
Rückkopplungssteuerung am Beispiel der linearen Wärmeleitungsgleichung

Ausarbeitung zum Vortrag

vom 12. Juli 2010
von: Alina Studinger
betreut durch: Prof. Dr. S. Volkwein
Vorlesung: Optimale Steuerung SS 2010
an der Universität Konstanz
Fachbereich Mathematik und Statistik
Arbeitsgruppe Numerik

1 Mehrdimensionale Problemstellung

Wir wollen folgende partielle Differentialgleichung lösen:

$$\left\{ \begin{array}{ll} y_t(t, x) - \operatorname{div}(\kappa(x)\nabla y(t, x)) + a(x)y(t, x) & = f(t, x) + \sum_{k=1}^m \chi_k(x)u_k(t) \\ & \text{für alle } (t, x) \in (0, T) \times \Omega, \\ y(t, x) & = 0 \quad \text{für alle } (t, x) \in (0, T) \times \partial\Omega \\ y(0, x) & = y_0(x) \quad \text{für alle } x \in \Omega \end{array} \right. \quad (1.1)$$

Dabei sei $x \in \Omega \subset \mathbb{R}^n$ (offen) die Raumvariable, zum Beispiel $\Omega = (0, 1)^n$, und $t \in [0, T]$ die Zeitvariable.

Bemerkung:

- Wenn $\kappa \equiv 1$ (konstant), dann ist $-\operatorname{div}(\kappa(x)\nabla y(t, x)) = -\Delta y(t, x)$, denn
 $\operatorname{div}(F) = \sum_{i=1}^n \partial_i F_i$ für $F: \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ und somit auch für $\nabla f = (\partial_1 f, \dots, \partial_n f)^T : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}^n, x \mapsto \nabla f(x)$.
 Es gilt also $\operatorname{div}(\nabla y(t, x)) = \partial_1^2 y(t, x) + \dots + \partial_n^2 y(t, x) = \Delta y(t, x)$

$\kappa(x)$ steht für den Wärmeleitkoeffizienten/Diffusionskoeffizienten am Ort x .

- κ, a und χ sind Funktionen von Ω nach \mathbb{R} , wohingegen die u_i für $i = 1, \dots, m$ Funktionen von $[0, T]$ nach \mathbb{R} sind.

Gegeben sei die **starke Formulierung** (1.1) der zu lösenden partiellen Differentialgleichung. Eine Funktion $y : [0, T] \times \Omega \rightarrow \mathbb{R}$ ist (*starke*) *Lösung*, wenn sie die Gleichung für jedes $x \in \Omega$ und jedes $t \in [0, T]$ erfüllt. Wir ersetzen die Punktauswertungen durch regulärere Funktionale $\langle \cdot, \varphi \rangle_{L^2(\Omega)}$, wobei $\varphi \in V$, der Klasse der Testfunktionen (später werden wir sehen, dass in dem vorliegenden Fall aufgrund der homogenen Dirichlet Bedingung gelten muss: $V = H_0^1(\Omega)$). Dadurch erhalten wir die entsprechende schwache Formulierung, die sogenannte **Variationsgleichung**, welche ich vereinfacht als $LS = RS$ (linke Seite = rechte Seite) schreiben möchte, um die Herleitung der beiden Seiten nun genauer zu erklären. Es gilt:

$$\begin{aligned} LS &= \int_{\Omega} y_t(t, x)\varphi(x) \, dx - \underbrace{\int_{\Omega} \operatorname{div}(\kappa(x)\nabla y(t, x))\varphi(x) \, dx}_{=: Int} + \int_{\Omega} a(x)y(t, x)\varphi(x) \, dx \quad (1.2) \\ RS &= \int_{\Omega} f(t, x)\varphi(x) \, dx + \int_{\Omega} \sum_{k=1}^m \chi_k(x)u_k(t)\varphi(x) \, dx \\ &= \int_{\Omega} f(t, x)\varphi(x) \, dx + \sum_{k=1}^m u_k(t) \int_{\Omega} \chi_k(x)\varphi(x) \, dx \quad (1.3) \end{aligned}$$

Um den Satz von Lax-Milgram, der die Existenz und Eindeutigkeit einer Lösung der Variationsgleichung liefert, anwenden zu können, müssen wir *Int* noch in Richtung gleicher Regularitätsanforderungen an y und φ umformen. Da y und φ aus dem gleichen Raum V kommen

sollen, gilt $\varphi \in V = H_0^1(\Omega)$.

$$\begin{aligned}
Int &= \int_{\Omega} \operatorname{div} \left(\underbrace{\kappa(x) \nabla y(t, x)}_{(\kappa(x) \partial_1 y(t, x), \dots, \kappa(x) \partial_n y(t, x))^T} \right) \varphi(x) \, dx \\
&= \sum_{i=1}^n \int_{\Omega} \partial_i (\kappa(x) \partial_i y(t, x)) \varphi(x) \, dx \\
&= \sum_{i=1}^n \left[- \int_{\Omega} \kappa(x) \partial_i y(t, x) \partial_i \varphi(x) \, dx + \int_{\partial\Omega} \kappa(x) \partial_i y(t, x) \varphi(x) n_i(x) \, dS(x) \right] \\
&= - \int_{\Omega} \kappa(x) \sum_{i=1}^n \partial_i y(t, x) \partial_i \varphi(x) \, dx + \int_{\partial\Omega} \kappa(x) \sum_{i=1}^n \partial_i y(t, x) n_i(x) \underbrace{\varphi(x)}_{=0} \, dS(x) \\
&= - \int_{\Omega} \kappa(x) \nabla y(t, x) \cdot \nabla \varphi(x) \, dx \tag{1.4}
\end{aligned}$$

wobei

- die dritte Gleichheit aufgrund von partieller Integration im mehr-dimensionalen Fall folgt,
- n_i der äußere Normaleneinheitsvektor ist,
- $dS(x)$ die Integration bezüglich dem $(n - 1)$ -dimensionalen Flächenmaß und
- $\sum_{i=1}^n \partial_i y(t, x) n_i(x) = \nabla y(t, x) \cdot n = \frac{\partial y}{\partial n}(t, x)$ die Normalenableitung ist, (welche im Fall von Neumann-Randbedingungen eine Rolle spielen würde, hier jedoch wegfällt) und
- in der letzten Gleichung \cdot für das Skalarprodukt im \mathbb{R}^n steht.

Setzt man nun (1.4) in (1.2) ein und setzt (1.2) = (1.3), erhält man die Variationsgleichung:

$$\begin{aligned}
\int_{\Omega} y_t(t, x) \varphi(x) \, dx + \int_{\Omega} \kappa(x) \nabla y(t, x) \cdot \nabla \varphi(x) \, dx + \int_{\Omega} a(x) y(t, x) \varphi(x) \, dx = \\
\int_{\Omega} f(t, x) \varphi(x) \, dx + \sum_{k=1}^m u_k(t) \int_{\Omega} \chi_k(x) \varphi(x) \, dx \tag{1.5}
\end{aligned}$$

Wir haben nun also eine Formulierung der Form $b(y(t, \cdot), \varphi) = l(\varphi)$ erhalten, wobei die linke Seite den bilinearen Teil und die rechte Seite den linearen Teil enthält.

Wähle nun $t \in [0, T]$ fest, aber beliebig. Wir wollen im Ort diskretisieren.

Die Problemstellung bisher lautet:

$$\begin{aligned}
&\text{Finde } y(t, \cdot) \in V, \text{ so dass (1.5) für alle } \varphi \in V \text{ erfüllt ist,} \\
&\text{wobei } V = \{v \in H^1(\Omega) \mid v(x) = 0 \, \forall x \in \partial\Omega\} = H_0^1(\Omega)
\end{aligned}$$

Wir diskretisieren mit der **Finite Elemente Methode**.

Für den Ansatzraum/Raum der Formfunktionen V_h soll gelten: $V_h \subset V$ mit $V_h = [\varphi_1, \dots, \varphi_{N_{FE}}]$, wobei φ_i stückweise Polynome sind.

Im ein-dimensionalen Fall wird dies dann genauer ausgeführt werden, siehe dazu Gleichung

(2.3).

Im Folgenden sei $N := N_{FE}$ die Dimension des Finite Element Ansatzraumes.
Gesucht ist $y^h(t, \cdot) \in V_h$ so, dass

$$b(y^h(t, \cdot), \varphi_i) = l(\varphi_i) \text{ für alle } i \in \{1, \dots, N\} \quad (1.6)$$

Es reicht (1.5) für die Basisfunktionen $\varphi = \varphi_i$ zu überprüfen/fordern, da V_h als lineare Hülle der φ_i ein Vektorraum ist.

Da $y^h(t, \cdot) \in V_h$ ist, gilt:

$$y^h(t, \cdot) = \sum_{j=1}^N c_j(t) \varphi_j \quad (1.7)$$

Daraus folgt dann entsprechend:

$$y_t^h(t, x) = \sum_{j=1}^N \frac{\partial}{\partial t} c_j(t) \varphi_j(x) \quad \text{und} \quad \nabla y^h(t, x) = \sum_{j=1}^N c_j(t) \nabla \varphi_j(x)$$

Setzen wir dies in (1.6) ein, erhalten wir die folgende diskrete, schwache Formulierung.

Für $i = 1, \dots, N$:

$$\begin{aligned} \sum_{j=1}^N (c_j'(t) \underbrace{\int_{\Omega} \varphi_j(x) \varphi_i(x) \, dx}_{=: M_{ij}}) + \sum_{j=1}^N c_j(t) \underbrace{\left(\int_{\Omega} \kappa(x) \nabla \varphi_j(x) \cdot \nabla \varphi_i(x) \, dx + \int_{\Omega} a(x) \varphi_j(x) \varphi_i(x) \, dx \right)}_{=: T_{ij}} = \\ \underbrace{\int_{\Omega} f(t, x) \varphi_i(x) \, dx}_{=: f_i(t)} + \sum_{k=1}^m u_k(t) \underbrace{\int_{\Omega} \chi_k(x) \varphi_i(x) \, dx}_{=: \bar{B}_{ik}} \end{aligned} \quad (1.8)$$

Setzen wir

$$c(t) := \begin{pmatrix} c_1(t) \\ \vdots \\ c_N(t) \end{pmatrix} \in \mathbb{R}^N \quad (1.9)$$

erhalten wir ein lineares Gleichungssystem für eben diese Koeffizienten.

$$\begin{cases} M c'(t) &= -T c(t) + f(t) + \bar{B} u(t) \\ M c(0) &= \bar{c} \end{cases} \quad (1.10)$$

mit

Massematrix $M = (M_{ij})_{1 \leq i, j \leq N} \in \mathbb{R}^{N \times N}$ mit M_{ij} wie in (1.8) definiert,
Steifigkeitsmatrix $T = (T_{ij})_{1 \leq i, j \leq N} \in \mathbb{R}^{N \times N}$ mit T_{ij} wie in (1.8) definiert,
 $f(t) = (f_i(t))_{1 \leq i \leq N} \in \mathbb{R}^N$,
 $\bar{B} = (\bar{B}_{ik})_{1 \leq i \leq N, 1 \leq k \leq m} \in \mathbb{R}^{N \times m}$,
 $u : [0, T] \rightarrow \mathbb{R}^m$ mit $t \mapsto (u_1(t), \dots, u_m(t))^T$ und
 $\bar{c} = (\int_{\Omega} y_0(x) \varphi_1(x) \, dx, \dots, \int_{\Omega} y_0(x) \varphi_N(x) \, dx)^T$

Der Anfangswert \bar{c} dieser gewöhnlichen Differentialgleichung ergibt sich folgenderweise:

Für $t = 0$ wird $y(0, \cdot)$ approximiert durch $y^h(0, \cdot) = \sum_{j=1}^N c_j(0) \varphi_j$.

Wir wissen nach DGL (1.1), dass $y(0, x) = y_0(x)$ für alle $x \in \Omega$. Also muss in schwacher Formulierung gelten, dass

$$\int_{\Omega} y^h(0, x) \varphi_i(x) \, dx = \int_{\Omega} y_0(x) \varphi_i(x) \, dx \text{ für } i = 1, \dots, N$$

Dies ist äquivalent zu

$$\sum_{j=1}^N c_j(0) \underbrace{\int_{\Omega} \varphi_j(x) \varphi_i(x) \, dx}_{=M_{ij}} = \underbrace{\int_{\Omega} y_0(x) \varphi_i(x) \, dx}_{=:c_i^0} \text{ für } i = 1, \dots, N$$

Dies ergibt in Matrix-/Vektornotation

$$Mc(0) = \bar{c} \text{ mit } \bar{c} = \begin{pmatrix} \int_{\Omega} y_0(x) \varphi_1(x) \, dx \\ \vdots \\ \int_{\Omega} y_0(x) \varphi_N(x) \, dx \end{pmatrix}$$

Wir haben also ausgehend von der vorgelegten partiellen Differentialgleichung (1.1) über die Diskretisierung der schwachen Formulierung eine gewöhnliche Differentialgleichung für die Koeffizienten der approximativen/numerischen Lösung erhalten.

Um (1.1) numerisch zu lösen werden wir also das hergeleitete äquivalente System (1.10) von gewöhnlichen Differentialgleichungen lösen. Mit der Assemblierung dieses Systems und dessen Lösung beschäftigen wir uns im Folgenden für den Fall $n = 1$.

2 Numerische Behandlung des eindimensionalen Problems

Wir betrachten nun einen Spezialfall der vorgestellten partiellen Differentialgleichung (1.1), denn wir untersuchen das folgende **eindimensionale** Problem:

$$\begin{cases} y_t(t, x) - \frac{d}{dx}(\kappa(x)y_x(t, x)) + a(x)y(t, x) = f(t, x) + \sum_{k=1}^m \chi_k(x)u_k(t) & \text{für alle } (t, x) \in (0, T) \times \Omega, \\ y(t, 0) = y(t, 1) = 0 & \text{für alle } t \in (0, T) \\ y(0, x) = y_0(x) & \text{für alle } x \in \Omega = (0, 1) \end{cases} \quad (1D)$$

2.1 Schwache Formulierung und FE-Diskretisierung

Wir suchen die Lösung $y : [0, T] \times \Omega \rightarrow \mathbb{R}$ von Gleichung (1D). Diese wird in einem ersten Schritt angenähert durch die Finite Element Lösung $y^h(t, \cdot) = \sum_{j=1}^N c_j(t)\varphi_j$. Die Koeffizienten $c_j : [0, T] \rightarrow \mathbb{R}$ werden durch Lösen einer gewöhnlichen Differentialgleichung bestimmt. Diese ergibt sich analog wie im mehr-dimensionalen Fall in Abschnitt 1, siehe (1.8).

Für jedes $i = 1, \dots, N$ muss gelten

$$\begin{aligned} \sum_{j=1}^N (c_j'(t) \underbrace{\int_{\Omega} \varphi_j(x)\varphi_i(x) dx}_{=: M_{ij}}) + \sum_{j=1}^N c_j(t) \left(\underbrace{\int_{\Omega} \kappa(x)\varphi_j'(x)\varphi_i'(x) dx}_{=: T_{ij}} + \int_{\Omega} a(x)\varphi_j(x)\varphi_i(x) dx \right) = \\ \underbrace{\int_{\Omega} f(t, x)\varphi_i(x) dx}_{=: f_i(t)} + \sum_{k=1}^m u_k(t) \underbrace{\int_{\Omega} \chi_k(x)\varphi_i(x) dx}_{=: \bar{B}_{ik}} \end{aligned} \quad (2.1)$$

Daher suchen wir $c : [0, T] \rightarrow \mathbb{R}^N$, das das folgendes System von gewöhnlichen Differentialgleichungen löst.

$$M c'(t) + T c(t) = f(t) + \bar{B} u(t) \quad (2.2a)$$

$$M c(0) = \bar{c} \quad (2.2b)$$

wobei

$$c(t) = (c_1(t), \dots, c_N(t))^T$$

$$M = (M_{ij})_{1 \leq i, j \leq N} \text{ („Massematrix“),}$$

$$T = (T_{ij})_{1 \leq i, j \leq N} \text{ je } \in \mathbb{R}^{N \times N} \text{ („Steifigkeitsmatrix“),}$$

$$f(t) = (f_i(t))_{1 \leq i \leq N} \in \mathbb{R}^N,$$

$$\bar{B} = (\bar{B}_{ik})_{1 \leq i \leq N, 1 \leq k \leq m} \in \mathbb{R}^{N \times m}$$

$$u : [0, T] \rightarrow \mathbb{R}^m \text{ mit } t \mapsto \begin{pmatrix} u_1(t) \\ \vdots \\ u_m(t) \end{pmatrix}$$

$$\bar{c} = \left(\int_{\Omega} y_0(x)\varphi_1(x) dx, \dots, \int_{\Omega} y_0(x)\varphi_N(x) dx \right)^T$$

Konkret verwenden wir den Ansatzraum $V_h = [\varphi_1, \dots, \varphi_N]$,

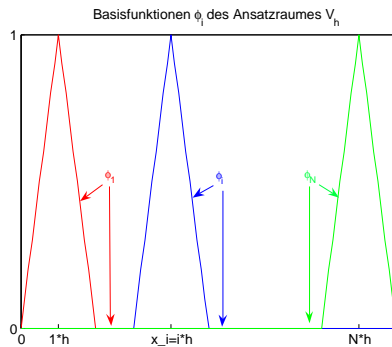


Abbildung 1: sog. „Hütchen-Funktionen“ als Basisformfunktionen

wobei für $i = 1, \dots, N$ gilt:

$$\varphi_i(x) = \begin{cases} -(i-1) + \frac{1}{h}x, & \text{für } x \in [(i-1)h, ih], \\ (i+1) - \frac{1}{h}x, & \text{für } x \in (ih, (i+1)h], \\ 0, & \text{sonst.} \end{cases} \quad (2.3)$$

h erfüllt die Beziehung $(N+1)h = 1$.

Da $\varphi_j(x_i) = \delta_{ij}$ mit $x_i = ih$ für $i, j = 1, \dots, N$ gilt, ist diese Basis eine sogenannte „nodale Basis“.

Da wir nun wissen, wie die Basisfunktionen des Ansatzraumes aussehen, können wir die Matrizen M, T, \bar{B} und die Vektoren $f(t)$ und \bar{c} aufstellen.

2.2 Assemblierung der Matrizen und Vektoren der GDL (2.2a)-(2.2b)

Als erstes wollen wir die **Massematrix** M aufstellen. Wir wissen bisher, dass $M \in \mathbb{R}^{N \times N}$ mit

$$M_{ij} = \int_0^1 \varphi_j(x) \varphi_i(x) \, dx$$

Aufgrund der obigen Wahl der Basisfunktionen ist dieses Integral nur für benachbarte φ_i ungleich 0, genauer:

$$\begin{aligned} \text{Wenn } i = j & & \text{dann } M_{ii} &= \int_{\Omega} \varphi_i^2(x) \, dx = \frac{2}{3}h \\ \text{Wenn } |i - j| = 1 & & \text{dann } M_{ij} &= M_{i,i+1} = M_{j-1,j} = \frac{1}{6}h \end{aligned}$$

denn für $i = 1, \dots, N$ gilt

$$\begin{aligned}
M_{ii} &= \int_0^1 \varphi_i^2(x) \, dx \\
&= \int_{(i-1)h}^{ih} \left(-i + 1 + \frac{1}{h}x\right)^2 dx + \int_{ih}^{(i+1)h} \left(i + 1 - \frac{1}{h}x\right)^2 dx \\
&= \int_{(i-1)h}^{ih} \left(i - 1\right)^2 - \frac{2}{h}(i - 1)x + \frac{x^2}{h^2} dx + \int_{ih}^{(i+1)h} \left(i + 1\right)^2 - \frac{2}{h}(i + 1)x + \frac{x^2}{h^2} dx \\
&= \left[(i - 1)^2 x - \frac{(i - 1)}{h} x^2 + \frac{1}{3h^2} x^3 \right]_{(i-1)h}^{ih} + \left[(i + 1)^2 x - \frac{(i + 1)}{h} x^2 + \frac{1}{3h^2} x^3 \right]_{ih}^{(i+1)h} \\
&= (i - 1)^2 ih - (i - 1)i^2 h + \frac{1}{3}i^3 h - (i - 1)^3 h + (i - 1)^3 h - \frac{1}{3}(i - 1)^3 h \\
&\quad + (i + 1)^3 h - (i + 1)^3 h + \frac{1}{3}(i + 1)^3 h - (i + 1)^2 ih + (i + 1)i^2 h - \frac{1}{3}i^3 h \\
&= i^3 h - 2i^2 h + ih - i^3 h + i^2 h - \frac{1}{3}i^3 h + i^2 h - ih + \frac{1}{3}h \\
&\quad + \frac{1}{3}i^3 h + i^2 h + ih + \frac{1}{3}h - i^3 h - 2i^2 h - ih + i^3 h + i^2 h \\
&= \underbrace{i^3 \left(h - h - \frac{1}{3}h + \frac{1}{3}h - h + h\right)}_{=0} + \underbrace{i^2 (-2h + h + h + h - 2h + h)}_{=0} + \underbrace{i(h - h + h - h)}_{=0} + h\left(\frac{1}{3} + \frac{1}{3}\right) \\
&= \frac{2}{3}h
\end{aligned}$$

und

$$\begin{aligned}
M_{i,i+1} &= \int_0^1 \varphi_i(x)\varphi_{i+1}(x) \, dx \\
&= \int_{ih}^{(i+1)h} \left(i + 1 - \frac{1}{h}x\right)\left(-i + \frac{1}{h}x\right) dx \\
&= \int_{ih}^{(i+1)h} \left(-i^2 - i + \frac{1}{h}xi + \frac{1}{h}xi + \frac{1}{h}x - \frac{1}{h^2}x^2\right) dx \\
&= \int_{ih}^{(i+1)h} \left(-i^2 - i + 2\frac{1}{h}xi + \frac{1}{h}x - \frac{1}{h^2}x^2\right) dx \\
&= \left[-i^2 x - ix + \frac{i}{h}x^2 + \frac{1}{2h}x^2 - \frac{1}{3h^2}x^3 \right]_{ih}^{(i+1)h}
\end{aligned}$$

$$\begin{aligned}
&= -i^2(i+1)h - i(i+1)h + \frac{1}{2h}(i+1)^2h^2 - \frac{1}{3h^2}(i+1)^3h^3 \\
&\quad + i^3h + i^2h - \frac{1}{2}i^2h + \frac{1}{3}i^3h + i(i+1)^2h - i^3h \\
&= -i^3h - i^2h - i^2h - ih + \frac{1}{2}i^2h + ih + \frac{1}{2}h - \frac{1}{3}i^3h - i^2h - ih + \frac{1}{3}h \\
&\quad + i^3h + i^2h - \frac{1}{2}i^2h + \frac{1}{3}i^3h + i^3h + 2i^2h + ih - i^3h \\
&= i^3 \underbrace{\left(-h - \frac{1}{3}h + h + \frac{1}{3}h + h - h\right)}_{=0} + i^2 \underbrace{\left(-h - h + \frac{1}{2}h - h + h - \frac{1}{2}h + 2h\right)}_{=0} \\
&\quad + i \underbrace{\left(-h + h - h + h\right)}_{=0} + h\left(\frac{1}{2} - \frac{1}{3}\right) \\
&= \frac{1}{6}h
\end{aligned}$$

Aufgrund von Symmetrie gilt $M_{i,i+1} = M_{i-1,i}$ und damit:

$$M = h \begin{pmatrix} \frac{2}{3} & \frac{1}{6} & & & 0 \\ \frac{1}{6} & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \frac{1}{6} \\ 0 & & & \frac{1}{6} & \frac{2}{3} \end{pmatrix} \in \mathbb{R}^{N \times N} \quad (2.4)$$

Aufgrund der Form der Massmatrix M empfiehlt sich die Implementierung im sparse-Format. Als Beispiel sehen wir hier den Programcode von [massmatrix.m](#).

```

function M = massmatrix(h)
%% code for assembling the mass matrix M in R^NxN
% where N = 1/h - 1
% of the discrete weak formulation (P1 - Finite Elements)
% of the one dimensional case of y_t(t,x) - div(Kappa(x)*grad(y(t,x))) +
% a(x)*y(t,x) = f(t,x) + sum_k (u_k(t)*chi_k(x)) forall (t,x) in (0,T)x(0,1)

N = round(1/h - 1);
main = (2/3).*ones(N,1);
sides = (1/6).*ones(N,1);
M = h.*spdiags([sides,main,sides],[-1,0,1],N,N);

```

Als nächstes wollen wir die **Steifigkeitsmatrix** T aufstellen. Wir wissen bisher, dass $T \in \mathbb{R}^{N \times N}$ mit

$$T_{ij} = \int_0^1 \kappa(x) \varphi_j'(x) \varphi_i'(x) dx + \int_0^1 a(x) \varphi_j(x) \varphi_i(x) dx \quad (2.5)$$

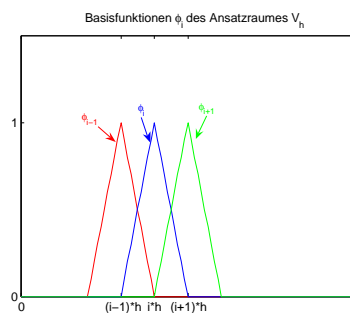
Aufgrund der obigen Wahl der Basisfunktionen sind diese Integrale nur für benachbarte φ_i ungleich 0. Wir betrachten im Folgenden immer $a(x) \equiv a$ mit $a \in \mathbb{R}$ konstant, später setzen wir sogar a als Nullfunktion voraus.

In einem ersten Schritt schauen wir uns an, welche Gestalt T hat, wenn κ ebenfalls konstant ist, dh. $\forall x \in \Omega : \kappa(x) = \kappa$ mit $\kappa > 0$. In diesem Fall können wir die Einträge von T „von Hand“

berechnen. Es gilt

$$\begin{aligned}
 T_{ij} &= \kappa \int_0^1 \varphi'_j(x) \varphi'_i(x) \, dx + a \int_0^1 \varphi_j(x) \varphi_i(x) \, dx \\
 &= \underbrace{\kappa \int_0^1 \varphi'_j(x) \varphi'_i(x) \, dx}_{=: I_{ij}} + a M_{ij}
 \end{aligned}$$

Was ist I_{ij} ?



Für $i = j \in \{1, \dots, N\}$ gilt:

$$T_{ii} = \int_0^1 (\varphi'_i(x))^2 \, dx = \int_{(i-1)h}^{(i+1)h} \frac{1}{h^2} \, dx = \frac{1}{h^2} [x]_{(i-1)h}^{(i+1)h} = \frac{1}{h^2} 2h = \frac{2}{h}$$

Für $|i - j| = 1$ hat eine der beiden Funktionen die Steigung $-\frac{1}{h}$ und die andere hat Steigung $\frac{1}{h}$, diese treffen sich auf einem Intervall der Breite h , damit folgt:

$$T_{ii-1} = T_{ii+1} = \int_0^1 \varphi'_{i+1}(x) \varphi'_i(x) \, dx = -\frac{1}{h^2} h = -\frac{1}{h}$$

Damit hat T im Fall von konstanten Funktionen κ und a die Form:

$$T = \frac{\kappa}{h} \begin{pmatrix} 2 & -1 & & 0 \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & \ddots \\ & & \ddots & \ddots & -1 \\ 0 & & & -1 & 2 \end{pmatrix} + aM \tag{2.6}$$

Im zweiten Schritt überlegen wir uns wie man T effizient berechnen kann, wenn κ nicht konstant ist. Wir betrachten nur das erste Integral in (2.5), denn das zweite verändert sich im Vergleich

zum ersten Schritt nicht. Dieses Integral nennen wir nun S_{ij} . Es gilt

$$S_{ij} = \int_0^1 \kappa(x) \varphi'_j(x) \varphi'_i(x) dx = \sum_{k=1}^{N+1} \int_{(k-1)h}^{kh} \kappa(x) \varphi'_j(x) \varphi'_i(x) dx$$

Für jedes $k \in \{1, \dots, N+1\}$ ist $\varphi'_j(x) \varphi'_i(x) = 0$ für $x \in [(k-1)h, kh], |i-j| > 1$. Für $i = j+1$ gilt:

$$\int_{(k-1)h}^{kh} \kappa(x) \varphi'_j(x) \varphi'_{j+1}(x) dx = \begin{cases} \int_{(k-1)h}^{kh} \kappa(x) (-\frac{1}{h^2}) dx, & \text{für } k = j+1, \\ 0, & \text{sonst.} \end{cases} \quad (2.7)$$

Ebenso gilt für $j = i+1$:

$$\int_{(k-1)h}^{kh} \kappa(x) \varphi'_{i+1}(x) \varphi'_i(x) dx = \begin{cases} \int_{(k-1)h}^{kh} \kappa(x) (-\frac{1}{h^2}) dx, & \text{für } k = i+1, \\ 0, & \text{sonst.} \end{cases} \quad (2.8)$$

wohingegen für $i = j$ gilt:

$$\int_{(k-1)h}^{kh} \kappa(x) (\varphi'_i(x))^2 dx = \begin{cases} \int_{(k-1)h}^{kh} \kappa(x) \frac{1}{h^2} dx, & \text{für } k \in \{i, i+1\}, \\ 0, & \text{sonst.} \end{cases} \quad (2.9)$$

Gleichung (2.7) gilt, da $\varphi'_j(x) \varphi'_{j+1}(x) = 0$ für $x \notin (jh, (j+1)h)$, analog folgt (2.8), wohingegen $(\varphi'_i(x))^2 = 0$ für $x \notin ((i-1)h, (i+1)h)$. Damit gilt also

$$S_{ii} = \frac{1}{h^2} \int_{(i-1)h}^{(i+1)h} \kappa(x) dx$$

$$S_{i,i+1} = S_{i+1,i} = -\frac{1}{h^2} \int_{ih}^{(i+1)h} \kappa(x) dx$$

Diese Integrale über $\kappa(x)$ können zum Beispiel mit der zusammengesetzten Trapezregel oder einer anderen geeigneten Quadraturformel berechnet werden. Da auch T eine Tridiagonalmatrix ist, kann man die Matrix wieder als Sparse-Matrix programmieren, das heißt effizient und speichersparend.

Als nächstes assemblieren wir den Vektor $f(t)$ mit Komponenten $f_i(t)$ wie in dem Gleichungssystem (2.1). Für $i = 1, \dots, N$ gilt

$$f_i(t) = \int_0^1 f(t, x) \varphi_i(x) dx$$

$$\approx h \left(\frac{1}{2} f(t, 0) \underbrace{\varphi_i(0)}_{=0 \forall i} + \frac{1}{2} f(t, 1) \underbrace{\varphi_i(1)}_{=0 \forall i} + \sum_{j=1}^N f(t, jh) \underbrace{\varphi_i(jh)}_{=\delta_{ij}} \right)$$

$$= hf(t, ih)$$

Damit ist der von der Zeit abhängige Vektor $f(t)$ näherungsweise gegeben durch

$$f(t) = h \begin{pmatrix} f(t, h) \\ \vdots \\ f(t, Nh) \end{pmatrix} \quad (2.10)$$

Das Integral zur Berechnung von \bar{B}_{ik} ersetzen wir ebenfalls durch eine Näherung durch das Anwenden der *zusammengesetzten Trapezregel*. Somit gilt

$$\begin{aligned} \bar{B}_{ik} &= \int_0^1 \chi_k(x) \varphi_i(x) \, dx \\ &\approx h \left(\frac{1}{2} \chi_k(0) \underbrace{\varphi_i(0)}_{=0 \, \forall i} + \frac{1}{2} \chi_k(0) \underbrace{\varphi_i(0)}_{=0 \, \forall i} + \sum_{j=1}^N \chi_k(jh) \underbrace{\varphi_i(jh)}_{\delta_{ij}} \right) \\ &= h \chi_k(ih) \end{aligned}$$

Daher ist

$$\bar{B} = h \begin{pmatrix} \chi_1(h) & \cdots & \chi_m(h) \\ \vdots & \ddots & \vdots \\ \chi_1(Nh) & \cdots & \chi_m(Nh) \end{pmatrix} \quad (2.11)$$

Nach dem gleichen Prinzip lassen sich die Integrale in den Komponenten des **Anfangswertes** annähern durch

$$\bar{c}_i = \int_0^1 y_0(x) \varphi_i(x) \, dx \approx h y_0(ih)$$

In Vektorenschreibweise und mit $=$ anstatt \approx folgt also

$$\bar{c} = h \begin{pmatrix} y_0(h) \\ \vdots \\ y_0(Nh) \end{pmatrix} \quad (2.12)$$

Da wir nun alle Bestandteile des Differentialgleichungssystems (2.2) kennen, können wir diese in Matlab implementieren und dann das so gegebene System numerisch lösen. Hierbei bietet sich das implizite Euler-Verfahren an. Es ist für jede zeitliche Diskretisierungsschrittweite Δt stabil. Beim Programmieren habe ich zum Beispiel $\Delta t = h$ gesetzt, dh. zeitliche Diskretisierungsschrittweite = örtliche Diskretisierungsschrittweite.

Wendet man den impliziten Euler auf das System (2.2) an und sei $c^k \in \mathbb{R}^N$ die Approximation von $c(t_k)$ für $k = 1, \dots, n$ mit $n = \frac{T-t_0}{\Delta t}$, dann erhält man ein lineares Gleichungssystem für die Vektoren c^k mit $k = 1, \dots, n$

$$\left(\frac{1}{\Delta t} M + T \right) c^{k+1} = \frac{1}{\Delta t} M c^k + f(t_{k+1}) + B u(t_{k+1}) \quad (2.13a)$$

$$M c^0 = \bar{c} \quad (2.13b)$$

wobei $t_k = t_0 + k\Delta t$ und in unserem Fall $t_0 = 0$.

Dieses Gleichungssystem kann man zum Beispiel unter Verwendung der Cholesky-Zerlegung und der `\`-Matlab function lösen.

Wenn wir die Lösungsvektoren in eine Matrix Y schreiben, also die k -te Spalte von Y entspricht c^k , dann gilt

$$Y_{ik} \approx y(t_k, x_i) \tag{2.14}$$

wobei $y(t_k, x_j)$ die Lösung von (1D) an der Stelle (t, x) ist.

Die Annäherung (2.14) ergibt sich aus den Definitionen von c^k und $c(t)$. $c(t)$ ist der Vektor mit Komponenten $c_1(t), \dots, c_N(t)$, wobei die numerische Lösung y^h die Gestalt $y^h(t, x) = \sum_{j=1}^N c_j(t) \varphi_j(x)$ hat. Durch numerisches Lösen der Anfangswertaufgabe (2.2) erhalten wir $Y(:, k) = c^k \approx c(t_k) \in \mathbb{R}^N$. Damit gilt $y(t_k, x_i) \approx y^h(t_k, x_i) \approx \sum_{j=1}^N Y(j, k) \underbrace{\varphi_j(x_i)}_{=\delta_{ij}} = Y(i, k)$. Dies

kann man sich beim Plotten der numerischen Lösung zunutze machen und die Befehle

```
mt = t_0:dt:t_end;
mx = 0:h:1;
[MT,MX] = meshgrid(mt,mx);
mesh(MT,MX,Y)
```

liefern die später folgenden Abbildungen der Lösung.¹

¹Um genau die Abbildungen von später zu erhalten, muss man Y noch um eine erste und letzte Zeile von Nullen erweitern, dh. um die Werte der Lösung an den Punkten $x = 0$ und $x = 1$, da nur auf den inneren Gitterpunkten $x \in \{h, \dots, Nh\}$ gelöst wurde. Die Tatsache, dass auf den Randpunkten $x = x_0 = 0$ und $x = x_{N+1} = 1$ die Lösung des Randwertproblems (1D) $y(t, x) = 0$ ist, ist in das Aufstellen der Gleichungssysteme, also auch in die numerische Lösung, eingegangen. Die oben genannte Erweiterung der Lösungsmatrix ist also mit dem sonstigen Verlauf der numerischen Lösung konform.

3 Das ungesteuerte bzw. manuell kontrollierte Problem

3.1 Lösung der PDGL ohne Steuerung

In diesem Abschnitt behandeln wir die vorgelegte eindimensionale partielle Differentialgleichung (1D) mit rechter Seite gleich Null. Insbesondere heißt dies, dass $f \equiv 0$ und $\chi_k = 0$ für $k = 1, \dots, m$ bzw. $u(t) = 0$ für alle $t \in [0, 1]$, also keine Steuerung, gesetzt wird. Des Weiteren seien die Funktionen a und κ in Gleichung (1D) im ersten Schritt konstant, wobei wir $a = 0$ setzen und für κ verschiedene Werte testen um ein Gefühl für die Bedeutung von κ für die Lösung zu erhalten.

Im Folgenden setzen wir als Anfangsbedingung $y_0(x) = \sin(5\pi x)$ voraus. Dies passt gut zu den gegebenen Dirichlet-Randbedingungen, $y(t, x) = 0$ für alle $x \in \partial\Omega = \{0, 1\}$ und alle $t \in (0, T)$. Die Schwingungen des sin sollte man dann auch bei der Lösungsdarstellung für $t = 0$ sehen.

In den Graphiken 2 und 3 zeigt sich, wie die Lösung $y(t, x)$ vom Diffusions-/ Wärmeleit-

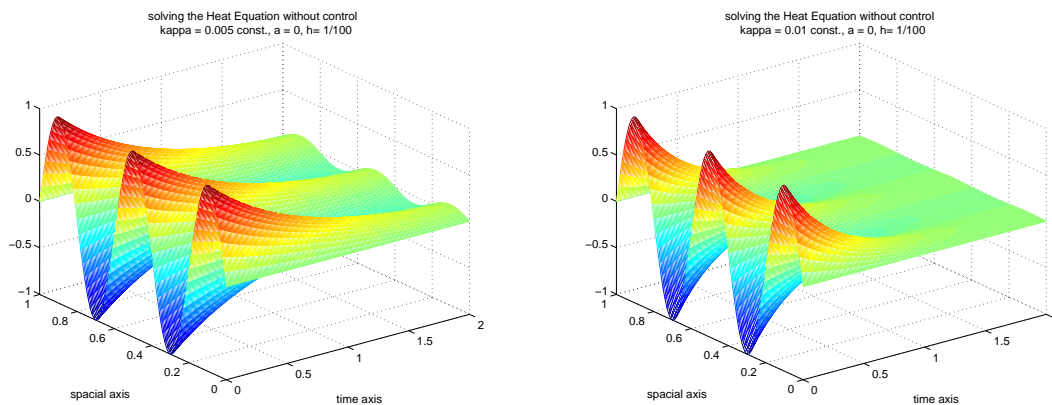


Abbildung 2: Lösungen/Zustände $y(t, x)$ für $\kappa = 0.005$ (links) und $\kappa = 0.01$ (rechts)

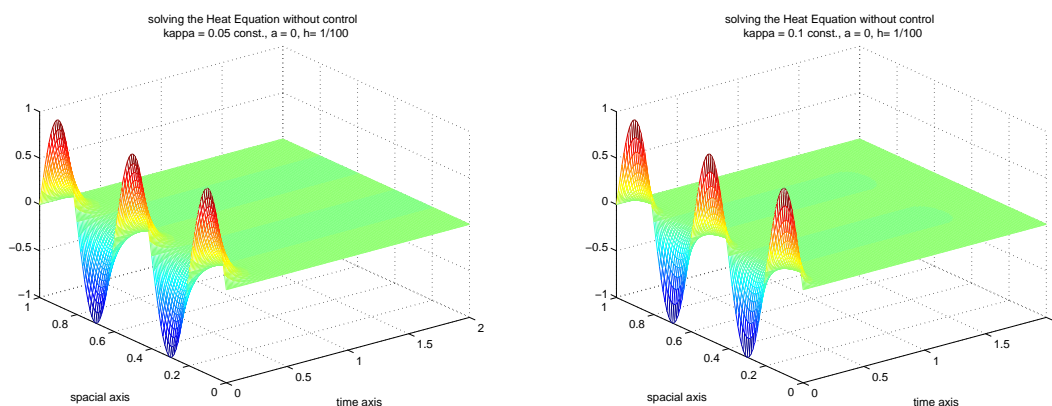


Abbildung 3: Lösungen/Zustände $y(t, x)$ für $\kappa = 0.05$ (links) und $\kappa = 0.1$ (rechts)

koeffizienten κ abhängt. Hierbei kann man sich $y(t, x)$ zum Beispiel als die Temperatur zum Zeitpunkt t am Ort $x \in [0, 1]$, einem Punkt auf einem Stab, vorstellen. Bei relativ großem κ

sieht man tatsächlich eine stärkere Diffusion der Wärme, die Wärme fließt nach außen, dh. an den Stabenden, ab bzw. sie verteilt sich/gleicht sich aus. Bei kleinem κ diffundiert die Wärme langsamer, die Temperatursprünge werden langsamer kleiner. Die rechte Seite der PDGL war hier auf Null gesetzt, was man z.B. als Situation ohne Heizkörper ansehen kann.

3.2 Lösung der PDGL mit manuell eingestellter Steuerung

Im zweiten Schritt wollen wir nun manuell eine Steuerung einbauen, dh. wir verändern die rechte Seite der Differentialgleichung. Weiterhin soll $f \equiv 0$ gelten. Die Steuerung bestimmt sich nicht aus dem System heraus, wie z.B. bei der optimalen Steuerung, sondern wir setzen eine Steuerung u als gegeben voraus, das ist was der Begriff „manuell“ beschreiben soll. Wir betrachten wieder $a = 0$ und κ konstant mit den gleichen Randbedingungen wie zuvor. Diesmal haben wir folgende DGL gegeben:

$$-y_t(t, x) - \frac{d}{dx} \left(\overbrace{\kappa y_{xx}(t, x)} \right) = \chi_1(x)u_1(t) + \chi_2(x)u_2(t) \quad (3.1)$$

wobei wir setzen:

$$\begin{aligned} m &= 2 \\ \chi_1 &= 2 \cdot \mathbf{1}_{[0,0.5]} \\ \chi_2 &= 0.5 \cdot \mathbf{1}_{(0.5,1]} \\ u(t) &= (u_1(t), u_2(t))' = (t^2, t^2)' \end{aligned}$$

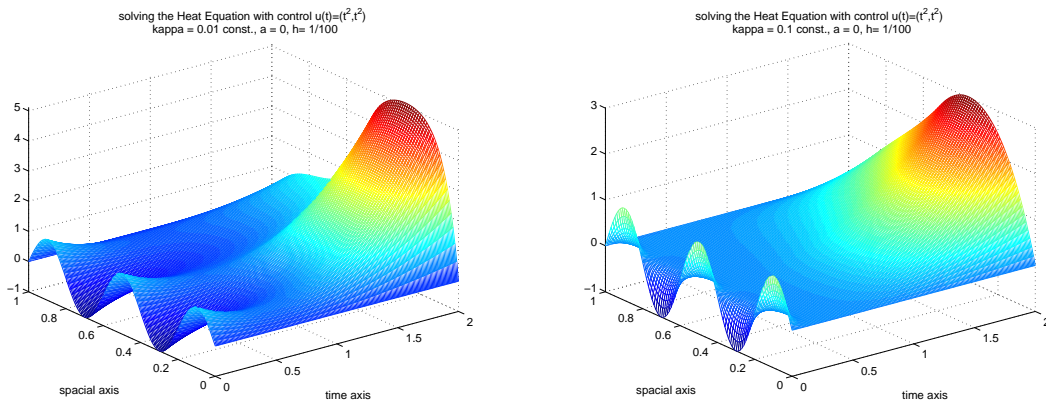


Abbildung 4: Zustand $y(t, x)$ bei manuell gesetzter Steuerung $u(t) = (t^2, t^2)$ und $\kappa = 0.01$ (links) und $\kappa = 0.1$ (rechts)

Man sieht in der Abbildung 4, dass wieder Energie in das System hineingesteckt wird, dh. die Temperatur erhöht sich wieder mit wachsender Zeit t . Für $t \nearrow$ steigt die Steuerung an, da $u_i(t) = t^2$. Auf dem Teilintervall $[0, 0.5]$ wird mehr Steuerung zugelassen/umgesetzt, da für $x \in [0, 0.5]$ gilt $\chi_2(x) + \chi_1(x) = 2$, wohingegen $\chi_2(x) + \chi_1(x) = 0.5 < 2$ für $x \in (0.5, 1]$ gilt. Zudem erkennt man, dass bei größerem Wärmeleitkoeffizienten κ , also stärkerer Diffusion, die

höhere zugeführte Energie (und die damit erzeugte Wärme) von $[0, 0.5]$ sich auch noch auf den Bereich $(0.5, 1]$ auswirkt/diffundiert. Dafür kann man sich die Zustands-Graphiken leicht gedreht anschauen und erhält Abbildung 5.

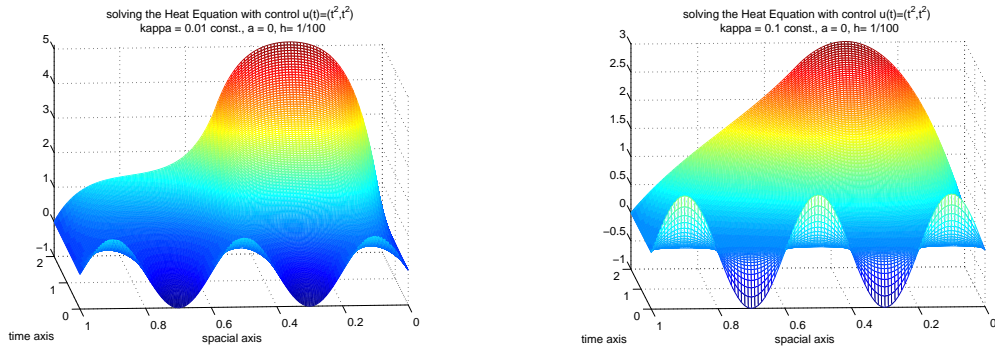


Abbildung 5: gedrehte Ansicht des Zustands bei manuell gesetzter Steuerung

Wir haben nun gesehen, wie das Einstellen einer Steuerung u wirken kann. Die obige Wahl von u war jedoch recht unrealistisch, da wir nicht Energie ins System pumpen wollen, damit die Temperatur $y(t, x)$ wächst und wächst, sondern vielmehr wollen wir so steuern, dass sich y schneller/stärker an die stationäre Lösung \bar{y} mit $A\bar{y} + B\bar{u} = 0$ wobei $\bar{u} = 0$, also $\bar{y} = 0$, annähert. Wir suchen die *optimale Steuerung*, die y möglichst schnell in den stationären Zustand $\bar{y} = 0$ überführt, und das *Rückkopplungsgesetz*, das besagt wie die optimale Steuerung vom optimalen Zustand abhängt.

4 Optimale Steuerung

4.1 Problemstellung

Wir haben die partielle Differentialgleichung (1D) mit $f \equiv 0$ und $a \equiv 0$ gegeben. Durch Diskretisierung in der Ortsdimension mit der Finite Elemente Methode haben wir daraus das System von gewöhnlichen Differentialgleichungen (2.2) erhalten. Da wir nur noch die Situation $f \equiv 0$ betrachten, hat dies die Form

$$Mc'(t) = -Tc(t) + \bar{B}u(t) \quad (4.1a)$$

$$Mc(0) = \bar{c} \quad (4.1b)$$

Am Ende des Abschnitts 2.2 auf Seite 13 wurde beschrieben, dass wenn man dieses System mit implizitem Euler löst und die Iterierten, spaltenweise, in eine Matrix Y schreibt, dass dann gilt:

$$c_i^k = Y_{ik} \approx y(t_k, x_i)$$

Wir schreiben fortan $y(t)$ an Stelle von $c(t)$ und bringen (4.1) auf die Form, welche wir auch in der Vorlesung für Differentialgleichungen als Nebenbedingung eines Optimierungsproblems behandelt haben:

$$\dot{y}(t) = Ay(t) + Bu(t) \quad (4.2a)$$

$$y(0) = y^0 \quad (4.2b)$$

wobei $A = -M^{-1}T$, $B = M^{-1}\bar{B}$ und $y^0 = (y_0(h), \dots, y_0(Nh))^T$.

Mit \bar{y} bezeichnen wir die stationäre Lösung, den stationären Zustand, des Systems (4.2), dh. es gilt $A\bar{y} + B0 = 0$. Da A invertierbar ist, gilt dies nur für den Nullvektor $\bar{y} = 0$. Durch Einsetzen der Steuerung u wollen wir erreichen, dass die Lösung y des Systems eine gewisse Form hat, nämlich „möglichst nahe“ an der stationären Lösung ist. Die Bedeutung von möglichst nahe geht in das zu minimierende Zielfunktional ein, welches daher wie folgt gewählt wird.

$$J(y, u) = \frac{1}{2} \int_0^\infty \int_\Omega |y(t, x)|^2 dx dt + \frac{\sigma}{2} \int_0^\infty \|u(t)\|_{\mathbb{R}^m}^2 dt \quad (4.3)$$

Aus numerischer Sicht brauchen wir jedoch ein endliches Integrationsintervall. Wir wählen unser Rechengebiet $[0, T]$ mit $T > 0$ so, dass für die Lösung des ungesteuerten Problems gilt: $y(T, \cdot) \approx 0$. Das heißt, dass zur Zeit T die gesamte Energie bereits aus dem System „hinausgeflossen“ ist. Nur dann können wir die im theoretischen entwickelten Methoden sinnvoll anwenden.

Wir betrachten nun also folgendes restringiertes Optimierungsproblem:

$$\begin{aligned} & \min J(y, u) \\ & \text{u.d.N } y'(t) = Ay(t) + Bu(t), \quad t \in (0, \infty) \\ & y(0) = y^0 \end{aligned}$$

Es gilt:

$$(y(t))_i \approx y(t, x_i) \text{ mit } x_i = 0 + ih \text{ für } i = 1, \dots, N \quad (4.4)$$

wobei N die Dimension des P1-FE-Ansatzraumes und $h = \frac{1}{N+1}$ die zugehörige örtliche Diskretisierungsschrittweite ist.

Sei (y^*, u^*) die Lösung von $\min J(y, u)$ u.d.N. (4.2). Dann heißt y^* *optimaler Zustand* und u^* *optimale Steuerung*. Der Zustand y^* ist durch die Differentialgleichung eindeutig aus u^* bestimmbar. Wir wollen nun aber u^* als Funktion von y^* angeben, dh. wir wollen ein *Rückkopplungs-/Feedbackgesetz*

$$u^*(t) = -Ky^*(t) \quad \text{mit } K \in \mathbb{R}^{m \times n}$$

Mit diesem Feedback Law vereinfacht sich das restringierte Minimierungsproblem auf das Lösen der Differentialgleichung

$$\begin{aligned} \dot{y}(t) &= (A - BK)y(t), \quad t \in (0, T) \\ y(0) &= y^0 \end{aligned}$$

Aus der Vorlesung wissen wir wie das Feedback Law aussieht für ein quadratisches Zielfunktional der Form

$$J(y, u) = \int_0^{+\infty} y(t)^T Q y(t) + u(t)^T R u(t) dt$$

und eine lineare Differentialgleichung, dh. für ein LQR-Problem². Dies liegt hier vor, wir müssen das Zielfunktional (4.3) nur noch in diese Form umschreiben.

Bestimmung der Matrizen Q und R :

Um Q zu bestimmen, diskretisieren wir das Ortsintegral in J mit der zusammengesetzten Trapezregel.

$$\begin{aligned} \frac{1}{2} \int_0^1 |y(t, x)|^2 dx &= \frac{1}{2} h \left(\frac{1}{2} |y(t, 0)|^2 + \frac{1}{2} |y(t, 1)|^2 + \sum_{i=1}^N |y(t, ih)|^2 \right) \\ &\stackrel{(1D)}{=} \frac{h}{2} \sum_{i=1}^N |y(t, x_i)|^2 \\ &\stackrel{(4.4)}{=} \frac{h}{2} \sum_{i=1}^N |y_i(t)|^2 \\ &= (y_1(t) \cdots y_N(t))^T \underbrace{\begin{pmatrix} \frac{h}{2} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{h}{2} \end{pmatrix}}_{=: Q} \begin{pmatrix} y_1(t) \\ \vdots \\ y_N(t) \end{pmatrix} \end{aligned}$$

Damit gilt

$$Q = \begin{pmatrix} \frac{h}{2} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{h}{2} \end{pmatrix} = \frac{h}{2} I_{N \times N} \in \mathbb{R}^{N \times N} \quad (4.5)$$

und analog wie bei der letzten Gleichheit der Herleitung von Q gilt für R

$$R = \begin{pmatrix} \frac{\sigma}{2} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\sigma}{2} \end{pmatrix} = \frac{\sigma}{2} I_{m \times m} \in \mathbb{R}^{m \times m} \quad (4.6)$$

²Vergleiche Gleichung (2.9) des Vorlesungsskriptes und die dazu gehörende Theorie.

denn damit ist $\frac{\sigma}{2}\|u(t)\|^2 = \sum_{i=1}^m |u_i(t)|^2 = u(t)^T R u(t)$.

Q und R sind symmetrische, positiv definite Matrizen, wenn $\sigma > 0$ ist.

Wir wissen gemäß Vorlesung, dass dann das Feedback Law gegeben ist durch

$$u^*(t) = -K y^*(t) \text{ mit } K = R^{-1} B^T P \quad (4.7)$$

wobei P eine symmetrische, positiv definite $N \times N$ -Matrix und Lösung der algebraischen Riccati-gleichung ist.

Die *algebraische Riccatigleichung*

$$A^T P + P A - P B R^{-1} B^T P + Q = 0 \quad (4.8)$$

lässt sich in Matlab mithilfe von `are.m` lösen. Dies ist ein Matlab interner Solver aus der Control System Toolbox. Der Aufruf $P = \text{are}(A, B R^{-1} B^T, Q)$ liefert die gesuchte Matrix P .

Beachte, wir invertieren R nicht, da R eine Diagonalmatrix ist, muss man einfach nur die Einträge auf der Diagonalen invertieren. Also ist $R^{-1} = \frac{2}{\sigma} I_{m \times m}$.

4.2 Konkretes Lösen des Optimalsteuerproblems

Für die weiteren Betrachtungen soll die Funktion $\kappa : [0, 1] \rightarrow \mathbb{R}_0^+$ nicht mehr nur konstant sein, sondern eine Stufenfunktion. Im ersten Fall setzen wir

$$\kappa(x) = \begin{cases} 0.1 & \text{für } x \in [0, 0.5] \\ 0.01 & \text{für } x \in (0.5, 1] \end{cases} \quad T = 5 \quad \text{und} \quad h = \frac{1}{50} \quad (4.9)$$

und schauen uns an, wie die Lösung des ungesteuerten Systems ($u = 0$ in (1D)) und des Optimalsteuerproblems, $\min J(y, u)$ u.d.N. (4.2), für verschiedene Werte des Gewichtungsfaktors σ von u im Zielfunktional aussieht.

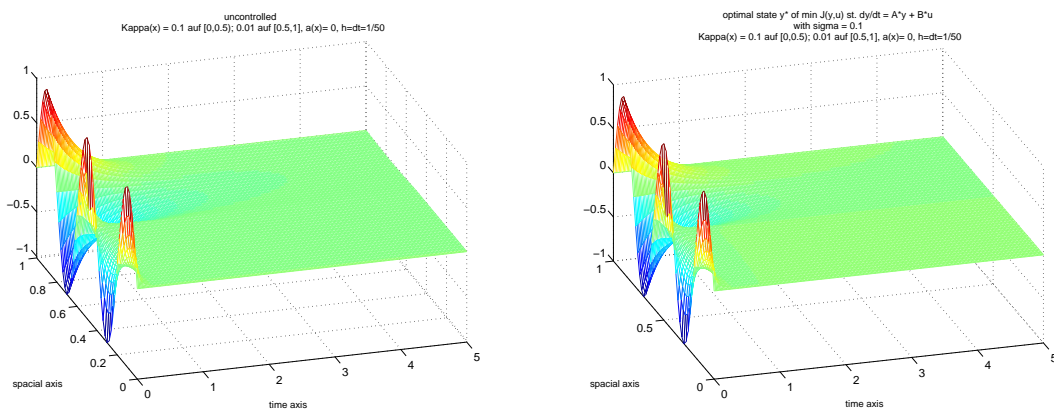


Abbildung 6: 1. Vergleich der Zustände $y(t, x)$ ohne Steuerung und mit optimaler Steuerung

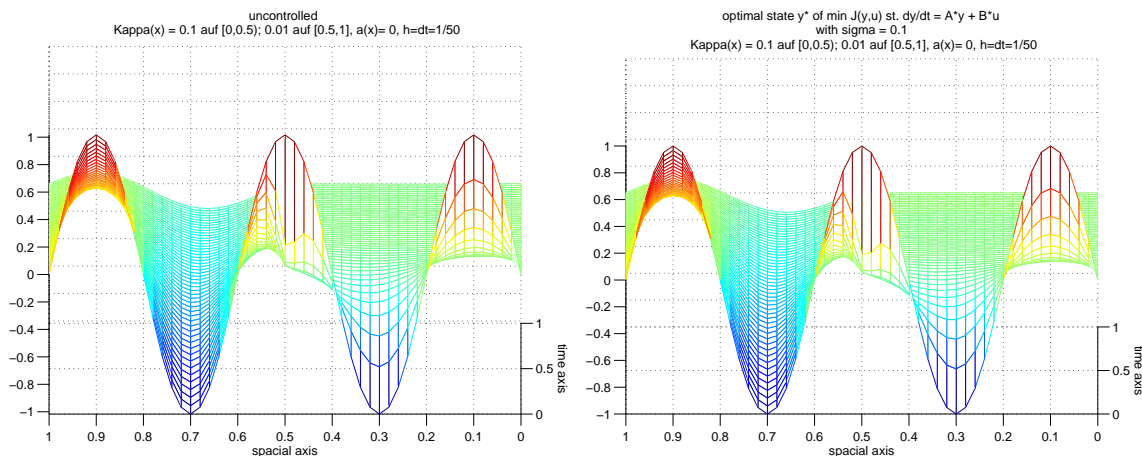


Abbildung 7: obige Zustände $y(t, x)$ aus anderer Perspektive

Mit bloßem Auge erkennt man anhand von Abb. 6 und 7 kaum eine Verbesserung beim Übergang vom ungesteuerten Fall zum optimalgesteuerten mit $\sigma = 0.1$. Wir werden aber später die

Entwicklung der Norm des Endzustands, also der Lösung y ausgewertet zur Zeit $t = T$, und des Verlaufs von $\|y(t, \cdot)\|_{L^2(\Omega)}^2$ für $t \nearrow T$ untersuchen und daran erkennen, was die optimale Steuerung bewirkt.

Beim Betrachten der Schaubilder in Abbildung 8 der optimalen Steuerung u zu zwei verschiedenen vorgegebenen Gewichten σ sieht man, dass bei größerem σ , also stärkerer Gewichtung der Steuerung im Zielfunktional, die optimale Steuerung geringer ausfällt. Man beachte dazu die Ordinate bzw. die unterschiedlichen maximal angenommenen Werte. Dies lässt sich anhand der Form des Zielfunktionals erklären. Bei größerem σ fällt die Steuerung stärker ins Gewicht, also vergrößert das Zielfunktional in größerem Maß. Um das Zielfunktional zu minimieren, wird aufgrund eines eh schon relativ kleinen Zustands dementsprechend nicht unnötig viel Steuerung eingesetzt.

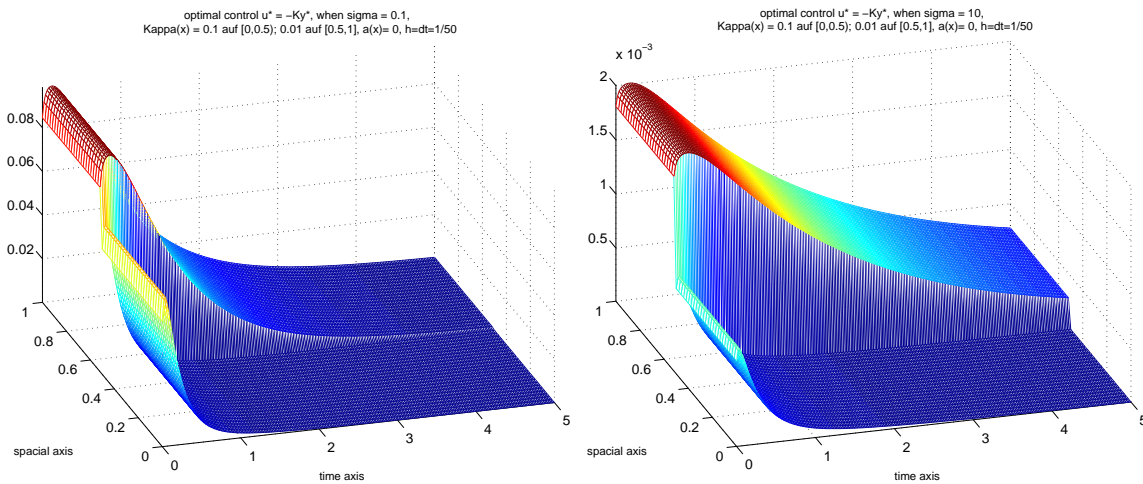


Abbildung 8: Optimale Steuerung bei $\sigma = 0.1$ und $\sigma = 10$

Zum Vergleich lassen wir die Programme noch in einem zweiten Fall, bei anderen Werten der Stufenfunktion κ , durchlaufen und erhalten bei

$$\kappa(x) = \begin{cases} 0.05 & \text{für } x \in [0, 0.5] \\ 0.005 & \text{für } x \in (0.5, 1] \end{cases} \quad T = 5 \quad \text{und} \quad h = \frac{1}{50} \quad (4.10)$$

die Abbildungen 9 bis 11.

Auch in diesem Fall lässt sich aus den Schaubildern von y^* nicht wirklich etwas folgern. Wir sehen aber auch hier den Effekt, den der Wert von σ auf die Ausprägung der Steuerung u hat (siehe vor allem Abbildung 12).

Vergleicht man bei gleichem σ , zum Beispiel $\sigma = 0.1$, die Steuerungsintensität bei verschiedenen κ , dann erkennt man, dass bei niedrigerem κ der über die Zeit maximale angenommene Wert der Steuerung u größer ist als bei größerem Diffusionskoeffizienten κ . Für $\kappa \in \{0.1, 0.05, 0.01, 0.005\}$ fallend, wird nicht nur $\int_0^T \|u(t)\|^2 dt$, sondern auch allgemein $u(t)$ für jedes $t \in [0, T]$ größer.

Ferner verhält sich die Intensität der Steuerung wie erwartet: Unabhängig von σ wird anfänglich relativ stark gesteuert, wenn dann jedoch der Zustand näher dem stationären Zustand kommt, dann wird die Steuerung gedrosselt, dh. $u(t)$ geht gegen 0.

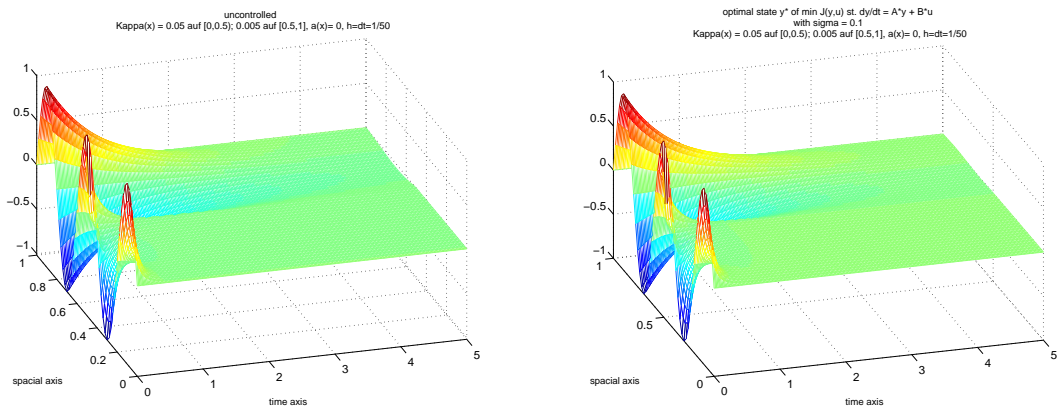


Abbildung 9: 2. Vergleich von optimalen Zuständen

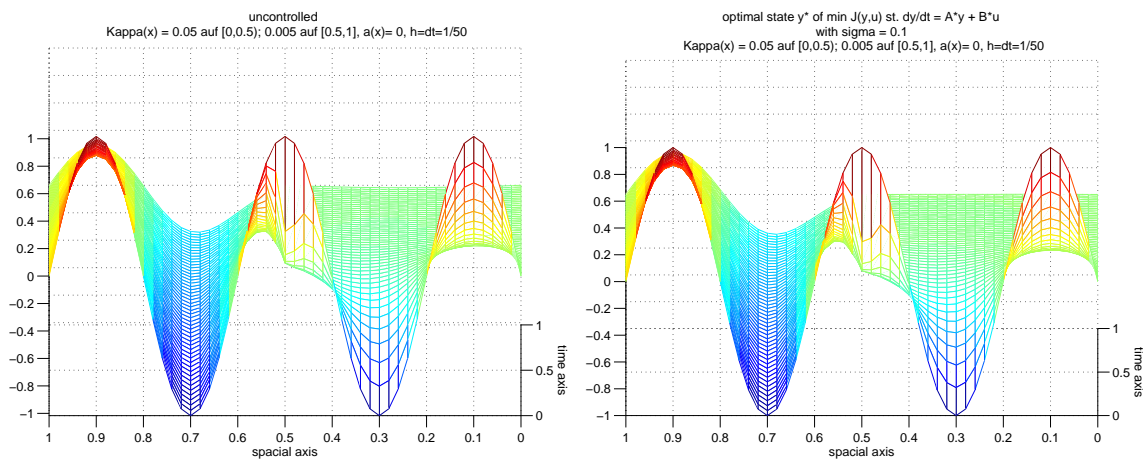


Abbildung 10: obige Schaubilder - gezoomt und gedreht

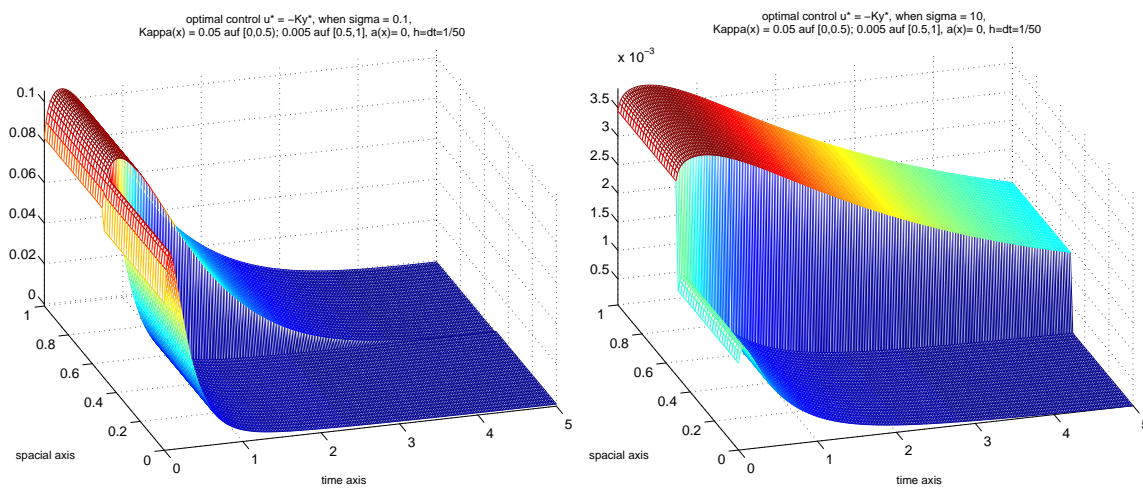


Abbildung 11: Optimale Steuerung bei $\sigma = 0.1$ und $\sigma = 10$ bei kleinerem κ

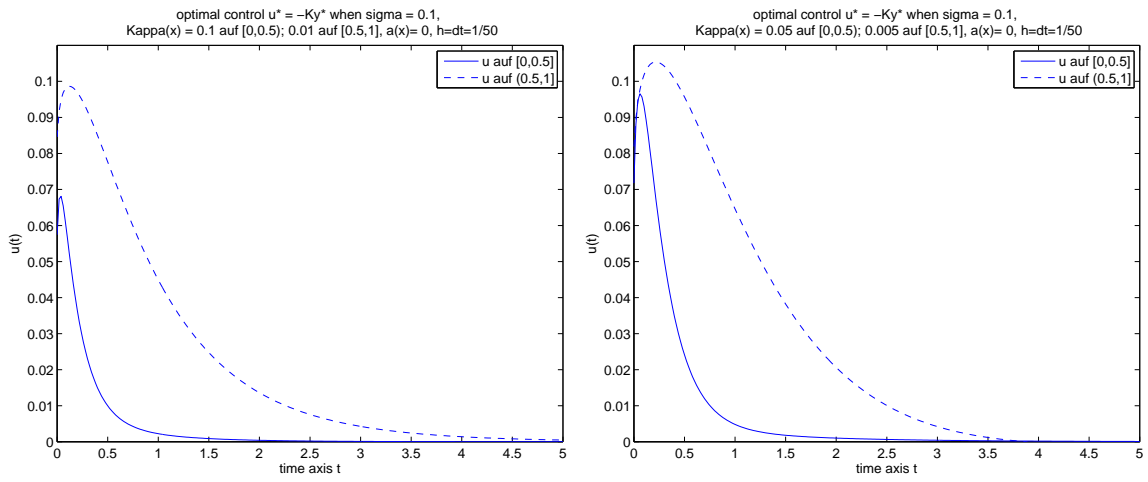


Abbildung 12: Verlauf der optimalen Steuerung u^*

4.3 Vergleich

Um genauere Kenntnisse über den optimalen Zustand y^* zur Zeit T im unkontrollierten bzw. optimalgesteuerten Fall abhängig von σ zu erhalten, lassen wir uns die euklidische Norm der letzten Spalte von Y in eine Tabelle schreiben.

norm01001.txt

```
This file contains the norm of the end state (vector)
[y(t_end,x_1);...;y(t_end,x_N)] for the different cases,
i.e. the uncontrolled system and closed loop systems with different sigma
when kappa(x) = 0.1 auf [0,0.5] und 0.01 auf (0.5,1], a(x)= 0, h=dt=0.01
-----
case          norm([y(t_end,x_1);...;y(t_end,x_n)])
uncontrolled          0.08262706
opt con sigma = 10.000 0.07643605
opt con sigma =  1.000 0.04664279
opt con sigma =  0.100 0.00234996
opt con sigma =  0.010 0.00102579
-----
```

Unser Ziel ist es ja, durch Einsatz der Steuerung u die Lösung y der gegebenen PDGL näher an die stationäre Lösung, hier $\bar{y} = 0$, zu bringen. Wir können anhand der Tabelle sehen, dass die Steuerung den gewünschten Effekt (betrachtet zur Zeit $t = T = 5$) erfüllt, dh. für jedes σ ist $\| (y(T, x_1), \dots, y(T, x_N))^T \|_{2,controlled} < \| (y(T, x_1), \dots, y(T, x_N))^T \|_{2,uncontrolled}$. Ferner sieht man, dass mit kleiner werdendem σ auch die Norm des Endzustands fällt. Das liegt daran, dass bei großem σ die Steuerung mehr zum Wert des Zielfunktional, das es zu minimieren gilt, beiträgt. Da das Anwachsen des hinteren Terms durch stärkere Steuerung (bei großem σ) in keinem Verhältnis steht zum Gewinn, den man durch ein kleineres $\|y(t, \cdot)\|_{L^2(\Omega)}^2$ erhalten würde (da y verhältnismäßig klein ist) wird bei großem σ die Steuerung klein gehalten. Dementsprechend erreichen wir dann auch kein so gutes/kleines Ergebnis wie bei kleinem σ und daher stärkerer Steuerung.

norm0050005.txt

```
This file contains the norm of the end state (vector)
[y(t_end,x_1);...;y(t_end,x_N)] for the different cases,
i.e. the uncontrolled system and closed loop systems with different sigma
when kappa(x) = 0.05 auf [0,0.5] und 0.005 auf (0.5,1], a(x)= 0, h=dt=0.01
-----
case          norm([y(t_end,x_1);...;y(t_end,x_n)])
uncontrolled          0.23142212
opt control 10.000    0.20487393
opt control  1.000    0.10288098
opt control  0.100    0.04918708
opt control  0.010    0.05108161
-----
```

κ ist im 2. Fall kleiner, dh. von alleine (ohne Steuerung) diffundiert die Wärme nicht so stark bzw. schnell wie im 1. Fall. Daher tritt unabhängig davon, ob man steuert und wenn ja mit welchem σ man arbeitet, im 2. Fall ein höherer Normwert des Endzustandes ein.

In den folgenden Schaubildern sieht man die Werte der Abbildungen $t \mapsto \|y(t, \cdot)\|_{L^2(\Omega)}^2$ in den verschiedenen Fällen ohne Steuerung und mit $\sigma \in \{10, 1, 0.1, 0.01\}$. Es zeigt sich, dass über die gesamte Zeit durch Einsatz der optimalen Steuerung u^* der Zustand y^* näher an die stationäre

Lösung $\bar{y} = 0$ herangedrückt wird. Je nachdem wie stark die Steuerung durch σ gewichtet wird, dh. wie die Steuerung in das Zielfunktional eingeht, wird mehr (optimale) Steuerung eingesetzt. Daraus folgt, dass y schneller/stärker zu 0 gedrückt wird.

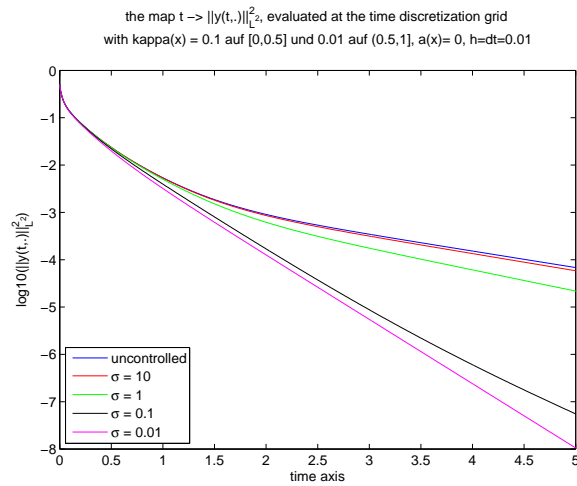


Abbildung 13: L^2 -Norm-Vergleich für $\kappa = 0.1 \cdot \mathbf{1}_{[0,0.5]} + 0.01 \cdot \mathbf{1}_{(0.5,1]}$

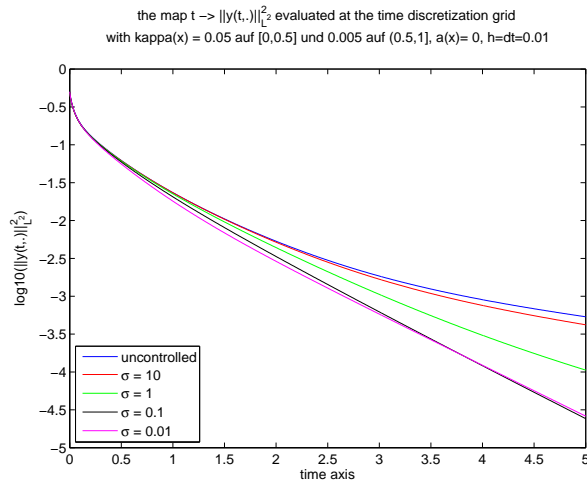


Abbildung 14: L^2 -Norm-Vergleich für $\kappa = 0.05 \cdot \mathbf{1}_{[0,0.5]} + 0.005 \cdot \mathbf{1}_{(0.5,1]}$

Dies sieht man deutlich, wenn man sich einen Ausschnitt der Abbildung 13 auf Seite 25 gezoomt ansieht.

the map $t \rightarrow \|y(t, \cdot)\|_{L^2}^2$, evaluated at the time discretization grid
 with $\kappa(x) = 0.1$ auf $[0, 0.5]$ und 0.01 auf $(0.5, 1]$, $a(x) = 0$, $h = dt = 0.01$

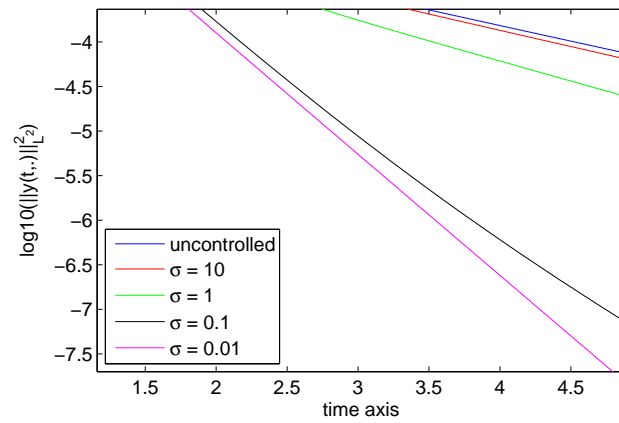


Abbildung 15: Unterschied in L^2 -Norm (Ausschnitt)