Optimale Steuerung 2

Kapitel 7: Kollokationsverfahren

Prof. Dr.-Ing. habil. Pu Li

Fachgebiet Prozessoptimierung

Fachgebiet Prozessoptimierung

Lösungsverfahren zur dynamischen Optimierung

Indirekte Verfahren:

- **Variationsverfahren, Optimalitätsbedingungen**
- **Das Maximum-Prinzip**
- **Dynamische Programmierung**
- **Riccati-Optimal-Regler**

Direkte Verfahren:

- **Methoden zur Diskretisierung, Orthogonale Kollokation**
- **Lösung mit nichtlinearen Programmierungsverfahren**
- **Simultane und Sequentielle Verfahren**

Lösungsverfahren zur dynamischen Optimierung

- **Lösungsverfahren**
  - Indirekte Methoden
    - Dynamische Programmierung
    - Maximum Prinzip
  - Direkte Methoden
    - Simultane Methoden
      - Diskretisierung
      - Gradienten
      - SQP
    - Sequentielle Methoden
      - Diskretisierung
      - Simulation
      - Gradienten
      - SQP

$$\min_{u(t)} \left\{ J\left[u(t), t_f\right] = G\left[x(t_f), t_f\right] + \int_{t_0}^{t_f} F(x,u,t)\,dt \right\}$$

mit
$$g(\dot{x}, x, u, t) = 0$$
$$h(\dot{x}, x, u, t) \le 0$$
$$x_{\min} \le x \le x_{\max}$$
$$u_{\min} \le u \le u_{\max}$$
$$x(t_0) = x_0$$

Problemdarstellung:

Das Gleichungssystem eines dynamischen Systems

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}), \quad \mathbf{x}(0) = \mathbf{x}_0, \quad 0 \le t \le t_f$$

Diskretisierungmethoden:

- Euler-Verfahren
- Orthogonale Kollokation
- BDF-Verfahren (Backward Differentiation Formulas)

An diskreten Zeitpunkten $t_1, \quad t_2, \quad \cdots \quad t_N$ werden die Variablen bewertet.

Welche Methode soll benutzt werden?

Wie groß soll die Schrittlänge sein?

Fachgebiet Prozessoptimierung

Das Euler-Verfahren:

Explizites Euler-Verfahren:

$$\frac{d\mathbf{x}}{dt} = \frac{\mathbf{x}^{k} - \mathbf{x}^{k-1}}{\Delta t} = \mathbf{f}\left(\mathbf{x}^{k-1}, t^{k-1}\right)$$

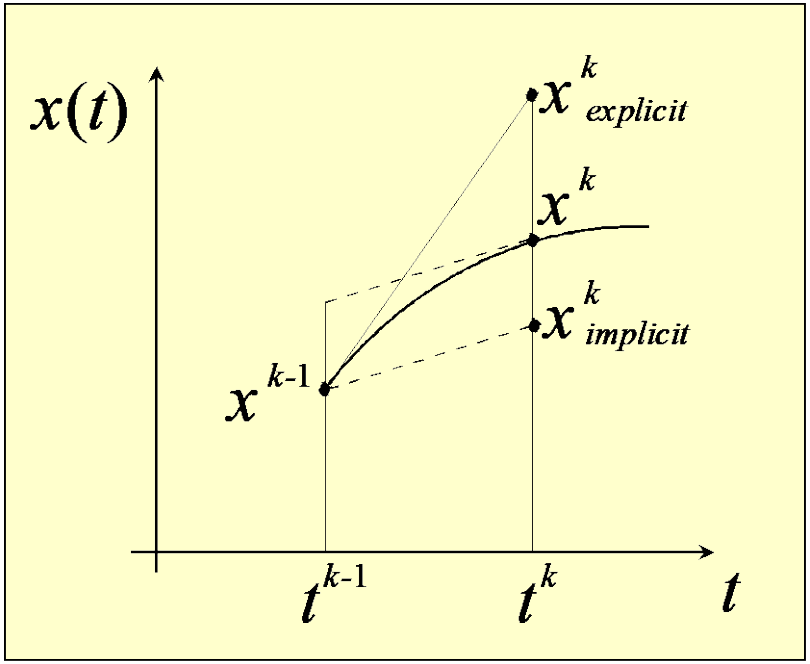
$$\mathbf{x}^{k} = \mathbf{x}^{k-1} + \Delta t \mathbf{f}\left(\mathbf{x}^{k-1}, t^{k-1}\right)$$

Implizites Euler-Verfahren:

$$\frac{\mathbf{x}^{k} - \mathbf{x}^{k-1}}{\Delta t} = \mathbf{f}\left(\mathbf{x}^{k}, t^{k}\right)$$

Ein Newton-Schritt wird benötigt.

Grafische Darstellung



Nachteil: niedrige Genauigkeit

Lösung: • kleine Zeitintervalle
• Polynomapproximation

Polynomapproximation:

Das Gleichungssystem eines dynamischen Systems:

Explizite Form: $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t), \quad \mathbf{x}(t_0) = \mathbf{x}_0, \quad t_0 \le t \le t_f$

Implizite Form: $\mathbf{g}(\dot{\mathbf{x}}, \mathbf{x}, t) = \mathbf{0}, \quad \mathbf{x}(t_0) = \mathbf{x}_0, \quad t_0 \le t \le t_f$

Bei numerischer Lösung wird $\mathbf{x}(t)$ approximiert: $\mathbf{x}(t) \approx \hat{\mathbf{x}}(t)$

Welche Funktion $\hat{\mathbf{x}}(t)$ hat eine hohe Genauigkeit?

Anforderungen:

(1) An bestimmten Zeitpunkten $t_1, \quad t_2, \quad \cdots \quad t_N$ gibt es $\hat{\mathbf{x}}(t_i) = \mathbf{x}(t_i)$

(2) An diesen Zeitpunkten sind mit $\hat{\mathbf{x}}(t_i)$ die Gleichungen erfüllt.

(3) Die Integration im betrachteten Bereich:

$$\int_{t_0}^{t_f} \hat{\mathbf{x}}(t)dt = \int_{t_0}^{t_f} \mathbf{x}(t)dt$$

Numerische Integration:

Nach der Definition: $$\int_a^b x(t)dt = \lim_{\substack{\Delta t_i \to 0 \\ N \to \infty}} \sum_{i=0}^{N} x(t_i)\Delta t_i$$

Der Rechenaufwand ist sehr hoch.

Die Quadratur: $$\int_a^b x(t)dt \approx \sum_{i=0}^{N} w_i x(t_i)$$

Nur die Auswertung der Funktion an vorgegebenen Zeitpunkten wird benötigt.

Die Parameter $w_i, i = 0, \cdots, N$ müssen bestimmt werden.

Bei N+1 Unbekannten braucht man N+1 Gleichungen.

Anforderung:

Bei $x(t) = 1, \quad t, \quad t^2, \cdots, t^N$ muss die Integration exakt sein.

Beispiel: Für den Fall: $N=1, t_0=a, t_1=b, \quad \int_a^b x(t)dt \approx w_0 x(a) + w_1 x(b)$

bei $x(t)=1$ $\quad \int_a^b dt = b-a = w_0 + w_1$

bei $x(t)=t$ $\quad \int_a^b tdt = \frac{1}{2}(b^2-a^2) = w_0 a + w_1 b$

Daher $\quad w_0 = w_1 = \frac{1}{2}(b-a)$

Dann $\quad \int_a^b x(t)dt \approx w_0 x(a) + w_1 x(b) = \frac{1}{2}(b-a)[x(a)+x(b)]$ (Trapez-Formel)

Die exakte Lösung erhält man bei linearen Funktionen.

Für den Fall $N=2, t_0=a, t_1=(b+a)/2, t_2=b$ erhält man die Simpson-Formel:

$$\int_a^b x(t)dt \approx w_0 x(t_0) + w_1 x(t_1) + w_2 x(t_2) = \frac{1}{6}(b-a)[x(a) + 4x((b+a)/2) + x(b)]$$

Die exakte Lösung erhält man bei quadratischen Funktionen.

(Übungsaufgabe!)

Gauß-Quadratur:

Erzielung exakter Lösungen durch die Auswahl der Zeitpunkte.

$$\int_a^b x(t)dt = \sum_{i=0}^{N} w_i x(t_i)$$

Sowohl w_i als auch t_i müssen bestimmt werden ($2(N+1)$ Unbekannte).

Man braucht $2(N+1)$ Gleichungen.

Anforderung:

Bei $x(t) = 1, \quad t, \quad t^2, \cdots, t^{2N+1}$ muss die Integration exakt sein.

Beispiel: Für den Fall $N = 1$

bei $x(t) = 1$ $\qquad b - a = w_0 + w_1$

bei $x(t) = t$ $\qquad (b^2 - a^2)/2 = w_0 t_0 + w_1 t_1$

bei $x(t) = t^2$ $\qquad (b^3 - a^3)/3 = w_0 t_0^2 + w_1 t_1^2$

bei $x(t) = t^3$ $\qquad (b^4 - a^4)/4 = w_0 t_0^3 + w_1 t_1^3$

Die Gleichungen sind schwer zu lösen!

Orthogonale Kollokation:

Erzielung exakter Lösungen durch die Auswahl der Zeitpunkte, so dass für $x(t)$ bis zu $(2N+1)$-ter Ordnung gültig:

$$\int_a^b x(t)dt = \sum_{i=0}^{N} w_i x(t_i)$$

Definition $N+1$ orthogonaler Polynome:

$$p_0(t), \quad p_1(t), \quad \cdots, \quad p_N(t)$$

Achtung: $p_N(t)$ ist ein Polynom mit bis zu $(N+1)$-ter Ordnung.

Definition der Orthogonalität:

$$\int_a^b p_i(t)\,p_j(t)dt = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

Zum Beispiel: Für den Fall $N=1$ gibt es im Bereich $[-1, 1]$:

$$p_0(t) = \frac{1}{\sqrt{2}}, \quad p_1(t) = \sqrt{\frac{3}{2}}\,t, \quad p_2(t) = \sqrt{\frac{5}{2}}\left(\frac{3t^2-1}{2}\right)$$

Fachgebiet Prozessoptimierung

Für den Fall: $x(t)$ ist eine Funktion mit bis zu $(2N+1)$-ter Ordnung, dann

$$x(t) = q_N(t)\,p_N(t) + r_N(t) \qquad \text{(generale Darstellung)}$$

Hierbei sind $q_N(t), r_N(t)$ Polynome mit bis zu N-ter Ordnung.

$$\int_a^b x(t)dt = \int_a^b q_N(t)\,p_N(t)dt + \int_a^b r_N(t)dt$$

Weil $p_0(t), \quad p_1(t), \quad \cdots, \quad p_{N-1}(t)$ mit einander orthogonal sind, dann

$$q_N(t) = \sum_{j=0}^{N-1} c_j\,p_j(t)$$

Daher $$\int_a^b q_N(t)\,p_N(t)dt = \sum_{j=0}^{N-1}\int_a^b c_j\,p_j(t)\,p_N(t)dt = 0$$

Setzt man $x(t) = q_N(t)\,p_N(t)$ in $\int_a^b x(t)dt = \sum_{i=0}^{N} w_i x(t_i)$ ein, ergibt sich

$$\int_a^b q_N(t)\,p_N(t)dt = \sum_{i=0}^{N} w_i q_N(t_i)\,p_N(t_i) = 0$$

Weil $q_N(t)$ beliebig ist, dann $q_N(t_i) \neq 0$

Es muss $p_N(t_i) = 0$

Achtung: $p_N(t)$ ist ein Polynom mit bis zu $(N+1)$-ter Ordnung.

Damit erhält man $N+1$ Zeitpunkte: $t_0, \quad t_1, \quad \cdots \quad t_N$ (Kollokationspunkte)

Sie sind Nullstellen des Polynoms $p_N(t)$.

Nun sind die Parameter $w_i, i = 0, \cdots, N$ zu bestimmen.

Bei $N+1$ Unbekannten braucht man $N+1$ Gleichungen.

Hierzu benutzt man Lagrange-Polynome:

$$l_k(t) = \frac{\prod_{j \neq k} (t - t_j)}{\prod_{j \neq k} (t_k - t_j)}, \quad k = 0, \cdots, N \qquad \Rightarrow \qquad l_k(t_j) = \begin{cases} 1, & j = k \\ 0, & j \neq k \end{cases}$$

Man setzt die Lagrange-Polynome in $\int_a^b x(t)dt = \sum_{i=0}^{N} w_i x(t_i)$ ein,

dann

$$\int_a^b l_k(t)dt = \int_a^b \frac{\prod_{j\neq k}(t-t_j)}{\prod_{j\neq k}(t_k-t_j)}dt = \sum_{i=0}^{N} w_i l_k(t_i) = w_k$$

D.h.

$$w_k = \int_a^b \frac{\prod_{j\neq k}(t-t_j)}{\prod_{j\neq k}(t_k-t_j)}dt, \quad k = 0,\cdots,N$$

Zum Beispiel: Für den Fall N=1 gibt es im Bereich [-1, 1]:

$$p_0(t) = \frac{1}{\sqrt{2}}, \quad p_1(t) = \sqrt{\frac{3}{2}}\,t, \quad p_2(t) = \sqrt{\frac{5}{2}}\left(\frac{3t^2-1}{2}\right)$$

Bestimmung der Kollokationspunkte durch $p_2(t) = 0$: $t_0 = -\frac{1}{\sqrt{3}}, \quad t_1 = \frac{1}{\sqrt{3}}$

$$w_0 = \int_{-1}^{1} \frac{(t-t_1)}{(t_0-t_1)}dt = 1, \quad w_1 = \int_{-1}^{1} \frac{(t-t_0)}{(t_1-t_0)}dt = 1$$

Kollokationsverfahren:

Man berechnet die Integration durch

$$\int_a^b x(t)dt = \sum_{i=0}^{N} w_i x(t_i) \qquad \text{mit} \qquad w_i = \int_a^b l_i(t)dt, \quad i = 0, \cdots, N$$

Also $$\int_a^b x(t)dt = \sum_{i=0}^{N} \left(\int_a^b l_i(t)dt \right) x(t_i) = \int_a^b \left[\sum_{i=0}^{N} l_i(t) x(t_i) \right] dt$$ **Die exakte Lösung!**

D.h.

$$\boxed{x(t) = \sum_{i=0}^{N} l_i(t) x(t_i)}$$

Jetzt wird die Lösung des Differentialgleichungssystems betrachtet:

Welche Funktion $\hat{\mathbf{x}}(t)$ hat eine hohe Genauigkeit?

Anforderungen:

(1) An bestimmten Zeitpunkten $t_1, \quad t_2, \quad \cdots \quad t_N$ gibt es $\hat{\mathbf{x}}(t_i) = \mathbf{x}(t_i)$

(2) An diesen Zeitpunkten sind mit $\hat{\mathbf{x}}(t_i)$ die Gleichungen erfüllt.

(3) Die Integration im betrachteten Bereich: $$\int_{t_0}^{t_f} \hat{\mathbf{x}}(t)dt = \int_{t_0}^{t_f} \mathbf{x}(t)dt$$

Fachgebiet Prozessoptimierung

Kollokationsverfahren:

Man benutzt einfach

$$\hat{\mathbf{x}}(t) = \sum_{i=0}^{N} l_i(t)\mathbf{x}(t_i)$$

Damit werden die Anforderungen erfüllt:

(1) An den Kollokationspunkten $t_0, \quad t_1, \quad \cdots \quad t_N$ gibt es $\hat{\mathbf{x}}(t_i) = \mathbf{x}(t_i)$.

(3) Die Integration: $\int_{t_0}^{t_f} \hat{\mathbf{x}}(t)dt = \int_{t_0}^{t_f} \sum_{i=0}^{N} l_i(t)\mathbf{x}(t_i)dt = \sum_{i=0}^{N} \left(\int_{t_0}^{t_f} l_i(t)dt \right) \mathbf{x}(t_i) = \int_{t_0}^{t_f} \mathbf{x}(t)dt$

(2) An den Kollokationspunkten $t_0, \quad t_1, \quad \cdots \quad t_N$ sind die Ableitungen:

$$\left. \frac{d\hat{\mathbf{x}}(t)}{dt} \right|_{t=t_j} = \sum_{i=0}^{N} \left. \frac{d\, l_i(t)}{dt} \right|_{t=t_j} \mathbf{x}(t_i)$$

wobei

$$\left. \frac{d\, l_i(t)}{dt} \right|_{t=t_j} = \frac{d}{dt} \left. \left(\frac{\prod_{k \neq i} (t - t_k)}{\prod_{k \neq i} (t_i - t_k)} \right) \right|_{t=t_j}$$

Sie sind konstant!

th

prop Fachgebiet Prozessoptimierung

Kollokationsverfahren:

Das Differentialgleichungssystem wird nun ein algebraisches Gleichungssystem:

$$\left.\frac{d\hat{\mathbf{x}}(t)}{dt}\right|_{t=t_j} = \sum_{i=0}^{N}\left[\left.\frac{d\, l_i(t)}{dt}\right|_{t=t_j} \mathbf{x}(t_i)\right] = \mathbf{f}\left(\mathbf{x}(t_j), t_j\right), \quad j = 0, \cdots, N$$

Es wird mit dem Newton-Raphson-Verfahren gelöst und somit erhält man:

$$\mathbf{x}(t_j), \quad j = 0, \cdots, N$$

Sie sind die Werte der Zustandsvariablen an den Kollokationspunkten.

Weil $\hat{\mathbf{x}}(t_i) = \mathbf{x}(t_i)$, sind die Gleichungen mit $\hat{\mathbf{x}}(t_i)$ erfüllt.

Vorteil des Kollokationsverfahrens:

Die Approximation hat eine hohe Genauigkeit!
Damit können große Zeitschritte verwendet werden, d. h. die Recheneffizienz wird erhöht.

Beispiel: Darstellung einer Variable mit zwei Polynormen

$$x(z) = p_0(z)c_0 + p_1(z)c_1, \quad z \in [0,1]$$

Annahme: $p_0(z) = 1$

$$p_1(z) = 1 + a\,z$$

$$p_2(z) = 1 + b\,z + c\,z^2$$

Bedingung der Orthogonalität:

$$\int_0^1 p_0(z)\,p_1(z)dz = \int_0^1 (1 + a\,z)dz = 1 + 0{,}5a = 0$$

$$\int_0^1 p_0(z)\,p_2(z)dz = \int_0^1 (1 + b\,z + c\,z^2)dz = 1 + \frac{b}{2} + \frac{c}{3} = 0$$

$$\int_0^1 p_1(z)\,p_2(z)dz = \int_0^1 (1 + a\,z)(1 + b\,z + c\,z^2)dz = 1 + \frac{a+b}{2} + \frac{ab+c}{3} + \frac{ac}{4} = 0$$

Fachgebiet Prozessoptimierung

Beispiel: Darstellung einer Variable mit zwei Polynormen

Die Lösung: $a = -2, \quad b = -6, \quad c = 6$

d. h. $p_2(z) = 1 - 6z + 6z^2 = 0$

Die Kollokationspunkte: $z_1 = \frac{1}{2}\left(1 - \frac{\sqrt{3}}{3}\right), \quad z_2 = \frac{1}{2}\left(1 + \frac{\sqrt{3}}{3}\right)$

weil $x(z) = c_0 + (1 - 2z)c_1$

Die Anforderung: $x(z_1) = c_0 - \frac{\sqrt{3}}{3}c_1 = x_1, \quad x(z_2) = c_0 + \frac{\sqrt{3}}{3}c_1 = x_2$

Es folgt: $c_0 = \frac{1}{2}(x_1 + x_2), \quad c_1 = \frac{\sqrt{3}}{2}(x_2 - x_1)$

Daher $x(z) = \left[\frac{1}{2} + \frac{\sqrt{3}}{2}(1 - 2z)\right]x_1 + \left[\frac{1}{2} - \frac{\sqrt{3}}{2}(1 - 2z)\right]x_2$

Legendre-Polynome:

Sie sind orthogonale Polynome im Bereich $0 \le z \le 1$

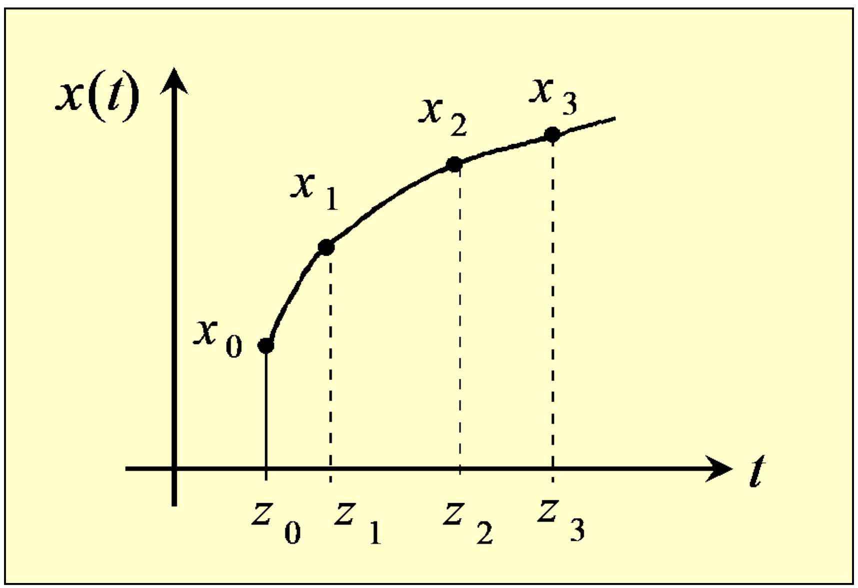
$$\int_0^1 p_i(z)\, p_j(z) dz = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

Liste der inneren Kollokationspunkte (Legendre-Polynorme) bis zur 5. Ordnung

1	0,50000 00000	
2	0,21132 48654	0,78867 51346
3	0,11270 16654	0,50000 00000
	0,88729 83346	
4	0,06943 18442	0,33000 94783
	0,66999 05218	0,93056 81558
5	0,04691 00771	0,23076 53450
	0,50000 00000	0,76923 46551
	0,95308 99230	

Beispiel: Drei-Punkte-Kollokation

$$x(z) = \sum_{i=0}^{3} l_i(z) x_i = \frac{z-z_1}{z_0-z_1}\frac{z-z_2}{z_0-z_2}\frac{z-z_3}{z_0-z_3}x_0 + \frac{z-z_0}{z_1-z_0}\frac{z-z_2}{z_1-z_2}\frac{z-z_3}{z_1-z_3}x_1$$
$$+\frac{z-z_0}{z_2-z_0}\frac{z-z_1}{z_2-z_1}\frac{z-z_3}{z_2-z_3}x_2 + \frac{z-z_0}{z_3-z_0}\frac{z-z_1}{z_3-z_1}\frac{z-z_2}{z_3-z_2}x_3$$



Ableitungen der Polynome an den Kollokationspunkten:

$$\frac{d\mathbf{l}}{d\mathbf{z}} = \begin{bmatrix} \frac{dl_0}{dz_0} & \cdots\cdots & \frac{dl_0}{dz_3} \\ & \cdots\cdots & \\ \frac{dl_3}{dz_0} & \cdots\cdots & \frac{dl_3}{dz_3} \end{bmatrix}$$

(Übungsaufgabe!)

Fachgebiet Prozessoptimierung

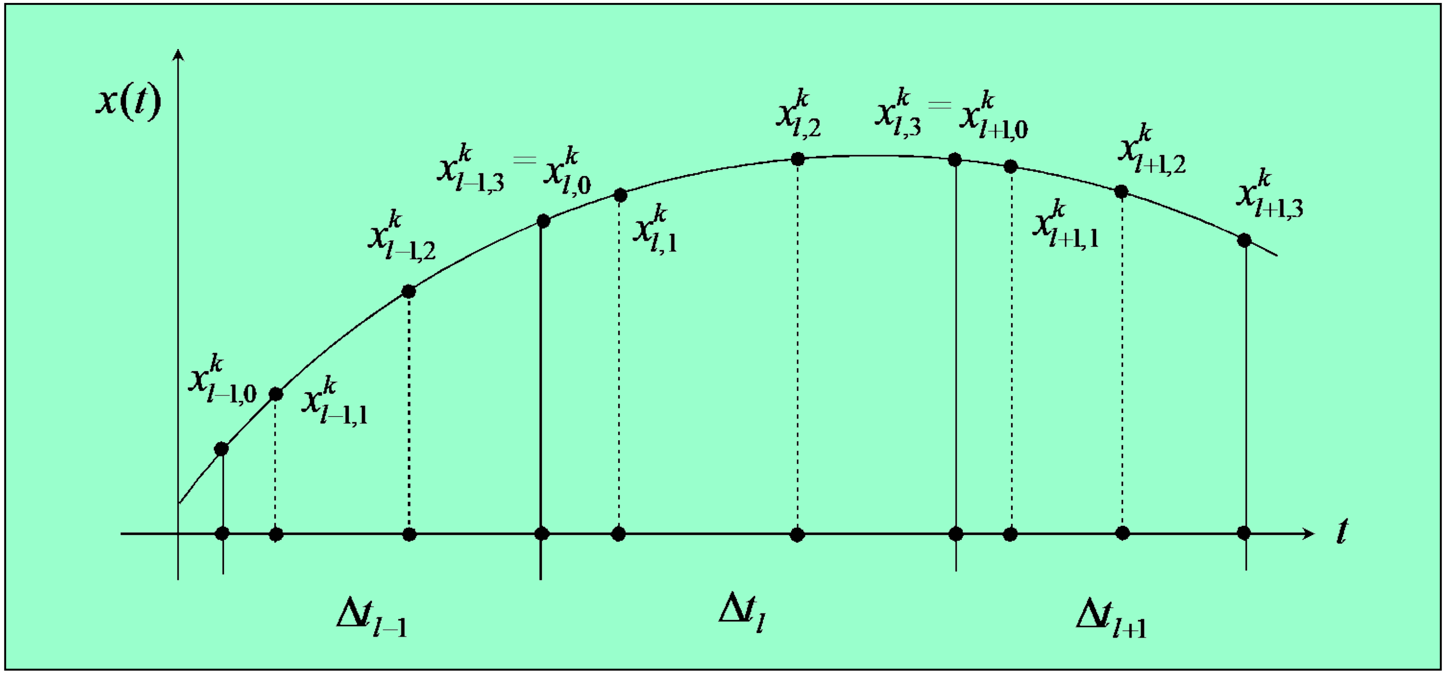
Transformation von $0 \le z \le 1$ **zu** $t_0 \le t \le t_f$ **:**

Weil $z = \dfrac{t - t_0}{t_f - t_0}$ dann $t = t_0 + z(t_f - t_0)$

Stückweise Kollokation:

$$t = t_{0,l} + z(t_{f,l} - t_{0,l}), \quad l = 1, \cdots, NL$$

Für die Kontinuität der Variablen wird der letzte Punkt des Intervalls als Anfangspunkt des nächsten Intervalls benutzt.



Fachgebiet Prozessoptimierung

Diskretisierung der Gleichungen: $\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x})$

An den Kollokationspunkten:

$$\mathbf{x}(t_i) = \sum_{j=0}^{NC} l_j(t_i)\mathbf{x}_j = \mathbf{x}_i$$

$$\dot{\mathbf{x}}(t_i) = \sum_{j=0}^{NC} \frac{dl_j(t_i)}{dt}\mathbf{x}_j$$

Das resultierte System:

$$\sum_{j=0}^{NC} \frac{dl_j(z_i)}{dz}\mathbf{x}_{l,j} = \Delta t_l \mathbf{f}(\mathbf{x}_{l,i}, t_{l,i})$$

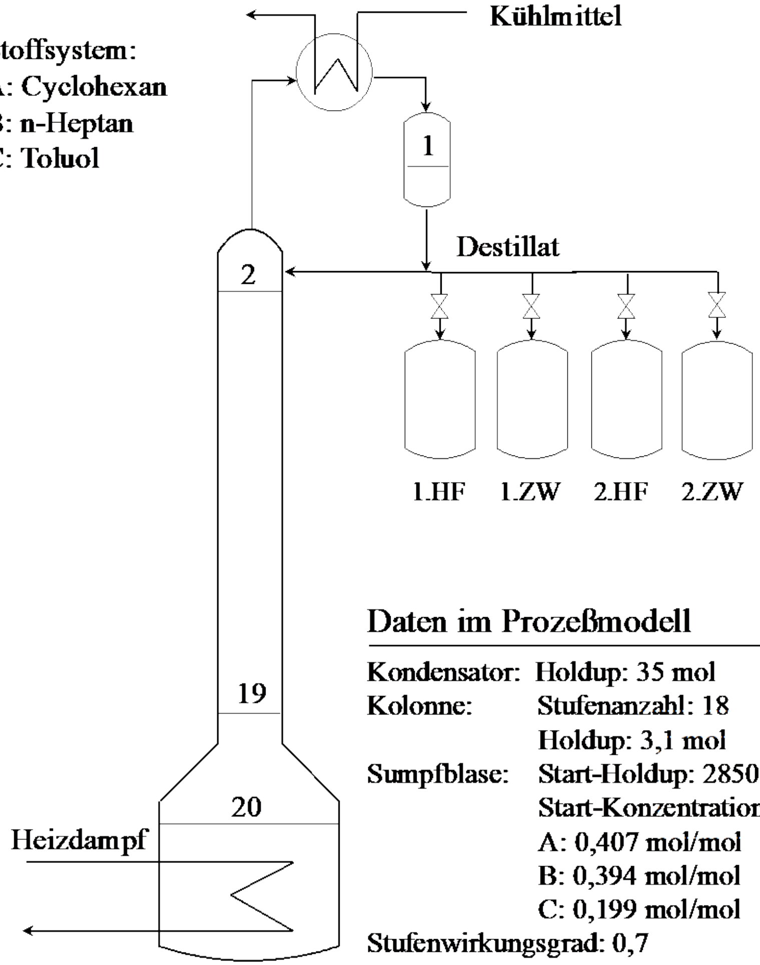
Diskretisierung Integralgleichungen:

$$Int_l = \int_{t_{l-1}}^{t_l} f_l(t)dt, \quad l = 1, \cdots, NL$$

$$\frac{d\,Int_l}{dt} = f_l(t), \quad Int_l(t_{l-1}) = 0$$

Simulation einer Pilotanlage für Batch-Destillation

Stoffsystem:
A: Cyclohexan
B: n-Heptan
C: Toluol

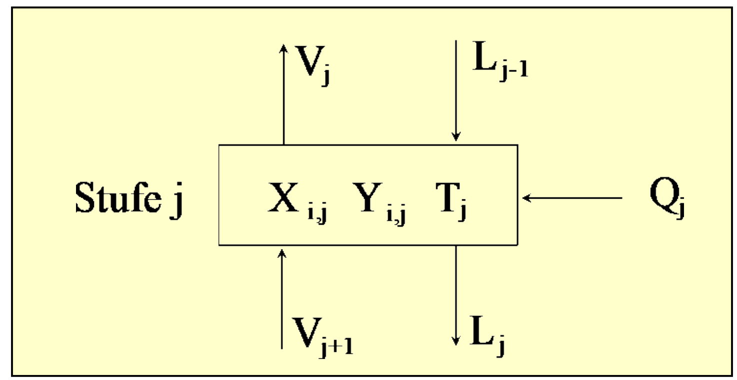


Daten im Prozeßmodell

Kondensator:	Holdup: 35 mol
Kolonne:	Stufenanzahl: 18
	Holdup: 3,1 mol
Sumpfblase:	Start-Holdup: 2850 mol
	Start-Konzentration:
	A: 0,407 mol/mol
	B: 0,394 mol/mol
	C: 0,199 mol/mol
Stufenwirkungsgrad: 0,7	

Modellierung:

- Komponentenbilanz
- Phasengleichgewicht
- Summenbeziehungen
- Energiebilanz



Konzentration der Fraktionen:

$$\overline{x}_i^{\,k} = \frac{\int_{t_{k-1}}^{t_k} D(t)\, x_{i,1}(t)\, dt}{\int_{t_{k-1}}^{t_k} D(t)\, dt}$$

Programmstruktur für die Simulation:

Eingabedaten

- Phasengleichgewichtsparameter
- Thermodynamische Parameter der Stoffe
- Reaktionskinetische Parameter
- Druck und Holdup jeder Stufe
- Anzahl der Stufen Stufenwirkungsgrad
- Profil der Steuergrößen
- Parameter in den Lösungsalgorithmen
- Schätzwert der Anfangsvariablen

Numerische Berechnung

- Diskretisierung mit Implizitem Euler oder Kollokationsverfahren
- Enthalpieberechnung
- Phasengleichgewichtsberechnung
- Berechnung der Reaktionsrate
- Berechnung der Modellgleichungen
- Besetzung der Jacobi-Matrix
- Iterative Lösung der Variablen mit Newton-Raphson-Verfahren

Ergebnisse Zeitliche Profile

<u>Fraktionen:</u>
- Konzentration
- Molmengen

<u>Kondensator:</u>
- Konzentration
- Siedetemperatur
- Wärmeabfuhr

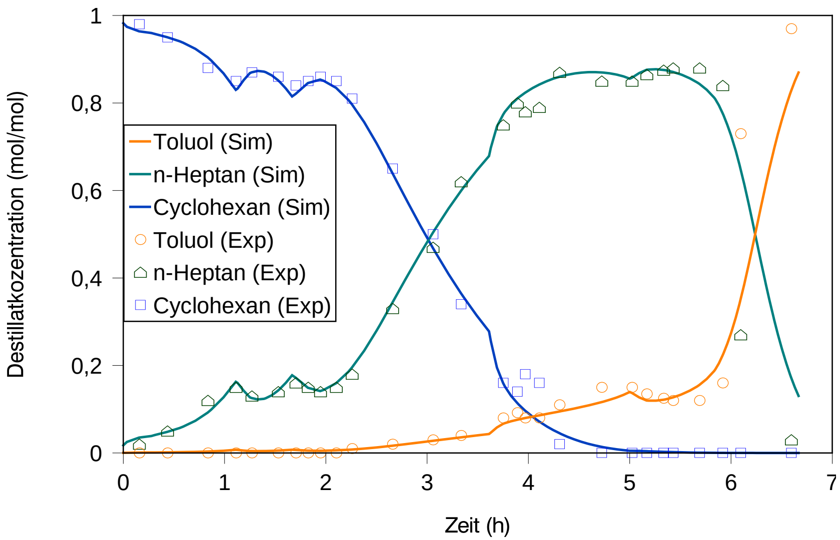
<u>Trennstufe:</u>
- Konzentration
- Temperatur
- Molarströme

<u>Sumpfblase:</u>
- Holdup
- Konzentration
- Temperatur
- Reaktionsrate
- Wärmezufuhr

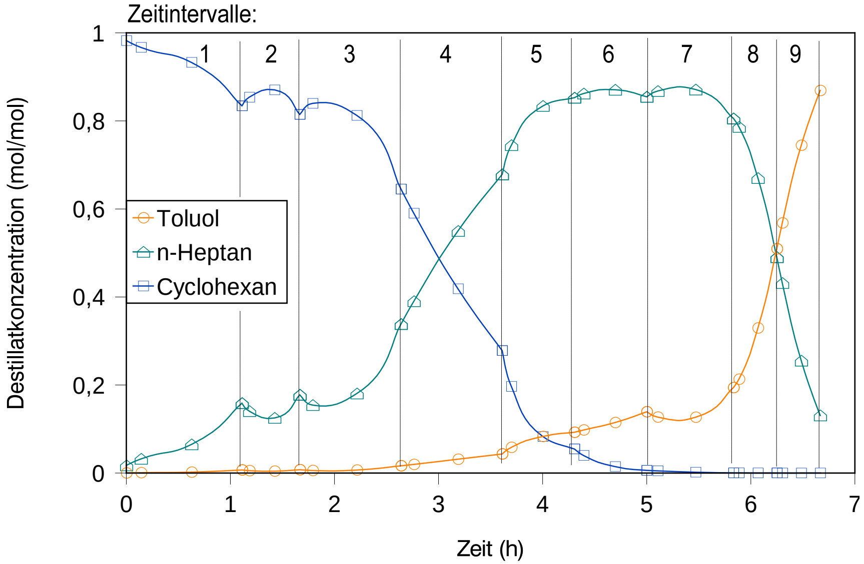
Fachgebiet Prozessoptimierung

Ergebnisse der Simulation:

Implizites Euler-Verfahren



Kollokation



	Länge der Intervalle	Anzahl der Intervalle	Anzahl der berechneten Punkte	CPU-Zeit**
Implizit Euler	100s	234	234	78,09s
Kollokation	600s	39	117	5,63s
Kollokation	2600*s	9	27	3,23s

* Durchschnittswert

Fachgebiet Prozessoptimierung