.19)
  - und für DAEs den Konsistenzbedingungen  $g(\tau_i, s_i^y, s_i^z, p) = 0$ ,  $i = 0, \dots, m-1$  (3.20).

Die Lösungen der Probleme (3.4) (Single-Shooting) und (3.15) (Multiple-Shooting) sind identisch. Das Problem (3.15) hat aber mehr Variablen.

Nachteile von Multiple-Shooting:

- Wesentlich höherer Programmieraufwand.
- Mehraufwand bei der Bestimmung von Startwerten für die Variablen
- Höherdimensionales Optimierungsproblem

Vorteile von Multiple-Shooting:

- Die Existenz einer (unstetigen) „Start-Trajektorie“ auf dem gesamten Intervall  $[t_0, t_{end}]$  ist gesichert.
- Oft können die Multiple-Shooting-Variablen  $s_i$  durch Messwerte initialisiert werden.
- Man kann viel näher an der Lösung  $(s_0^*, s_1^*, \dots, s_{m-1}^*, p)$  des Problems starten. Der Einfluss schlechter Startwerte für die  $p$  wird abgemildert. Die Chance auf Konvergenz des GN-Verfahrens ist höher.
- Die Nichtlinearität des Problems wird reduziert: Bei Single-Shooting  $y(t_{end}; t_0, y_0, p)$  (3.21), bei Multiple-Shooting  $y(t_{i+1}; \tau_i, s_i, p)$  (3.22). Nach der Trompetenabschätzung ist die Nichtlinearität in (3.21) i. A. größer als in (3.22).

Zur Lösung von (3.15) verwenden wir wieder das verallgemeinerte GN-Verfahren und benötigen dazu die Ableitungen  $J_1 = \partial_x F_1$  und  $J_2 = \partial_x F_2$  (3.23). Diese haben die folgende Gestalt:

$$\begin{pmatrix} J_1 \\ J_2 \end{pmatrix} = \dots\dots\dots (3.22a)$$

mit

$$\begin{aligned} D_1^i &= \frac{\partial F_1}{\partial s_i} & i = 0, \dots, m-1 \\ D_1^p &= \frac{\partial F_1}{\partial p} & \text{Ableitung der ZF-Terme} \\ D_2^o &= \frac{\partial r}{\partial s_i} & i = 0, \dots, m-1 \\ D_2^p &= \frac{\partial r}{\partial p} & \text{Ableitung der Randbedingungen} \end{aligned}$$

Ableitung der Anschlussbedingungen:

$$\begin{aligned} -s_{i+1} + y(\tau_{i+1}; \tau_i, s_i, p) &= 0 & i = 0, \dots, m-2 \\ \text{nach } s_i: \frac{\partial}{\partial s_i} y(\tau_{i+1}; \tau_i, s_i, p) &=: G_i \\ &\text{nach } s_{i+1}: -I \\ \text{nach } p: \frac{\partial}{\partial p} y(\tau_{i+1}; \tau_i, s_i, p) &=: G_i^p & (\text{sonst } 0) \end{aligned}$$

Ableitung der Konsistenzbedingungen:

$$H_i := \frac{\partial \{ \partial s_i (\tau; i, s_i^y, s_i^z, p) \}}{g} \quad i = 0, \dots, m-1$$

$$H_i^p = \frac{\partial}{\partial p} g(\tau_i, s_i^y, s_i^z, p) \quad i = 0, \dots, m-1$$

Lösungsverfahren müssen diese spezielle Struktur ausnutzen („Kondensierung“, siehe Kapitel 4). Die D-Matrizen können mit der Kettenregel berechnet werden, z. B.

$$\frac{\partial F_1}{\partial p} = \sum_{j=1}^M \frac{\partial F_1}{\partial y(t_j)} \frac{\partial y(t_j)}{\partial p} + \frac{\partial F_1}{\partial p}$$

$$\frac{\partial F_1}{\partial s_i} = \sum_{j=1}^M \frac{\partial F_1}{\partial y(t_j)} \frac{\partial y(t_j)}{\partial s_i}$$

### Bemerkung 3.5

Der Aufwand zur Integration und zur Berechnung der  $G_i$  ist bei Single-Shooting und Multiple-Shooting im Wesentlichen gleich.

## 3.3 Kollokation

Literatur: Ascher, Mettheij, Russel: Numerical Solution of Boundary Value Problems for Ordinary Differential Equations

Biegler: Nonlinear Programming

### Kollokations-Diskretisierung

Approximiere die Lösung der ODE  $\dot{y} = f(t, y, p)$  (3.24) durch stückweise Polynome vom Grad  $k$  auf einem Gitter  $t_0 = \tau_0 < \tau_1 < \dots < \tau_{m-1} < \tau_m = t_{end}$  (3.25). Auf jedem dieser Teilintervalle wird die Lösung dargestellt durch

$$y_j^\tau(t; s_j) := \sum_{l=0}^k s_{jl} \chi_l \left( \frac{t - \tau_j}{\tau_{j+1} - \tau_j} \right) \quad (3.2)$$

$$t \in [\tau_j, \tau_{j+1}], \quad j = 0, \dots, m-1$$

Wobei  $\{\chi_l\}_{l=0, \dots, k}$  eine Basis des Polynomraums  $P_k([0, 1])$  der Polynome vom Grad  $\leq k$  ist

Die Koeffizienten  $s_{jl} \in \mathbb{R}^{n_y}$ ,  $j = 0, \dots, m-1$ ,  $l = 0, \dots, k$ ,  $s_j = (s_{j0}, \dots, s_{jk})$ ,  $j = 0, \dots, m-1$  (3.27) sind Variablen und bestimmt durch diese Bedingungen:

- Die approximierte Lösung  $y_j^\tau$  soll auf einer Unterteilung des Gitters die ODE erfüllen:

$$t_{jl} = \bar{c}_j + \rho_l h_j \quad l = 1, \dots, k, \quad j = 0, \dots, m-1 \quad (3.28)$$

(Kollokationspunkte)  $\rho_l \in [0, 1]$ ,  $h_j = \tau_{j+1} - \tau_j$ . Also:

$$\dot{y}_j^\tau(t_{jl}; s_j) = f(t_{jl}, y^\tau(t_{jl}, s_j), p), \quad j = 0, \dots, m-1 \quad (3.29)$$

- Die approximierte Lösung  $y^\tau$  soll an den Gitterpunkten  $\tau_j$  stetig sein:

$$y_j^\tau(\tau_{j+1}; s_j) = y_{j+1}^\tau(\tau_{j+1}, s_{j+1}) \quad j = 0, \dots, m-2 \quad (3.30)$$

- Anfangsbedingung:

$$y^\tau(\tau_0; s_0) = y_0(p) \quad (3.31)$$

Das sind  $mk$  Kollokationsbedingungen,  $m-1$  Stetigkeitsbedingungen und eine Anfangsbedingung, also  $m(k+1)$  Bedingungen für die  $m(k+1)$  Variablen  $s_{jl}$ .

### Wahl der Polynombasis und der Kollokationspunkte

z. B. B-Splines, Hermite-Splines. Wir benutzen die Runge-Kutta-Basis:

$$y_j^\tau(t; s_j) = s_{jl} + h_j \sum_{l=1}^k s_{jl} \Psi_l \left( \frac{t - \tau_j}{\tau_{j+1} - \tau_j} \right) \quad (3.32)$$

mit  $\Psi_l(0) = 0$

$$\text{und } \dot{\Psi}_l(\rho_i) = \begin{cases} 1 & i = l \\ 0 & \text{sonst} \end{cases}$$

$$\Psi_l \in \mathcal{P}_k([0, 1])$$

Dann lauten die Kollokationsbedingungen:

$$\begin{aligned} \dot{y}_j^\tau(t_{jl}; s_j) &= \dot{y}_j^\tau(\tau_j + \rho_l h_j; s_j) \\ &= f(t_{jl}, y^\tau(t_{jl}, s_j), p) \quad l = 1, \dots, k \end{aligned} \quad (3.33)$$

und die Stetigkeitsbedingung:

$$y_{j-1}^\tau(\tau_j; s_{j-1} - s_{j0}) = 0 \quad (3.34)$$

Die Basis  $\{1, \Psi_1, \dots, \Psi_k\}$  heißt Runge-Kutta-Basis, weil die  $s_{jl}$ ,  $l = 1, \dots, k$  dabei die Stufen des Runge-Kutta-Schemas

$$\begin{pmatrix} \rho_1 | & \Psi_1(\rho_1) & \cdots & \Psi_k(\rho_1) \\ \vdots & \vdots & & \vdots \\ \rho_k | & \Psi_1(\rho_k) & \cdots & \Psi_k(\rho_k) \\ & \Psi_1(1) & \cdots & \Psi_k(1) \end{pmatrix} \quad (3.35)$$

$$\text{mit } \Psi_l(\rho) = \int_0^\rho L_l(\bar{\rho}) d\bar{\rho}$$

$$\Psi_l(1) = \int_0^1 L_l(\rho) d\rho$$

$$\text{und } L_l(\rho) = \prod_{i \neq l} \frac{\rho - \rho_i}{\rho_l - \rho_i}$$

sind

Man kann also in jedem Intervall  $[\tau_j, \tau_{j+1})$   $y_j^\tau(\tau_{j+1}, s_j)$  als Ergebnis eines Schrittes eines impliziten Runge-Kutta-Verfahrens, gestartet bei  $y_j^\tau(\tau_j, s_j)$  auffassen: „Kollokationsschema“

### Konsistenzfehler:

$$y(\tau_{j+1}) = y(\tau_j) + h_j \sum_{l=1}^k f(t_j + \rho_l h_j, y(\tau_j + \rho_l h_j), p) \int_0^1 L_l(\rho) d\rho + h_j \mathcal{O}(h_j^q) \quad (3.36)$$

Die Konsistenzordnung  $q$  hängt ab von der Wahl der  $\rho_l$ ,  $l = 1, \dots, k$ . Für Gauß-Punkte (Nullstellen der Legendre-Polynome) erhält man die maximal mögliche Ordnung  $q = 2(k-1) + 2 = 2k$ . Für die Lobatto-Punkte erhält man die Ordnung  $q = 2k - 2$ , „Radau-Kollokations-Schema“.

### Beispiel Impliziter Euler, k=1

$$y_j^\tau(t; s_j) = s_{j0} + h_j s_{j1}$$

Kollokationsbedingung:

$$s_{j1} = f(\tau_{j+1} + h_j, s_{j0} + h_j s_{j1}, p)$$

Stetigkeitsbedingung:

$$s_{j-10} + h_{j-1} s_{j-1} - s_{j0} = 0$$



## Kollokation für beschränkte Parameterschätzprobleme

Approximiere die Lösung der ODE durch eine Kollokationsdiskretisierung  $y_j^T(t; s_j)$ ,  $j = 0, \dots, m-1$  Setze diese an den Messzeitpunkten in die Least-Squares-Terme und an den Randbedingungs-Punkten in die Randbedingung ein. Variablen sind:

$$x = (s_0, s_1, \dots, s_{m-1}, p) \in \underbrace{\mathbb{R}^{(k+1)n_y} \times \dots \times \mathbb{R}^{(k+1)n_y}}_m \times \mathbb{R}^{n_p}$$

$F_1$  besteht auf den Least-Squares-Termen,  $F_2$  besteht aus den Randbedingungen, den Kollokationsbedingungen (3.29), den Stetigkeitsbedingungen (3.30) und den Anfangsbedingungen (3.31). Es ergibt sich wieder ein beschränktes nichtlineares Ausgleichsproblem:

$$\min_x \frac{1}{2} \|F_1(x)\|_2^2 \quad s. t. \quad F_2(x) = 0 \quad (3.37)$$

Zur Lösung mit dem verallgemeinerten Gauß-Newton-Verfahren benötigen wir wieder

$$J_1 = \frac{\partial F_1}{\partial x}$$

$$J_2 = \frac{\partial F_2}{\partial y}$$

### Bemerkung 3.6: Eigenschaften von $\begin{pmatrix} \partial_1 \\ \partial_2 \end{pmatrix}$

- Es müssen keine Variations-DGL gelöst werden. Die Ableitungen nach  $s$  und  $p$  ergeben sich allein aus den Ableitungen der Kollokationspolynome und Ableitungen von  $f$ , Messfunktionen und Randbedingungen.
- Das System ist sehr groß, aber auch sehr dünn besetzt.

### Vorteile von Kollokation

- Es werden keine Integratoren und Variations-DGL-Löser benötigt.
- Es können Standard-Sparse-Löser eingesetzt werden
- Für lineare DGL, lineare Messfunktion und lineare Randbedingungen ist das Ausgleichs-Problem linear.
- Simulationsproblem und Optimierungsproblem werden in einer einzigen Schleife gelöst: All-At-Once

### Nachteile von Kollokation

- Die DGL wird nicht adaptiv diskretisiert
- Das Problem ist sehr hochdimensional

## 3.4 Ansätze zur Optimierung von DGL-Modellen

Wir haben bisher kennengelernt:

- Single-Shooting („Black-Box-Ansatz“)
- Multiple Shooting
- Kollokation

Das Optimierungsproblem muss iterativ gelöst werden.

Die konzeptionellen Unterschiede sind:

- Beim Single-Shooting wird in jeder Optimierungsiteration die DGL des AWP komplett gelöst. Das ist der sogenannte sequentielle Ansatz.
- Bei Multiple Shooting wird die Differentialgleichung in jeder Optimierungsiteration nur auf Teilintervallen gelöst, das AWP wird relaxiert, d. h. wir lassen während der Iterationen unstetige Lösungen zu. Erst im Lösungspunkt wird das AWP gelöst, simultan mit Nebenbedingungen und der Optimalität. Das ist ein Beispiel für einen simultanen Ansatz.
- Bei Kollokation wird die DGL vollständig diskretisiert und gemeinsam mit dem Optimierungsproblem gelöst. Das nennt man den All-At-Once-Ansatz.

Unsere Optimierungsverfahren lassen unzulässige Iterierte zu. Die Lösungen dieser Formulierungen sind gleich, da die Formulierungen mathematisch äquivalent sind. Die Iterationen der Verfahren können sich aber stark unterscheiden. Es werden während der Optimierung Schritte in verschiedenen hochdimensionalen Suchräumen berechnet. Dies kann bessere Konvergenzeigenschaften bedeuten.

Verallgemeinerung: Lifting von beliebigen nichtlinearen Problemen in höherdimensionale Räume:

$$\begin{aligned}x^{16} &= 2 \\ \Leftrightarrow \\ x_1^2 - x_2 &= 0 \\ x_2^2 - x_3 &= 0 \\ x_3^2 - x_4 &= 0 \\ x_4^2 - 2 &= 0\end{aligned}$$

siehe Albersmeyer, Diehl: The lifted Newton Method and its Application in Optimization

### 3.5 Relaxierte Formulierung von DAEs

DAE:

$$\begin{aligned} \dot{y} &= f(t, y, z, p) \\ 0 &= g(t, y, z, p) \end{aligned} \quad \text{Annahme: Index 1}$$

Anfangsbedingung:  $y(t_0) = y_0, \quad z(t_0) = z_0$   
 Konsistenzbedingung:  $0 = g(t_0, y_0, z_0, p)$

Für numerische Lösungsverfahren werden konsistente Anfangswerte benötigt. Dafür gibt es zwei Ansätze:

- Berechnung konsistenter Anfangswerte durch Lösen der Konsistenzbedingung mit z. B. einem Newton-Verfahren. Kann Schwierigkeiten bereiten, weil das Newton-Verfahren zunächst nur lokal konvergiert und globalisiert werden muss, z. B. mit Homotopie-Ansätzen. Das muss vor jeder Integration der DAE erfolgen.
- Integration mit „inkonsistenten“ Anfangswerten, relaxierte Formulierung:

$$\begin{aligned} \dot{y} &= f(t, y, z, p) \\ 0 &= g(t, y, z, p) - \beta(t)g(t_0, y_0, z_0, p) \end{aligned} \quad (3.40)$$

$$\begin{aligned} y(t_0) &= y_0 \\ z(t_0) &= z_0 \end{aligned}$$

mit  $\beta(t_0) = 1$   
 z. B.  $\beta(t) \equiv 1$   
 oder  $\beta(t) = e^{-\alpha(t-t_0)}$

Das modifizierte Problem (3.40) ist automatisch konsistent und kann numerisch integriert werden. Interpretation: Die Nebenbedingungen geben eine Mannigfaltigkeit vor, auf der die Lösung liegt. In der relaxierten Formulierung ist diese Mannigfaltigkeit verschoben. Die Konsistenzbedingung  $g(t_0, y_0, z_0, p) = 0$  (3.41) löst man im übergeordneten Optimierungsproblem als Nebenbedingung mit.

Bei Single-Shooting wird die Konsistenzbedingung im Anfangszeitpunkt erfüllt, dafür wird das Gleichungssystem gelöst.

Bei Multiple-Shooting wird für jedes Mehrzielintervall eine relaxierte Konsistenzbedingung eingesetzt. In jedem Mehrzielknoten:  $g(\tau_i, s_i^y, s_i^z, p) = 0$  für  $i = 0, \dots, m-1$ .

Bei Kollokation wird kein Integrator verwendet. Die algebraischen Nebenbedingungen werden zusätzlich in den Kollokationspunkten gefordert. (in RK-Basis-Darstellung):

$$\begin{aligned}
 y_j^\tau(t_{jl}; s_j) &= s_{jl}^y \\
 &= f(t_{jl}, y_j^\tau(t_{jl}; s), s_{jl}^z, p) \\
 0 &= g(t_{jl}, y_j^\tau(t_{jl}; s_j), s_{jl}^z, p) \\
 \text{mit } s_{jl}^y &\cong \dot{y}(t_{jl}), \quad s_{jl}^z \cong z(t_{jl})
 \end{aligned}$$

Stetigkeitsbedingung nur für  $s^y$ :

$$y_{j-1}^\tau(\tau_j; s_{j-1}) - s_{j_0}^y = 0$$

Die Bestimmung konsistenter Werte für die algebraischen Gleichungen wird von der Simulationsaufgabe in die Ebene des Optimierungsproblems verlagert.

## 4. Verallgemeinerte Gauß-Newton-Verfahren

Problem:

$$\begin{aligned} & \min \frac{1}{2} \|F_1(x)\|_2^2 \text{ s. t. } F_2(x) = 0 \quad (4.1) \\ & \text{mit } F_1: D \subset \mathbb{R}^n \rightarrow \mathbb{R}^{m_1}, \quad F_1 \in C^2(D) \\ & \quad F_2: D \subset \mathbb{R}^n \rightarrow \mathbb{R}^{m_2}, \quad F_2 \in C^2(D) \\ & \quad n \geq m_2 \\ & \quad m_1 \geq n - m_2 \\ & \text{Jacobi-Matrizen: } J_1: = \frac{\partial F_1}{\partial x}, \quad J_2: = \frac{\partial F_2}{\partial x} \end{aligned}$$

- Unbeschränkter Fall:  $\min \frac{1}{2} \|F(X)\|_2^2$
- Spezialfall: nichtlineare Gleichung:  $F(x) = 0$

### Algorithmus 4.1 (Verallgemeinertes Gauß-Newton-Verfahren)

- Startpunkt  $x^0$ ,  $k := 0$
- Solange ein Abbruchkriterium verletzt ist (z. B.  $\|\Delta x\| > \varepsilon$ ):
  - Berechne

$$\begin{aligned} F_1^k &:= F_1(x^k) \\ F_2^k &:= F_2(x^k) \\ J_1^k &:= J_1(x^k) \\ J_2^k &:= J_2(x^k) \end{aligned}$$

- Löse das lineare Ausgleichsproblem

$$\min \frac{1}{2} \|F_1^k + J_1^k \Delta x\|_2^2 \text{ s. t. } F_2^k + J_2^k \Delta x = 0 \quad (4.2)$$

Lösung:  $\Delta x^k$

- Bestimme eine Schrittweite  $\alpha^k \in (0, 1]$
- Iteriere  $x^{k+1} := x^k + \alpha^k \Delta x^k$

„Newton“: löse iterativ, linearisiere in jeder Iteration.

„Gauß“: linearisiere innerhalb der Norm, löse das lineare Ausgleichsproblem.

## Bemerkung 4.2

Zur Globalisierung der Konvergenz kann z. B. Linesearch verwendet werden. Es ergeben sich gedämpfte Schritte  $\alpha^k < 1$ . In der Nähe der Lösung können Vollschrte  $\alpha^k = 1$  erwartet werden.

## Annahmen 4.3: Regularitätsannahmen

- (CQ) „Constraint Qualification“:  $\text{Rg} J_2 = m_2$  ( $J_2$  hat vollen Rang, die Nebenbedingungen sind widerspruchsfrei und nicht redundant).
- (PD) „Positive Definiteness“:

$$\text{Rg} \begin{pmatrix} J_1 \\ J_2 \end{pmatrix} = n \quad (4.4)$$

bedeutet: die nicht durch die Nebenbedingungen festgelegten Variablen können aus den experimentellen Daten eindeutig geschätzt werden.

## Lemma 4.4

Gelten (PD) und (CQ), dann gilt:

- Die Matrix  $J_1^T J_1$  ist positiv definit auf  $\ker J_2$
- Die Matrix

$$\begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}$$

ist regulär.

Beweis: Übungsaufgabe.

## Lösung der linearen Ausgleichsprobleme

### 1. Unbeschränkter Fall

$$\min \frac{1}{2} \|F_1 + J_1 \Delta x\|_2^2 \quad (4.5)$$

Es gelte (PD), d. h.  $J_2$  habe vollen Rang.  $\Delta x^*$  ist Lösung von (4.5) genau dann, wenn es das sogenannte Normalgleichungssystem löst:

$$J_1^T J_1 \Delta x^* + J_1^T F_1 = 0$$

$\Delta x^*$  ist Minimum genau dann wenn

$$\begin{aligned} & F_1 + J_1 \Delta x^* \perp \{J_1 \Delta x\} \\ \Leftrightarrow & (\Delta x^* J_1^T)(J_1 \Delta x^* + F_1) = 0 \forall \Delta x \\ \Leftrightarrow & \Delta x^* (J_1^T J_1 \Delta x^* + J_1^T F_1) = 0 \\ \Leftrightarrow & J_1^T J_1 \Delta x^* + J_1^T F_1 = 0 \\ \Leftrightarrow & \Delta x^* = -(J_1^T J_1)^{-1} J_1^T F_1 \\ & = -J^\dagger F_1 \end{aligned} \tag{4.6}$$

$J^\dagger$  heißt Moore-Penrose-Pseudoinverse und erfüllt die vier Moore-Penrose-Axiome:

- $(J^\dagger J)^T = J^\dagger J$
- $(J J^\dagger)^T = J J^\dagger$
- $J J^\dagger J = J$
- $J^\dagger J J^\dagger = J^\dagger$

Umgekehrt:  $J^\dagger$  ist durch die vier Axiome eindeutig bestimmt. Die Lösung der Normallengleichung ist eindeutig, wenn (PD) erfüllt ist.

### Bemerkung 4.10

Wenn  $\text{Rg} J_1 < n$ , also (PD) nicht erfüllt ist, dann ist die Lösung von  $\min \frac{1}{2} \|F_1 + J_1 \Delta x\|_2^2$  nicht eindeutig. Dann ist

$$\Delta x = -J^\dagger F_1$$

Die Lösung kleinster euklidischer Norm.  $J^\dagger$  erfülle dabei die vier Moore-Penrose-Axiome.  
Beweis: Übungsaufgabe

## 2. Beschränkter Fall

### Lemma 4.5

Gelten (CQ) und (PD) dann gilt

$\Delta x^* \in \mathbb{R}^n$  ist genau dann Lösung von  $\min \frac{1}{2} \|F_1 + F_1 \Delta x\|_2^2$ , s. t.  $F_2 + J_2 \Delta x = 0$  (4.6), wenn

$$\Delta x^* = - \begin{pmatrix} I & 0 \end{pmatrix} \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-1} \begin{pmatrix} J_1^T & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} \quad (4.7)$$

Sei  $\Delta x^*$  Minimum.

$$f(t) := \frac{1}{2} \|F_1 + J_1(\Delta x^* + t\Delta y)\|_2^2$$

mit  $\Delta y$  so dass

$$\begin{aligned} F_2 + J_2(\Delta x^* + t\Delta y) &= 0 \\ \Rightarrow J_2\Delta y &= 0 \\ \text{d. h. } \Delta y &\in \ker(J_2) \\ 0 &= f'(0) \\ &= \Delta y^T (J_1^T J_1 \Delta x^* + J_1^T F_1) \\ \Rightarrow J_1^T J_1 \Delta x^* + J_1^T F_1 &\in (\ker J_2)^\perp = \text{Bild}(J_2^T) \\ \Rightarrow \exists! \lambda \text{ mit } J_1^T J_1 \Delta x^* + J_1^T F_1 + J_2^T \lambda &= 0 \\ \text{außerdem } J_2 \Delta x^* + F_2 &= 0 \quad (\text{KKT-Bedingungen}) \\ \Rightarrow \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix} \begin{pmatrix} \Delta x^* \\ \lambda \end{pmatrix} &= - \begin{pmatrix} J_1^T & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} \\ \Rightarrow \Delta x^* &= - \underbrace{\begin{pmatrix} I & 0 \end{pmatrix} \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-1} \begin{pmatrix} J_1^T & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} F_1 \\ F_2 \end{pmatrix}}_{=: J^+ \quad (4.9)} \\ \Delta x^* &= -J^+ F \end{aligned} \quad (4.8)$$

Gelte umgekehrt (4.8) für  $\Delta x^*$  und  $\lambda$ . Dann folgt  $f'(0) = 0$  für alle  $\Delta y \in \ker(J_2)$ . Aus (PD) folgt, dass  $\Delta x^*$  ein Minimum ist.

#### Definition 4.6: Verallgemeinerte Inverse

Der Lösungsoperator  $J^+$  von (4.8) heißt verallgemeinerte Inverse von

$$J = \begin{pmatrix} J_1 \\ J_2 \end{pmatrix}$$

#### Lemma 4.7

$J^+$  erfüllt das Moore-Penrose-Axiom  $J^+ J J^+ = J^+$ .



Beweis: Satz

$$F = \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} = \begin{pmatrix} J_1 J^+ y \\ J_2 J^+ y \end{pmatrix}$$

in (4.6) ein:

$$\min \frac{1}{2} \|F_1 + J_1 \Delta x\|_2^2 \text{ s. t. } F_2 + J_2 \Delta x = 0$$

Erste Lösung:

$$\begin{aligned} \Delta x_1 &= -J^+ F \\ &= -J^+ (J J^+ y) \end{aligned}$$

Zweite Lösung:

$$\begin{aligned} \Delta x_2 &= -J^+ y \\ \text{NR: } \underbrace{J_1 J^+ y}_{=F_1} - J_1 J^+ y &= 0 \\ \underbrace{J_2 J^+ y}_{=F_2} - J_2 J^+ y &= 0 \end{aligned}$$

Wegen der Eindeutigkeit der Lösung ist  $\Delta x_1 = \Delta x_2$ .  $-J^+ J J^+ y = -J^+ y$  für beliebige  $y$ ,  $J^+ J J^+ = J^+$ .

#### Lemma 4.8

$J^+$  Erfüllt die Moore-Penrose-Axiome 1,2 und 4. Beweis: Übungsaufgabe.

#### Bemerkung 4.9

Die Moore-Penrose-Pseudoinverse für den beschränkten Fall würde das folgende Problem lösen:

$$\min_{\Delta x} \frac{1}{2} \left\| \begin{pmatrix} J_1 \\ J_2 \end{pmatrix} \Delta x + \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} \right\|_2^2$$

Das ist nicht äquivalent zu

$$\begin{aligned} \min_{\Delta x} \frac{1}{2} \|F_1 + J_1 \Delta x\|_2^2 \\ \text{s. t. } F_2 + J_2 \Delta x = 0 \end{aligned}$$

## Numerische Lösung

Man stellt nicht  $J^+$  oder  $J^\dagger$  auf, sondern zerlegt  $J_1$  und  $J_2$  geeignet.

### 1. Unbeschränkter Fall

$$J_1^T J_1 \Delta x = -J_1^T F_1$$

- 1. Variante: QR-Zerlegung von  $J_1 = QR = \overline{Q}\overline{R}$

$$\begin{aligned} J_1^T J_1 \Delta x &= \overline{R}^T \underbrace{\overline{Q}^T \overline{Q}}_{=I} \overline{R} \Delta x \\ &= -\overline{R}^T \overline{Q}^T F_1 \\ \Delta x &= -\overline{R}^{-1} \overline{Q}^T F_1 \\ J^\dagger &= \overline{R}^{-1} \overline{Q}^T \end{aligned}$$

- 2. Variante: Singulärwertzerlegung von  $J_1$ :

$$J_1 = U \Sigma V^T$$

mit  $U, V$  orthogonal und  $\Sigma$  Diagonalmatrix mit Singulärwerten von  $J_1$  auf der Diagonalen.

$$\begin{aligned} \Delta x &= -V \begin{pmatrix} \sigma_1^{-1} & & & \\ & \ddots & & \\ & & \sigma_n^{-1} & \\ & & & 0 \end{pmatrix} U^T F_1 \\ J^\dagger &= V \begin{pmatrix} \sigma_1^{-1} & & & \\ & \ddots & & \\ & & \sigma_n^{-1} & \\ & & & 0 \end{pmatrix} U^T \end{aligned}$$

### Beschränkter Fall

- 1. Variante („Bildraumvariante“):  $\Delta x$  ist Lösung von

$$\begin{pmatrix} J_q^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \lambda \end{pmatrix} = - \begin{pmatrix} J_1^T F_1 \\ F_2 \end{pmatrix}$$

Diese Matrix ist symmetrisch aber indefinit, also wende Gaußsche LR-Zerlegung an

- 2. Variante („Nullraumvariante“):

$$\min_{\Delta x} \frac{1}{2} \|F_1 + J_1 \Delta x\|_2^2 \quad (4.10) \quad \text{s. t.} \quad F_2 + J_2 \Delta x = 0 \quad (4.11)$$

1. Bestimme Lösungsmenge von (4.11) durch LR-Zerlegung von  $P_2 J_2$ :

$$P_2 J_2 = L_2 \begin{pmatrix} R_2 & D_2 \end{pmatrix} = \begin{pmatrix} P_2 J_{21} & P_2 J_{22} \end{pmatrix} \quad (4.12)$$

Mit  $R_2$  reguläre obere Dreiecksmatrix,  $L_2$  normierte untere Dreiecksmatrix,  $P_2$  Permutationsmatrix zur Zeilenpivotierung. Voraussetzung ist (CQ). Spalte  $\Delta x$  auf:

$$\begin{aligned} \Delta x &= \begin{pmatrix} \underbrace{\Delta y}_{m_2} & \underbrace{\Delta z}_{n-m_2} \end{pmatrix} \\ P_2 J_2 \Delta x &= L_2 (R_2 \Delta y + D_2 \Delta z) = -P_2 F_2 \\ \Rightarrow \Delta y &= -R_2^{-1} (D_2 \Delta z + L_2^{-1} P_2 F_2) \quad (4.13) \end{aligned}$$

2. Minimiere (4.10) auf der Lösungsmenge von (4.11). Spalte  $J_1$  analog auf.

$$J_1 = \begin{pmatrix} \underbrace{J_{11}}_{m_1 \times m_2} & \underbrace{J_{12}}_{m_1 \times (n-m_2)} \end{pmatrix}$$

$$\begin{aligned} \text{Einsetzen: } \frac{1}{2} \|F_1 + J_1 \Delta x\|_2^2 &= \frac{1}{2} \|F_1 + J_{11} \Delta y + J_{12} \Delta z\|_2^2 \\ &= \frac{1}{2} \left\| \underbrace{F_1 + J_{11} R_2^{-1} L_2^{-1} P_2 F_2}_{=: b} + \underbrace{(J_{12} - J_{11} R_2^{-1} D_2) \Delta z}_{=: B} \right\|_2^2 \\ &= \frac{1}{2} \|b + B \Delta z\|_2^2 \end{aligned}$$

QR-Zerlegung von  $B$ :

$$B = Q_1 R_1 = (\overline{Q}_1 \quad \hat{Q}_1) \begin{pmatrix} \overline{R}_1 \\ 0 \end{pmatrix} = \overline{Q}_1 \overline{R}_1 \quad (4.15)$$

$$B^T B \Delta z = \overline{R}_1^T \overline{Q}_1^T \overline{Q}_1 \overline{R}_1 \Delta z = -B^T b = -\overline{R}_1^T \overline{Q}_1^T b$$

$$\Delta z = -\overline{R}_1^{-1} \overline{Q}_1^T b \quad (4.16)$$

$$\text{Einsetzen: } \Delta y = -R_2^{-1} (D_2 \Delta z + L_2^{-1} P_1 F_2)$$

Formale Darstellung:

$$\begin{aligned}
 \begin{pmatrix} P_2 J_2 \\ \dots \\ J_1 \end{pmatrix} &= \begin{pmatrix} P_1 J_{21} & \vdots & P_2 J_{22} \\ \dots & & \dots \\ J_{11} & \vdots & J_{12} \end{pmatrix} \\
 &= \begin{pmatrix} L_2 & \vdots & 0 \\ \dots & & \dots \\ L_1 & \vdots & I \end{pmatrix} \begin{pmatrix} R_2 & \vdots & D_2 \\ \dots & & \dots \\ 0 & \vdots & B \end{pmatrix} \\
 &= \underbrace{\begin{pmatrix} L_2 & \vdots & 0 \\ \dots & & \dots \\ L_1 & \vdots & Q_1 \end{pmatrix}}_{=:T} \begin{pmatrix} R_2 & \vdots & D_2 \\ \dots & & \dots \\ 0 & \vdots & R_1 \end{pmatrix} \quad (4.17) \\
 L_1 &:= J_{11} R_2^{-1} \\
 \Rightarrow J_{12} &= L_1 D_2 + B = L_1 D_2 + Q_1 R_1
 \end{aligned}$$

Rechte Seite entsprechend:

$$- \begin{pmatrix} P_2 F_2 \\ F_1 \end{pmatrix} = T \begin{pmatrix} -L_2^{-1} P_2 F_2 \\ -Q_1^T b \end{pmatrix} \quad (4.18)$$

nachrechnen:

$$\begin{aligned}
 -L_2 L_2^{-1} P_2 F_2 &= -P_2 F_2 \\
 -L_1 L_2^{-1} P_2 F_2 - \underbrace{Q_1 Q_1^T}_{=:I} \underbrace{(F_1 - \overbrace{J_{11} R_2^{-1} L_2^{-1} P_2 F_2}^{=:L_1})}_{=:b} &= -F_1
 \end{aligned}$$

Lösen:

$$\begin{aligned}
 R_2 \Delta y + D_2 \Delta z &= -L_2^{-1} P_2 F_2 \\
 \bar{R}_1 \Delta z &= -\bar{Q}_1^T b \\
 \begin{pmatrix} R_2 & D_2 \\ 0 & \hat{R}_1 \end{pmatrix} \begin{pmatrix} \Delta y \\ \Delta z \end{pmatrix} &= -T^{-1} \begin{pmatrix} P_2 F_2 \\ F_1 \end{pmatrix} \\
 &= -T^{-1} \begin{pmatrix} 0 & P_2 \\ I & 0 \end{pmatrix} \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} \\
 \Delta x &= - \underbrace{\left[ \begin{pmatrix} R_2 & D_2 \\ 0 & \bar{R}_1 \end{pmatrix}^{-1} \vdots 0 \right]}_{=:J^+} T^{-1} \begin{pmatrix} 0 & P_2 \\ I & 0 \end{pmatrix} \underbrace{\begin{pmatrix} F_1 \\ F_2 \end{pmatrix}}_F \quad (4.19) \\
 &= -J^+ F
 \end{aligned}$$

**Lemma 4.11 (Berechnung der adjungierten**

$$\begin{aligned}\lambda &= -P_2 L_2^{-T} R_2^{-T} J_{11}^T (J_1 \Delta x + F_1) \\ &= -P_2 L_2^{-T} L_1^T (J_1 \Delta x + F_1) \quad (4.20)\end{aligned}$$

Beweis:  $\lambda$  erfüllt eindeutig die Gleichungen

$$\begin{aligned}J_1^T J_1 \Delta x + J_2^T \lambda &= -J_1^T F_1 \quad (\lambda \in \mathbb{R}^{m_2}) \\ \text{bzw. } J_1^T (J_1 \Delta x + F_1) &= -J_2^T \lambda\end{aligned}$$

Es reicht,  $m_2$  dieser Gleichungen zu betrachten:

$$\begin{aligned}J_{11}^T (J_1 \Delta x + F_1) &= -J_{21}^T \lambda = -R_2^T L_2^T P_2^{-1} \lambda \\ \Rightarrow \lambda &= -P_2 L_2^{-T} \underbrace{R_2^{-T} J_{11}^T}_{=L_1} (J_1 \Delta x + F_1)\end{aligned}$$

**3. Variante: Stoer 1979**

QR-Zerlegung von  $J_2^T$ :  $J_2 = L_2 Q_2$ . Transformation von rechts:

$$Q_1 \begin{pmatrix} J_2 \\ \dots \\ J_1 \end{pmatrix} Q_2^T = \text{TODO!!!!}$$

Rechte Seite transformieren mit  $Q_1$ ,  $\Delta \tilde{y}$ ,  $\Delta \tilde{z}$  ausrechnen:

$$\Delta x = Q_2^T \begin{pmatrix} \Delta \tilde{y} \\ \Delta \tilde{z} \end{pmatrix}$$

**Anwendung auf die Mehrzielmethode**

TODO!!!! Große Matrix

mit  $\Delta x = (\Delta)$ ,  $G$ ,  $G_i^p D_1^{\dot{c}}$ ,  $D_1^p$ ,  $D_2^i$ ,  $D_2^p$  wie in (3.23) und

$$r_1 = (\sigma_j^{-1}(\eta_j - h_j(t_j, y(t_j; \tau_{ij}, s_{ij}, p), p))), \quad j = 1, \dots, M, \quad t_j \in [\tau_{ij}, \tau_{ij+1}], \quad r_2 = r(y(t_o; \tau_0, s_0, p), \dots, y(t_K, \tau_K, s_K, p))$$

mit  $t_j \in [\tau_{ij}, \tau_{ij+1}]$ ,  $j = 1, \dots, K$

$$d_i = -s_{i+1} + y(\tau_{i+1}; \tau_i, s_i, p) \quad i = 0, \dots, m-2$$

Lösen:

- Wähle  $-I$  unten rechts als Blockpivot-Element. Multiplizieren unterste Blockzeile mit  $D_i^{m-1}$ ,  $i = 1, 2$  und addiere auf die  $i$ -te Zeile  $i = 1, 2$ .

- Es entsteht

$$D_i^0 D_i^1, \dots, \underbrace{(D_i^{m-2} + D_1^{m-1} \cdot G_{m-2})}_{(*)} \underbrace{(D_1^{m-1} + D_i^{m-1} \cdot (-I))}_{=0} (D_i^p + D_i^{m-1} \cdot G_{m-2}^p)$$

- Eliminiere  $(*)$  ebenso mit der zweituntersten Blockzeile usw.

Die führt auf die kompakte Rekursion:

#### Algorithmus 4.12 (Eliminationsalgorithmus)

$$\begin{aligned} u^{m-1} &:= \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} \\ E^{m-1} &:= \begin{pmatrix} D_1^{m-1} \\ D_2^{m-1} \end{pmatrix} \\ P^{m-1} &:= \begin{pmatrix} D_1^p \\ D_2^p \end{pmatrix} \end{aligned}$$

Für  $i = m-1, \dots, 1$ :

- $u^{i-1} := u^i + E^i d_{i-1}$
- $p^{i-1} := p^i + E^i G_{i-1}^p$
- $E^{i-1} := D^{i-1} + E^i G_{i-1}$

Es entstehen folgende Matrix und Vektor:

$$\begin{pmatrix} E^0 & 0 & \cdots & \cdots & 0 & P^0 \\ G_0 & -I & & & & G_0^p \\ & G_1 & -I & & & G_1^p \\ & & \ddots & \ddots & & \vdots \\ & & & G_{m-2} & -I & G_{m-2}^p \end{pmatrix} \begin{pmatrix} u^0 \\ d_0 \\ d_1 \\ \vdots \\ d_{m-2} \end{pmatrix} \quad (4.22)$$

$$\begin{aligned} E^0 &= \begin{pmatrix} E_1 \\ E_2 \end{pmatrix} \\ P^0 &= \begin{pmatrix} P_1 \\ P_2 \end{pmatrix} \\ u^0 &= \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \end{aligned}$$

**Algorithmus 4.13: Lösen**

- Löse beschränktes Ausgleichsproblem nur in den Variablen  $\Delta s_0$  und  $\Delta p$ :

$$\min_{\Delta s_0, \Delta p} \frac{1}{2} \|E_1 \Delta s_0 + P_1 \Delta p + u_1\|_2^2 \text{ s. t. } E_2 \Delta s_0 + P_2 \Delta p = 0$$

z. B. mit Variante 2.

- Berechne  $\Delta s_1, \dots, \Delta s_{m-1}$  durch:

Für  $i = 1, \dots, m-1$ :

$$\Delta s_i = G_{i-1} \Delta s_{i-1} + G_{i-1}^p \Delta p + d_{i-1} \quad (4.23)$$

**Korollar:**

Das Lösen der linearen beschränkten Probleme erfordert für Multiple Shooting im wesentlichen den selben Aufwand wie für Single Shooting.

**Bemerkung 4.14:**

- Es reicht,  $u$ ,  $P$  und  $E$  abzuspeichern, da im Normalfall  $u^i$ ,  $P^i$  und  $E^i$  nicht im Speicher gehalten werden müssen.
- Diese blockweise Gauß-Elimination kann als Dreieckszerlegung aufgefasst werden:

$$\begin{pmatrix} I & 0 & \dots & 0 & E^{m-1} \\ & \ddots & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & I \end{pmatrix} \begin{pmatrix} D^0 & D^1 & \dots & \dots & D^{m-1} & D^p \\ G_0 & -I & & & & G_0^p \\ & G_1 & -I & & & G_1^p \\ & & \ddots & \ddots & & \vdots \\ & & & G_{m-2} & -I & G_{m-2}^p \end{pmatrix} \\ = \begin{pmatrix} D^0 & D^1 & \dots & E^{m-2} & 0 & P^{m-2} \\ G_0 & -I & & & & G_0^p \\ & G_1 & -I & & & G_1^p \\ & & \ddots & \ddots & & \vdots \\ & & & G_{m-2} & -I & G_{m-2}^p \end{pmatrix}$$

Insgesamt:

$$\begin{pmatrix} I & E^1 & 0 & \dots & 0 \\ & \ddots & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & I \end{pmatrix} \dots \begin{pmatrix} I & 0 & \dots & 0 & E^{m-1} \\ & \ddots & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & I \end{pmatrix} (M Z M)$$

$$\begin{aligned}
 &= \begin{pmatrix} I & E^1 & \dots & E^{m-1} \\ & \ddots & & \\ & & \ddots & \\ & & & I \end{pmatrix} (M Z M) \\
 &\begin{pmatrix} E^0 & 0 & \dots & \dots & 0 & P^0 \\ G_0 & -I & & & & G_0^p \\ & G_1 & -I & & & G_1^p \\ & & \ddots & \ddots & & \vdots \\ & & & G_{m-2} & -I & G_{m-2}^p \end{pmatrix} \begin{pmatrix} u^0 \\ d_0 \\ d_1 \\ \vdots \\ d_{m-2} \end{pmatrix} \quad (4.24)
 \end{aligned}$$

Variante: Man kann auch Blockspalten-Elimination durch Transformationen von rechts durchführen.

$$\begin{aligned}
 &\begin{pmatrix} E^0 & 0 & \dots & \dots & 0 & P^0 \\ G_0 & -I & & & & G_0^p \\ & G_1 & -I & & & G_1^p \\ & & \ddots & \ddots & & \vdots \\ & & & G_{m-2} & -I & G_{m-2}^p \end{pmatrix} \\
 &= \begin{pmatrix} (D^0 + D^1 G_0) & D^1 & \dots & \dots & D^{m-1} & (D^p + D^1 D_0^p) \\ 0 & -I & 0 & \dots & \dots & 0 \\ (G_1 G_0) & G_1 & -I & 0 & \dots & (G_1^p + G_1 G_0^p) \\ & & \ddots & \ddots & & \vdots \\ & & & G_{m-2} & -I & G_{m-2}^p \end{pmatrix} \begin{pmatrix} I & & & & & \\ -G_0 & I & & & & -G_0^p \\ & & I & & & \\ & & & \ddots & & \\ & & & & I & \\ & & & & & I \end{pmatrix}
 \end{aligned}$$

Transformierte Variablen:

$$\begin{pmatrix} \Delta \tilde{s} \\ \Delta \tilde{p} \end{pmatrix} = \begin{pmatrix} I & & & & \\ -G_0 & I & & & -G_0^p \\ & & I & & \\ & & & \ddots & \\ & & & & I & \\ & & & & & I \end{pmatrix} \begin{pmatrix} \Delta s \\ \Delta p \end{pmatrix}$$

$\Delta \tilde{s}_1 := d_0$  kann gesetzt werden

Streiche zweite Blockzeile und zweite Blockspalte und Variable  $\Delta \tilde{s}_1$ . Nach dem Streichen hat das Restsystem die gleiche Struktur wie das Ausgangssystem. Fahre mit nächster Blockspalte fort usw. Das führt auf ein kondensiertes System:

$$\begin{pmatrix} E^{m-1} & P^{m-1} \end{pmatrix}$$



Aus diesem können  $\tilde{\Delta s}$ ,  $\tilde{\Delta p}$  berechnet werden durch Lösen des linearen Ausgleichsproblems  $\min \frac{1}{2} \|E_q^{m-1} \tilde{\Delta s}_0 + P^{m-1} \tilde{\Delta p}\|_2^2$ ,

Rücktransformation der Variablen liefert  $\Delta s_0, \dots, \Delta s_{m-1}, \Delta p$ .

Diese Elimination kann parallelisiert werden:

- Führe den Schritt für jede zweite Spalte aus.
- Streiche Zeilen, Spalten, Variablen; Matrix hat wieder Ausgangsstruktur.
- Algorithmus kann  $\frac{n}{2}$  Prozessoren beschäftigen
- Die Berechnung der  $G_i$ -Matrizen durch Lösen des Systems ODE/VDE kann auch im Rahmen dieser Parallelisierung erfolgen.
- Dadurch wird die „Wall-Time“-Berechnungszeit für Multiple Shooting sogar kürzer als für Single Shooting.
- Literatur: Gallitendörfer: Parallele Algorithmen für Optimierungsprobleme, Dissertation 1997
- Es gibt eine dritte Variante: Sukzessive QR-Elimination von rechts, ebenfalls parallelisierbar.

#### **Bemerkung 4.15**

Die strukturausnutzenden Zerlegungen für die „Mehrzielmatrizen“ nennen Kondensierung/Kondensung.



## 5. Lokale Konvergenz von Newton-Typ-Verfahren

Newton-Typ-Verfahren:

- Löse iterativ mit einem guten Startwert  $x_0 \in \mathbb{R}^n$
- Löse in jeder Iteration ein linearisiertes Problem
- Wende eine Globalisierungsstrategie an

$$F: \mathbb{R}^n \rightarrow \mathbb{R}^m \quad M(x): \text{Lösungsoperator des linearen Problems}$$

### Algorithmus 5.1 (Newton-Typ-Verfahren)

- Startwert  $x^0$ ,  $k := 0$
- Solange ein geeignetes Abbruchkriterium verletzt ist:
  - Berechne  $\Delta x^k := -M(x^k)F(x^k)$  (5.1)
  - Berechne  $\alpha^k$  auf einer Globalisierungsstrategie
  - Iteriere  $x^{k+1} := x^k + \alpha^k \Delta x^k$  (5.2)

Kann angewendet werden:

- Zur Bestimmung von Nullstellen von  $F(\cdot)$ ,  $m = n \Rightarrow$  Newton-Verfahren oder Quasi-Newton-Verfahren
- Insbesondere zur Bestimmung von Nullstellen von  $\Delta L(x, \lambda) = 0$  (notwendige Optimalitätsbedingung)  $\Rightarrow$  SQP-Verfahren.
- Zur Bestimmung von Lösungen unbeschränkter nichtlinearer Ausgleichsprobleme  $\min \frac{1}{2} \|F(x)\|_2^2$ ,  $m \geq n$
- Zur Bestimmung von Lösungen beschränkter nichtlinearer Ausgleichsprobleme  $\min \frac{1}{2} \|F_1(x)\|$  s. t.  $F_2(x) = 0$ ,  $m = m_1 + m_2 \geq n$ ,  $m_2 \leq n \Rightarrow$  Verallgemeinertes Gauß-Newton-Verfahren.

$$\text{Sei } J := \frac{\partial F}{\partial x} \quad (5.3) \text{ Jacobimatrix}$$

## Bemerkung 5.2

- Bei Newton-Verfahren ist  $M(x) = J(x)^{-1}$  die Inverse von  $J$ .
- bei Quasi-Newton-Verfahren ist  $M(x) \cong J(x)^{-1}$
- Bei SQP-Verfahren ist  $M(x, \lambda) = \nabla_{x,\lambda}^2 L(x, \lambda)^{-1}$  oder  $M(x, \lambda) \cong \nabla_{x,\lambda}^2 L(x, \lambda)^{-1}$
- Bei Gauß-Newton-Verfahren ist  $M(x) = (J(x)^T J(x))^{-1} J(x)^T$  die Moore-Penrose-Pseudoinverse.
- Bei Verallgemeinerten Gauß-Newton-Verfahren ist

$$M(x) = \begin{pmatrix} I & 0 \end{pmatrix} \begin{pmatrix} J_1(x)^T J_1(x) & J_2(x)^T \\ J_2(x) & 0 \end{pmatrix}^{-1} \begin{pmatrix} J_1(x)^T 0 \\ 0 & I \end{pmatrix}$$

die verallgemeinerte Inverse von  $J$ .

## Satz 5.3: Lokaler Kontraktionssatz (Bock 1987)

$$\text{Sei } F: D \subset \mathbb{R}^n \rightarrow \mathbb{R}^m, \quad F \in \mathbb{C}^1(D, \mathbb{R}^m), \quad J := \frac{\partial F}{\partial x}$$

Für alle  $x, y \in D$  mit  $y - x = -M(x)F(x)$  und  $\theta \in [0, 1]$  gelte:

- Es existiert ein  $\omega < \infty$  so dass  $\|M(y)(J(x + \theta(y - x)) - J(x))(y - x)\| \leq \omega \theta \|x - y\|^2$  (5.4)
- Es existiert ein  $\kappa(x) \leq \kappa < 1$ , so dass  $\|M(y)R(x)\| \leq \kappa(x)\|y - x\|$  (5.5) für das Residuum  $R(x) := F(x) - J(x)M(x)F(x)$
- Sei  $x_0 \in D$  gegeben mit  $\Delta x^j := -M(x^j)F(x^j)$ ,  $\delta_j := \kappa + \frac{\omega}{2}\|\Delta x^j\|$ ,  $\delta_0 = \kappa + \frac{\omega}{2}\|\Delta x^0\| < 1$  (5.6)
- 

$$D^0 := \left\{ z : \|z - z_0\| \leq \frac{\|\Delta x^0\|}{1 - \delta_0} \right\} \subset D$$

Dann gilt:

- a) Die Iterierten  $x^{j+1} = y^j + \Delta x^j$  sind wohldefiniert und bleiben in  $D^0$ .
- b) Es existiert ein  $x^* \in D^0$  so dass  $x^j \rightarrow x^*$  ( $j \rightarrow \infty$ )
- c)

$$\|x^{j+k} - x^*\| \leq \frac{\Delta x^0}{1 - s_j} s_j^k \text{ „hoch k“ (a-priori-Abschätzung)}$$

• d)

$$\|\Delta x^{j+1}\| \leq s_j \|\Delta x^j\| = \kappa \|\Delta x^j\| + \frac{\omega}{2} \|\Delta x^j\|^2 \quad (5.7)$$

Beweis:

$$\begin{aligned} \|\Delta x^{j+1}\| &= \|M(x^{j+1})F(x^{j+1})\| =: \|M^{j+1}F^{j+1}\| \\ &= \|M^{j+1}(F^{j+1} - F^j - J^j \Delta x^j) + M^{j+1}R^j\| \\ &\leq \|M^{j+1}\left(\int_0^1 J(x^j + t\Delta x^j)\Delta x^j dt - \int_0^1 J^j \Delta x^0 dt\right)\| + \|M^{j+1}R^j\| \\ &\leq \int_0^1 \|M^{j+1}(J(x^j + t\Delta x^j) - J^j)\Delta x^j\| dt + \|M^{j+1}R^j\| + \kappa \|\Delta x^j\| \\ &\leq \int_0^1 \omega t \|\Delta x^j\|^2 dt + \kappa \|\Delta x^j\| \\ &= \frac{\omega}{2} \|\Delta x^j\|^2 + \kappa \|\Delta x^j\| \\ &= \delta_j \|\Delta x^j\| \Rightarrow d) \end{aligned}$$

Zeige:  $(\delta_j)_{j \in \mathbb{N}}$ ,  $(\|\Delta x^j\|)_{j \in \mathbb{N}}$  sind monoton fallend:

Induktion:

$$\begin{aligned} \delta_j - \delta_{j+1} &= \frac{\omega}{2} (\|\Delta x^j\| - \|\Delta x^{j+1}\|) \\ &\geq \frac{\omega}{2} (\|\Delta x^j\| - \delta_j \|\Delta x^j\|) \\ &\geq \frac{\omega}{2} \|\Delta x^j\| (1 - \delta_j) > 0 \\ \|\Delta x^{k+k}\| &\leq \delta_{j+k} \|\Delta x^{j+k-1}\| \\ &\leq \delta_{j+k-1} \cdots \delta_j \|\Delta x^j\| \\ &\leq \delta_j^k \|\Delta x^j\| \\ \|x^{j+2} - x^0\| &\leq \|\Delta x^j + 1\| + \cdots + \|\Delta x^0\| \\ &\leq (\delta_0^{j+1} + \cdots + \delta s_0^1) \|\Delta x^0\| \\ &\leq \frac{1}{1 - \delta_0} \|\Delta x^0\| \Rightarrow a) \end{aligned}$$

$(x^j)_{j \in \mathbb{N}}$  ist Cauchy-Folge:

$$\begin{aligned}
 \|x^{i+j+1} - x^i\| &\leq \sum_{k=0}^j \|\Delta x^{i+k}\| \\
 &\leq \sum_{k=0}^j \delta_j^k \|\Delta x^i\| \\
 &\leq \sum_{k=0}^i \delta_0^{i+k} \|\Delta x^0\| \\
 &\leq \delta_0^i \left( \sum_{k=0}^{\infty} \delta_0^k \right) \|\Delta x^0\| \\
 &= \delta_0^i \frac{\|\Delta x^0\|}{1 - \delta_0} \rightarrow 0 (i \rightarrow \infty) \\
 &\Rightarrow x^j \rightarrow x^* \\
 D \text{ kompakt} &\Rightarrow x^* \in D^0 \Rightarrow b)
 \end{aligned}$$

Beweis:

$$\begin{aligned}
 \|x^j + k + i - x^{j+k}\| &\leq \frac{\|\Delta x^{j+k}\|}{1 - \delta_{j+k}} \\
 &\leq \delta_j^k \frac{\|\Delta x^j\|}{1 - \delta_j}
 \end{aligned}$$

$\forall i \geq 0$  also auch für  $i \rightarrow \infty$ :

$$\|x^* - x^{j+k}\| \leq \delta_j^k \frac{\|\Delta x^j\|}{1 - \delta_j} \Rightarrow c)$$

#### Korollar 5.4

Für  $F(x) = 0$ ,  $F: \mathbb{R}^n \rightarrow \mathbb{R}^m$  konvergiert das Newton-Verfahren mit  $M(x) = J(x)^{-1}$  lokal quadratisch.

Beweis:

$$\begin{aligned}
\|M^{j+1}R^j\| &= \|(J^{j+1})^{-1}(I - J^j(J^j)^{-1})F^j\| = 0 \Rightarrow \kappa = 0 \\
\|\Delta x^{j+1}\| &\leq \frac{\omega}{2}\|\Delta x^j\|^2 \\
\|x^{j+1} - x^*\| &= \|x^j - x^* + \Delta x^j\| = \|M^j(J^j(x^j - x^*) - (F^j - F^*))\| \quad (*): \text{ Im Lösungspunkt} \\
&= \left\| M^j \int_0^1 (J^j - J(x^j + t(x^j - x^*))) (x^j - x^*) dt \right\| \quad (\text{HDI}) \\
&\leq \underbrace{\|M^j J^*\|}_{\leq \Gamma} \int_0^1 \|M^*(J^j - J(x^j + t(x^j - x^*))) (x^j - x^*)\| dt \\
&\leq \Gamma \int_0^1 \omega t \|x^j - x^*\|^2 dt \\
&= \Gamma \frac{\omega}{2} \|x^j - x^*\|^2
\end{aligned}$$

### Bemerkung 5.5: Quasi-Newton-Verfahren

Für näherungsweise Newton-Verfahren („Quasi-Newton-Verfahren“) ist  $x^{j+1} = x^j - M(x^j)F(x^j)$  mit  $M(x^j) \cong J(x^j)^{-1}$   $\kappa > 0$ , zur Konvergenz muss  $\kappa < 1$  sein:

$$\begin{aligned}
\|M^{j+1}R^j\| &= \|M^{j+1}((M^j)^{-1} - J^j)M^jF^j\| \\
&= \|M^{j+1}((M^j)^{-1} - J^j)(x^{j+1} - x^j)\| \\
&\leq \kappa \|x^{j+1} - x^j\|
\end{aligned}$$

Notwendig für Konvergenz ist also

$$\|M^j\| \leq \gamma \text{ und } \|(M^j)^{-1} - J^j\| \text{ klein}$$

### Satz 5.6 (Dennis-Moré)

Sei  $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$  stetig differenzierbar. Betrachte die Iteration  $x^{j+1} = x^j + \Delta x^j$  und sei  $\Delta x^j$  gegeben durch  $\Delta x^j = -M(x^j)F(x^j)$ . Wir nehmen an, dass die Folge der  $x^j$  gegen einen Punkt  $x^*$  mit  $F(x^*) = 0$  konvergiert mit  $J(x^*)$  regulär. Dann konvergiert  $(x^j)_{j \in \mathbb{N}}$  Q-superlinear gegen  $x^*$ , d. h.  $\lim_{j \rightarrow \infty} \frac{\|x^{j+1} - x^*\|}{\|x^j - x^*\|} = 0$  genau dann wenn

$$\lim_{j \rightarrow \infty} \frac{\|(M(x^j)^{-1} - J(x^*))\Delta x^j\|}{\|\Delta x^j\|} = 0 \quad (5.8)$$

Beweis: siehe Vorlesung Algorithmische Optimierung 1.

## Varianten von Quasi-Newton-Verfahren

$$\Delta x^j = -M(x^j)F(x^j), \quad M(x^j) \cong J(x^j)^{-1}$$

- Berechne  $J(x^j)$  durch Differenzenquotienten
- Halte  $M(x^j)$  fest
  - für alle Iterationen:  $M(x^j) = J(x_0)^{-1}$
  - für einige Iterationen:  $M(x^j) = J(x^{\bar{j}})^{-1}$  solange  $\frac{\Delta x^{j+1}}{\|\Delta x^j\|} \leq \delta$  z. B.  $\delta = \frac{1}{4}$ , danach neues  $\bar{j} = j$
- Nähere  $J(x^j)$  bzw.  $M(x^j)$  durch Update-Formeln aus  $J(x^{j-1})$  bzw.  $M(x^{j-1})$  an, siehe unten.

## Bemerkung 5.7 (Bedeutung von $\omega$ )

$$\|M(y)(J(x+t(y-x)) - J(x))(x-y)\| \leq t\omega\|y-x\|^2$$

- Wenn  $M(y)$  in einer Umgebung von  $x^*$  beschränkt ist:  $\|M(y)\| \leq \gamma$ ,  $\gamma < \infty$  und
- $J$  eine Lippschitz-Bedingung erfüllt:  $\|(J(x+t(x-y)) - J(x))(y-x)\| \leq \beta t\|y-x\|^2$ ,  $\beta < \infty$ , dann ist  $\omega = \gamma\beta < \infty$  in einer Umgebung von  $x^*$

$\omega$  kann sehr groß werden wenn

- $\|M\|$  sehr groß ist, d. h.  $J$  ist fast singulär.
- die erste Ableitung von  $J$  bzw. die zweite Ableitung von  $F$  groß ist, d. h. das Problem ist sehr nichtlinear.

## Anwendung auf (verallgemeinerte) Gauß-Newton-Verfahren:

$$M(x) = J^+(x) = \begin{pmatrix} I & 0 \end{pmatrix} \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-1} \begin{pmatrix} J_1^T & 0 \\ 0 & I \end{pmatrix}$$

Es gelte (CQ) und (PD).  $\|M\|$  ist groß, wenn

$$\begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}$$

fast singulär ist.

Residuum:

$$\begin{aligned} R(x) &= F(x) - J(x)M(x)F(x) \\ &= F(x) + J(x)\Delta x \end{aligned}$$

Im Lösungspunkt:  $\Delta x^* = 0$ :  $R(x^*) = F(x^*)$ ,

$$\|R(x^*)\| = \|F_1(x^*)\|, \quad \text{da } F_2(x^*) = 0$$



**Bemerkung 5.8: Bedeutung von  $\kappa$**

Wenn  $M(x)$  stetig differenzierbar ist gilt in  $D^0$ :

$$\|M(y) - M(x)\| \leq L\|y - x\| \text{ und} \\ \|R(x)\| = \|F(x) - J(x)M(x)F(x)\| \leq \rho$$

$$\text{Au\ss}erdem \text{ ist } M(x)R(x) = \underbrace{(M(x) - M(x)J(x)M(x))}_{=0 \text{ (4. Moore-Penrose-Axiom)}} F(x) = 0$$

$$\|M(y)R(x)\| = \|(M(y) - M(x))R(x)\| \leq \rho L\|y - x\| \\ \kappa = \rho L$$