

# Kollokation einer Zwei-Punkt Randwertaufgabe unter Verwendung der Bernsteinbasis

Björn Ludwig  
im Studiengang

Bachelor of Science  
Mathematik, Informatik (NF)

an der FernUniversität in Hagen  
im Mai 2017

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Grundlagen</b>	<b>1</b>
2.1	Inhalte aus dem Kurs 01277 <i>Einführung in Computergrafik</i> . . . . .	2
2.2	Inhalte aus dem Kurs 01270 <i>Numerische Mathematik I</i> . . . . .	3
2.3	Inhalte aus dem Kurs 01143 <i>Lineare Algebra</i> . . . . .	3
2.4	Notationen . . . . .	4
<b>3</b>	<b>Die Aufgabe</b>	<b>4</b>
3.1	Beschreibung der verwendeten Methode . . . . .	4
3.1.1	Wahl der Kollokationspunkte . . . . .	4
3.1.2	Der Ansatz . . . . .	5
3.2	Bestimmung des Gleichungssystems . . . . .	5
3.2.1	Die Randbedingungen . . . . .	5
3.2.2	Die Kollokationsbedingungen . . . . .	5
3.3	Lösen des Gleichungssystems . . . . .	8
<b>4</b>	<b>Numerische Beispiele</b>	<b>8</b>
<b>5</b>	<b>Fazit und Ausblick</b>	<b>9</b>

## 1 Einleitung

Wir beschäftigen uns im vorläufigen Entwurf dieser Arbeit mit der Lösung eines Zwei-Punkt Randwertproblems mit einer linearen Differentialgleichung zweiter Ordnung. Zur näherungsweisen Bestimmung der Lösung verwenden wir die Kollokationsmethode und geben schließlich basierend auf der BERNSTEINbasis ein lineares Gleichungssystem an, dessen Lösung die BÉZIER-Punkte eines Polynoms in BÉZIER-Darstellung repräsentiert, welches das gegebene Randwertproblem in bestimmten Schranken löst. Bei der Formulierung des Problems und des allgemeinen Lösungsansatzes stützen wir uns vor Allem auf RUSSEL und SHAMPINE [7] sowie auf REINHARDT [6]. Zur Arbeit mit den BERNSTEIN-Polynomen stellen neben dem Kurs 01277 *Einführung in Computergrafik* der Fernuniversität in Hagen [8] auch BÖHM et al. [10] in den Abschnitten 6 bis 8 und PRAUTZSCH et al. [3] Kapitel 2 wertvolles Rüstzeug zur Verfügung. Die Bestimmung einer Menge bestimmter Kollokationspunkte mit besonders günstigen Eigenschaften ermöglicht uns ein Satz aus dem Kurs 01270 *Numerische Mathematik I* der Fernuniversität in Hagen [4]. Das Verfahren wurde objektorientiert in der Programmiersprache Java unter Verwendung der externen Programmbibliothek `org.apache.commons.math3` implementiert um anhand zweier Beispiele die Näherungseigenschaften des Verfahrens aufzuzeigen.

## 2 Grundlagen

Wir greifen häufig auf Inhalte mehrerer der oben genannten Publikationen zurück und setzen diese als bekannt voraus, stützen uns dabei allerdings vor allem auf die Form der Darstellung in [8], welche sich vermutlich vor Allem an [10] orientiert, und führen deren wesentliche Ergebnisse hier auf.

## 2.1 Inhalte aus dem Kurs 01277 *Einführung in Computergrafik*

Wir bedienen uns der Inhalte des Kapitels 4.3 über die BERNSTEINbasis und BÉZIER-Kurven. Besondere Bedeutung kommt der Basis selbst sowie dem Algorithmus von DE CASTELJAU zu.

**Definition 2.1.1** (BERNSTEIN-Polynome). *Über dem Intervall  $[s, t] \subset \mathbb{R}, s < t$  lässt sich aus den binomischen Formeln entwickeln*

$$(t - s)^n = \left( (t - x) + (x - s) \right)^n = \sum_{i=0}^n \binom{n}{i} (t - x)^{n-i} (x - s)^i.$$

Für die durch  $(t - s)^n$  dividierten Summanden

$${}_s^t B_i^n(x) := \frac{1}{(t - s)^n} \binom{n}{i} (t - x)^{n-i} (x - s)^i, \quad i = 1, \dots, n,$$

gilt  ${}_s^t B_i^n \in \Pi_n, i = 1, \dots, n$  und sie heißen BERNSTEIN-Polynome vom Grad  $n$ .

Aus der Definition lassen sich unmittelbar folgende Aussagen ablesen:

**Korollar 2.1.2** (Eigenschaften der BERNSTEIN-Polynome). *Die BERNSTEIN-Polynome vom Grad  $n$  haben für  $x \in [s, t]$  folgende Eigenschaften:*

- **Partition der 1:**  $\sum_{i=0}^n {}_s^t B_i^n(x) = 1$ .
- **Positivität:**  ${}_s^t B_i^n(x) \geq 0$ .
- **Symmetrie:**  ${}_s^t B_i^n(x) = {}_s^t B_{n-i}^n(t - (x - s))$ .

Darüber hinaus gilt:

**Proposition 2.1.3** (Rekursivität der BERNSTEIN-Polynome). *Seien  ${}_s^t B_i^n, i = 1, \dots, n$ , die BERNSTEIN-Polynome vom Grad  $n$  über dem Intervall  $[s, t]$ , dann gilt:*

$${}_s^t B_i^n(x) = \frac{x - s}{t - s} {}_s^t B_{i-1}^{n-1}(x) + \frac{t - x}{t - s} {}_s^t B_i^{n-1}(x).$$

Die Darstellung einer Funktion mittels der BERNSTEINbasis ist wie folgt definiert.

**Definition 2.1.4** (BÉZIER-Darstellung und Kontrollpolygon). *Die Darstellung einer Funktion*

$$g: [s, t] \rightarrow \mathbb{R}: x \mapsto g(x) := \sum_{i=0}^n {}_s^t B_i^n(x) \cdot b_i, \quad b_i \in \mathbb{R},$$

heißt BÉZIER-Darstellung. Die Punkte  $b_i, i = 0, \dots, n$  heißen BÉZIER-Punkte und definieren das BÉZIER- oder Kontrollpolygon.

Eine Funktion in BÉZIER-Darstellung verläuft durch Anfangs- und Endpunkt ihres Kontrollpolygons.

**Proposition 2.1.5.** *Sei  $g \in \Pi_n$  ein Polynom in BÉZIER-Darstellung mit BÉZIER-Punkten  $b_i, i = 0, \dots, n$ , dann gilt*

$$g(s) = b_0, \quad g(t) = b_n.$$

Funktionswerte lassen sich mittels des folgenden Algorithmus rekursiv numerisch stabil berechnen.

**Algorithmus 2.1.6** (DE CASTELJAU). *Gegeben:* Eine Funktion

$$g: [s, t] \rightarrow \mathbb{R}: x \mapsto g(x) := \sum_{i=0}^n {}^t_s B_i^n(x) \cdot b_i, \quad b_i \in \mathbb{R},$$

in BÉZIER-Darstellung und ein  $x \in [s, t]$ .

$$b_n^0(x) := b_n$$

**for**  $r = 1, \dots, n$  **do**

$$b_{n-r}^0(x) := b_{n-r}$$

**for**  $j = 1, \dots, r$  **do**

$$b_{n-r}^j(x) := \frac{x-s}{t-s} \cdot b_{n-r+1}^{j-1}(x) + \left(1 - \frac{x-s}{t-s}\right) \cdot b_{n-r}^{j-1}(x)$$

**end for**

**end for**

**Ergebnis:**  $g(x) = b_0^n(x)$ .

Ableitungswerte liefert der Algorithmus ebenfalls mit dem folgenden Satz.

**Satz 2.1.7.** Sei  $g \in \Pi_n$  ein Polynom in BÉZIER-Darstellung mit BÉZIER-Punkten  $b_i, i = 0, \dots, n$ , dann ergibt sich die  $i$ -te Ableitung  $g^{(i)}$  von  $g$  aus dem Algorithmus von de Casteljau in der  $(n-i)$ -ten Stufe zu

$$g^{(i)} = \frac{1}{(t-s)^i} \frac{n!}{(n-i)!} \sum_{k=0}^i \binom{i}{k} (-1)^{i-k} \cdot b_k^{n-i}(x).$$

## 2.2 Inhalte aus dem Kurs 01270 *Numerische Mathematik I*

Die Bestimmung von Nullstellen einer Menge orthogonaler Polynome, zum Beispiel der LEGENDRE-Polynome, ermöglicht der folgende Satz aus Kapitel 5.3.4.

**Satz 2.2.1** (Äquivalenz von Nullstellen orthogonaler Polynome und zugeordneter JACOBI-Matrix.). Seien  $I \subset \mathbb{R}$  offen,  $\varrho$  eine Gewichtsfunktion über  $I$  und  $(u_i)_{i=0}^\infty$  eine Folge bezüglich  $\varrho$  orthogonaler Polynome. Dann sind die Nullstellen von  $u_n, n \in \mathbb{N}$ , und die Eigenwerte der zugeordneten JACOBI-Matrix

$$J_{u,n} := \begin{pmatrix} \alpha_0 & \chi_1 & & 0 \\ \chi_1 & \alpha_1 & \ddots & \\ & \ddots & \ddots & \chi_{n-1} \\ 0 & & \chi_{n-1} & \alpha_{n-1} \end{pmatrix}, \quad \chi_i := \frac{\gamma_{i-1} \delta_i}{\gamma_i \delta_{i-1}}, \quad \alpha_i := \frac{\langle \mu_1 u_i, u_i \rangle_\varrho}{\delta_i^2},$$

identisch.

## 2.3 Inhalte aus dem Kurs 01143 *Lineare Algebra*

Weiterhin finden sich einzelne Inhalte aus [9] über den Binomialkoeffizienten.

**Definition 2.3.1** (Binomialkoeffizient). Seien  $n, k \in \mathbb{N}_0$ . Der Binomialkoeffizient  $\binom{n}{k}$  ist definiert durch  $\binom{n}{k} := \frac{n!}{k!(n-k)!}$  für alle  $0 \leq k \leq n$  und  $\binom{n}{k} := 0$  für  $k > n$ .

An einer Stelle in Abschnitt 3.2.2 greifen wir auf die folgende Formel zur Vereinfachung des erhaltenen Ausdrucks zurück.

**Proposition 2.3.2.** *Es ist*

$$\binom{n}{k} + \binom{n}{k+1} = \binom{n+1}{k+1}. \quad (1)$$

## 2.4 Notationen

Wir verwenden folgende nicht allgemein übliche Notationen. Sei  $x \in \mathbb{R}$ , dann schreiben wir für den häufig im Algorithmus von DE CASTELJAU auftretenden Quotienten

**Notation 2.4.1.**  $\mu := \mu(x) = \frac{x-s}{t-s}$ ,

und

**Notation 2.4.2.**  $\mu_j := \mu(\tau_j) = \frac{\tau_j-s}{t-s}$ ,  $j = 1, \dots, k$ ,

mit der Definition der  $\tau_j$ , wie wir sie in Abschnitt 3.1.1 einführen werden.

## 3 Die Aufgabe

Wir betrachten das Zwei-Punkt Randwertproblem

$$y'' + a(x)y' + b(x)y = f(x), \quad x \in I := [s, t] \subset \mathbb{R}, \quad s < t, \quad y(s) = \eta_1, y(t) = \eta_2,$$

mit einer gewöhnlichen linearen Differentialgleichung zweiter Ordnung.

### 3.1 Beschreibung der verwendeten Methode

Wir werden eine polynomiale Näherung  $g \in C^1(I)$  für die Lösung mittels der Kollokationsmethode finden, indem wir geeignete Kollokationspunkte

$$\Delta := \{\tau_1, \dots, \tau_k\} \subset (s, t), \quad \tau_1 < \dots < \tau_k$$

für ein  $k \in \mathbb{N}$  wählen, und dann basierend auf den BERNSTEIN-Polynomen  ${}_s^t B_0^{k+1}, \dots, {}_s^t B_{k+1}^{k+1}$  eine Funktion  $g \in \Pi_{k+1}$  ansetzen, deren Koeffizienten wir durch Lösen eines linearen Gleichungssystems bestehend aus den zwei Rand- und den  $k$  Kollokationsbedingungen ermitteln.

#### 3.1.1 Wahl der Kollokationspunkte

Zunächst wählen wir  $-1 < \rho_1 < \dots < \rho_k < 1$  und setzen dann

$$\tau_i := \frac{s + t + \rho_i(t - s)}{2}, \quad i = 1, \dots, k.$$

So erhalten wir eine Zerlegung des Intervalls  $(s, t)$  in Abhängigkeit der  $\rho_j, j = 1, \dots, k$ , die wir dann nach unseren Wünschen oder Erkenntnissen bestmöglich wählen können.

### 3.1.2 Der Ansatz

Unsere gesuchte Näherung setzen wir an als reelles Polynom

$$g(x) = \sum_{i=0}^{k+1} {}^t B_i^{k+1}(x) \cdot b_i \in \Pi_{k+1}, \quad x \in [s, t],$$

wobei es sich bei den  ${}^t B_i^{k+1}$  um die BERNSTEIN-Polynome vom Grad  $k+1$  über dem Intervall  $[s, t]$  handelt.

## 3.2 Bestimmung des Gleichungssystems

Mit dem Algorithmus von DE CASTELJAU lassen sich sowohl die Funktions- wie auch zusammen mit Satz 2.1.7. die Ableitungswerte von  $g$  rekursiv in Abhängigkeit von den BÉZIER-Punkten bestimmen und so im Zusammenhang mit  $a, b: [s, t] \rightarrow \mathbb{R}$  die  $k$  Kollokationsbedingungen formulieren. Zusammen mit den Randbedingungen ergibt sich daraus das folgende lineare Gleichungssystem

$$\begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ a_{10} & a_{11} & \dots & a_{1k} & a_{1,k+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{k0} & a_{k1} & \dots & a_{kk} & a_{k,k+1} \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_k \\ b_{k+1} \end{pmatrix} = \begin{pmatrix} y(s) \\ f(\tau_1) \\ \vdots \\ f(\tau_k) \\ y(t) \end{pmatrix},$$

mit einer  $(k+2) \times (k+2)$ -Koeffizientenmatrix für den Vektor der BÉZIER-Punkte  $b_0, \dots, b_{k+1}$ .

### 3.2.1 Die Randbedingungen

Die erste und letzte Gleichung unseres linearen Gleichungssystems ergeben sich wegen Proposition 2.1.5. aufgrund der Randbedingungen

$$\eta_1 = y(s) = g(s) = b_0,$$

und

$$\eta_2 = y(t) = g(t) = b_{k+1}.$$

Auch Zusatzbedingungen der Form

$$\begin{aligned} w_{10}y(x_1) + w_{11}y'(x_1) &= c_1, \\ w_{20}y(x_2) + w_{21}y'(x_2) &= c_2, \end{aligned}$$

wären mit geringem Mehraufwand und Rechnungen analog zu denen im Abschnitt 3.2.2 möglich.

### 3.2.2 Die Kollokationsbedingungen

Als Kollokationsbedingungen setzen wir an:

$$g''(\tau_j) + a(\tau_j)g'(\tau_j) + b(\tau_j)g(\tau_j) = f(\tau_j), \quad j = 1, \dots, k,$$

bestimmen also  $g$ , so dass die Differentialgleichung in den Punkten  $\tau_1, \dots, \tau_k$  erfüllt ist. Dazu bestimmen wir zunächst mit dem Algorithmus 2.1.6 von DE CASTELJAU die Funktions- und Ableitungswerte

unserer Näherungslösung und anschließend daraus die Einträge unserer Koeffizientenmatrix. Aus den Definitionen 2.1.1 der BERNSTEIN-Polynome und 2.1.4 von Funktionen in BÉZIER-Darstellung folgt für die Funktionswerte unmittelbar

$$g(x) = b_0^{k+1}(x) = \sum_{i=0}^{k+1} \binom{k+1}{i} (1-\mu)^{k+1-i} \mu^i b_i.$$

Für die erste Ableitung gilt wegen Satz 2.1.7.

$$\begin{aligned} g'(x) &= \frac{k+1}{t-s} (b_1^k(x) - b_0^k(x)) \\ &= \frac{k+1}{t-s} \left( \sum_{i=0}^k \binom{k}{i} (1-\mu)^{k-i} \mu^i b_{i+1} - \sum_{i=0}^k \binom{k}{i} (1-\mu)^{k-i} \mu^i b_i \right) \\ &= \frac{k+1}{t-s} \left( \mu^k b_{k+1} + \sum_{i=1}^k \left( \binom{k}{i-1} (1-\mu)^{k+1-i} \mu^{i-1} - \binom{k}{i} (1-\mu)^{k-i} \mu^i \right) b_i - (1-\mu)^k b_0 \right) \\ &= \frac{k+1}{t-s} \left( \mu^k b_{k+1} + \sum_{i=1}^k \left( \binom{k}{i-1} (1-\mu) - \binom{k}{i} \mu \right) (1-\mu)^{k-i} \mu^{i-1} b_i - (1-\mu)^k b_0 \right) \\ &= \frac{k+1}{t-s} \left( \mu^k b_{k+1} + \sum_{i=1}^k \left( \binom{k}{i-1} - \binom{k}{i-1} \mu - \binom{k}{i} \mu \right) (1-\mu)^{k-i} \mu^{i-1} b_i - (1-\mu)^k b_0 \right) \\ &= \frac{k+1}{t-s} \left( \mu^k b_{k+1} + \sum_{i=1}^k \left( \binom{k}{i-1} - \binom{k+1}{i} \mu \right) (1-\mu)^{k-i} \mu^{i-1} b_i - (1-\mu)^k b_0 \right) \text{ mit (1),} \end{aligned}$$

und für die zweite Ableitung ebenfalls wegen Satz 2.1.7.

$$\begin{aligned} g''(x) &= \frac{(k+1)k}{(t-s)^2} \sum_{j=0}^2 \binom{2}{j} (-1)^{2-j} b_j^{k-1}(x) = \frac{(k+1)k}{(t-s)^2} (b_2^{k-1} - 2b_1^{k-1} + b_0^{k-1}) \\ &= \frac{(k+1)k}{(t-s)^2} \left( \sum_{i=0}^{k-1} \binom{k-1}{i} (1-\mu)^{k-1-i} \mu^i b_{i+2} - 2 \sum_{i=0}^{k-1} \binom{k-1}{i} (1-\mu)^{k-1-i} \mu^i b_{i+1} \right. \\ &\quad \left. + \sum_{i=0}^{k-1} \binom{k-1}{i} (1-\mu)^{k-1-i} \mu^i b_i \right) \\ &= \frac{(k+1)k}{(t-s)^2} \left( \mu^{k-1} b_{k+1} + \left( (k-1)(1-\mu) - 2\mu \right) \mu^{k-2} b_k \right. \\ &\quad \left. + \sum_{i=2}^{k-1} \left( \binom{k-1}{i-2} (1-\mu)^2 - 2 \binom{k-1}{i-1} (1-\mu) \mu + \binom{k-1}{i} \mu^2 \right) (1-\mu)^{k-1-i} \mu^{i-2} b_i \right. \\ &\quad \left. + \left( (k-1)\mu - 2(1-\mu) \right) (1-\mu)^{k-2} b_1 + (1-\mu)^{k-1} b_0 \right). \end{aligned}$$

Damit folgt für die Einträge unserer Koeffizientenmatrix in der  $j$ -ten Zeile,  $j = 1, \dots, k$ , aus den Summanden, die  $b_0$  enthalten

$$\begin{aligned} a_{j0} &= \frac{(k+1)k}{(t-s)^2}(1-\mu_j)^{k-1} - a(\tau_j)\frac{k+1}{t-s}(1-\mu_j)^k + b(\tau_j)(1-\mu_j)^{k+1} \\ &= (1-\mu_j)^{k-1} \left( \frac{(k+1)k}{(t-s)^2} - a(\tau_j)\frac{k+1}{t-s}(1-\mu_j) + b(\tau_j)(1-\mu_j)^2 \right) \\ &= (1-\mu_j)^{k-1} \left( \frac{(k+1)k}{(t-s)^2} + (1-\mu_j) \left( b(\tau_j)(1-\mu_j) - a(\tau_j)\frac{k+1}{t-s} \right) \right), \end{aligned}$$

und analog

$$\begin{aligned} a_{j,k+1} &= \frac{(k+1)k}{(t-s)^2}\mu_j^{k-1} - a(\tau_j)\frac{k+1}{t-s}\mu_j^k + b(\tau_j)\mu_j^{k+1} \\ &= \mu_j^{k-1} \left( \frac{(k+1)k}{(t-s)^2} - a(\tau_j)\frac{k+1}{t-s}\mu_j + b(\tau_j)\mu_j^2 \right) \\ &= \mu_j^{k-1} \left( \frac{(k+1)k}{(t-s)^2} + \mu_j \left( b(\tau_j)\mu_j - a(\tau_j)\frac{k+1}{t-s} \right) \right), \end{aligned}$$

aus den Koeffizienten von  $b_{k+1}$ , sowie

$$\begin{aligned} a_{j1} &= \frac{(k+1)k}{(t-s)^2} \left( (k-1)\mu_j - 2(1-\mu_j) \right) (1-\mu_j)^{k-2} \\ &\quad + a(\tau_j)\frac{k+1}{t-s}(1-(k+1)\mu_j)(1-\mu_j)^{k-1} + b(\tau_j)(k+1)(1-\mu_j)^k \mu_j \\ &= (k+1)(1-\mu_j)^{k-2} \left( \frac{k}{(t-s)^2} \left( (k-1)\mu_j - 2(1-\mu_j) \right) \right. \\ &\quad \left. + a(\tau_j)\frac{1}{t-s}(1-(k+1)\mu_j)(1-\mu_j) + b(\tau_j)(1-\mu_j)^2 \mu_j \right), \\ a_{jk} &= \frac{(k+1)k}{(t-s)^2} \left( (k-1)(1-\mu_j) - 2\mu_j \right) \mu_j^{k-2} + a(\tau_j)\frac{k+1}{t-s}(k-(k+1)\mu_j)\mu_j^{k-1} \\ &\quad + b(\tau_j)(k+1)\mu_j^k(1-\mu_j) \\ &= (k+1)\mu_j^{k-2} \left( \frac{k}{(t-s)^2} \left( (k-1)(1-\mu_j) - 2\mu_j \right) + a(\tau_j)\frac{1}{t-s}(k-(k+1)\mu_j)\mu_j \right. \\ &\quad \left. + b(\tau_j)\mu_j^2(1-\mu_j) \right), \end{aligned}$$

aus den Summanden, die  $b_1$  und  $b_k$  enthalten. Für die anderen BÉZIER-Punkte folgt

$$\begin{aligned} a_{ji} &= \frac{(k+1)k}{(t-s)^2} \left( \binom{k-1}{i-2}(1-\mu_j)^2 - 2\binom{k-1}{i-1}(1-\mu_j)\mu_j + \binom{k-1}{i}\mu_j^2 \right) (1-\mu_j)^{k-1-i}\mu_j^{i-2} \\ &\quad + a(\tau_j)\frac{k+1}{t-s} \left( \binom{k}{i-1} - \binom{k+1}{i}\mu_j \right) (1-\mu_j)^{k-i}\mu_j^{i-1} \\ &\quad + b(\tau_j)\binom{k+1}{i}(1-\mu_j)^{k+1-i}\mu_j^i, \quad i = 2, \dots, k-1. \end{aligned}$$



Tabelle 1: Fehler  $E_k^\nu$  von Beispiel 4.1.

$\nu$	$k$									
	1	2	3	4	5	6	7	8	9	10
0	2,48e-1	8,02e-2	6,09e-2	1,22e-2	1,49e-3	4,64e-4	5,90e-5	1,43e-5	1,81e-6	4,51e-7
1	0,0	6,40e-1	1,91e-1	1,38e-1	2,13e-2	1,21e-2	1,16e-3	7,62e-4	5,86e-5	3,82e-5
2	4,96e-1	1,60e-1	1,22e-1	2,44e-2	2,98e-3	9,29e-4	1,18e-4	2,86e-5	3,61e-6	9,03e-7

### 3.3 Lösen des Gleichungssystems

Mit einem geeigneten Verfahren lässt sich dann die eindeutige Näherungslösung für das Randwertproblem ermitteln. Hierzu kann zum Beispiel der GAUSS-Algorithmus mit anschließender Rückwärtsersetzung verwendet werden.

## 4 Numerische Beispiele

Zur Verifizierung der Güte unserer Näherungslösung betrachten wir zwei Beispiele aus [5]. Wir stellen dazu für verschiedene Werte von  $k \in \mathbb{N}$  die Fehler

$$E_k^\nu := \|g^{(\nu)} - u^{(\nu)}\|_\Delta = \max_{j=1,\dots,k} |g^{(\nu)}(\tau_j) - u^{(\nu)}(\tau_j)|, \quad \nu = 0, 1, 2,$$

dar. Dabei ist  $u: [s, t] \rightarrow \mathbb{R}$  jeweils die exakte Lösung der betreffenden Randwertaufgabe. Zur Festlegung unserer Kollokationspunkte wählen wir in Anlehnung an [1] und [2] für  $\rho_1, \dots, \rho_k$  die Nullstellen des  $k$ -ten LEGENDRE-Polynoms. Diese Bestimmen wir mittels Satz 2.2.1 als Eigenwerte der zugehörigen JACOBI-Matrix

$$J_{P,k} := \begin{pmatrix} 0 & \frac{1}{\sqrt{3}} & & & 0 \\ \frac{1}{\sqrt{3}} & 0 & \frac{2}{\sqrt{15}} & & \\ & \frac{2}{\sqrt{15}} & 0 & \ddots & \\ & & \ddots & \ddots & \frac{k-1}{\sqrt{4(k-1)^2-1}} \\ 0 & & & \frac{k-1}{\sqrt{4(k-1)^2-1}} & 0 \end{pmatrix}.$$

Zur Veranschaulichung des Konvergenzverhaltens haben wir jeweils die Graphen aller Näherungslösungen  $g^1, \dots, g^r$  dargestellt, bis für  $i \geq r$  visuell  $g^i$  kaum von der exakten Lösung zu unterscheiden ist. Zur Lösung des gebildeten Gleichungssystems verwenden wir die in der externen Programmbibliothek `org.apache.commons.math3` bereitgestellten Algorithmen zur Ermittlung der  $LR$ -Faktorisierung unserer quadratischen Koeffizientenmatrix und zum zweistufigen Lösen des aus  $Ax = c$  resultierenden Gleichungssystems  $LRx = P^{-1}c$  durch Vor- und Rückwärtsersetzung.

#### Beispiel 4.1.

$$y''(x) - 2y(x) = 4x^2 \exp(x^2), \quad y(-1) = y(1) = 0,$$

mit der exakten Lösung

$$u(x) = \exp(x^2) - 0.623979053607 \left( \exp(\sqrt{2}x) + \exp(-\sqrt{2}x) \right).$$

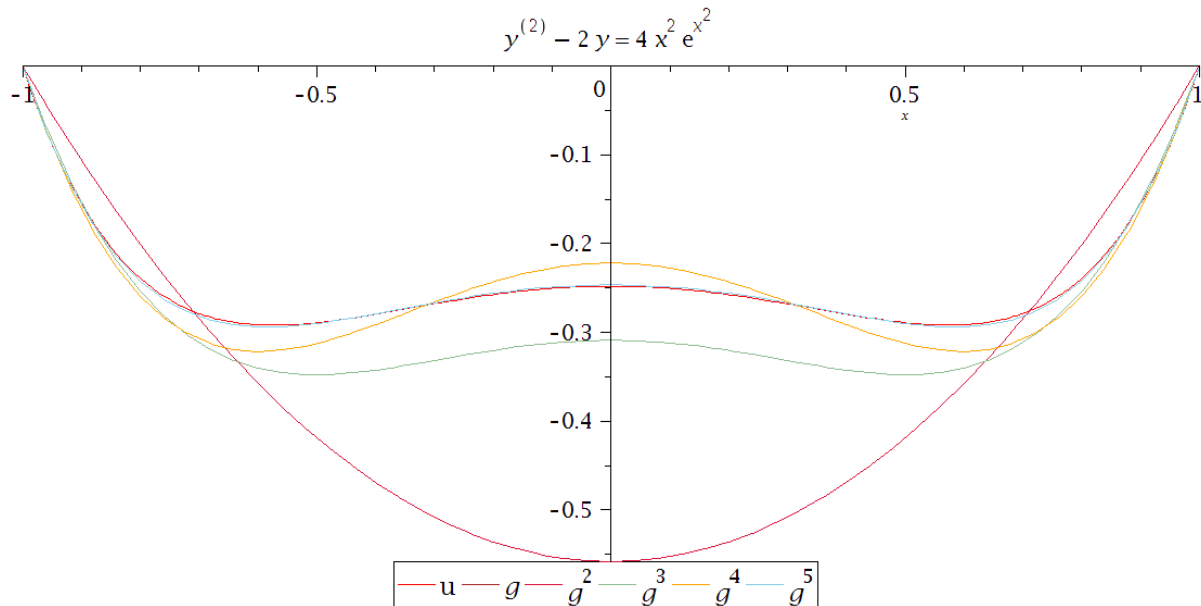


Abbildung 1: Die Graphen der exakten Lösung  $u$  und der Näherungslösungen von Beispiel 4.1 für  $k = 1, \dots, 5$ .

Tabelle 2: Fehler  $E_k^\nu$  von Beispiel 4.2.

$\nu$	$k$									
	1	2	3	4	5	6	7	8	9	10
0	2,87e-2	1,40e-2	5,05e-4	5,87e-5	8,09e-7	1,38e-7	1,41e-9	2,86e-10	1,88e-12	4,16e-13
1	0,0	1,16e-1	2,58e-3	1,59e-3	1,45e-5	9,38e-6	4,64e-8	3,22e-8	1,02e-10	7,27e-11
2	1,15e-1	5,62e-2	2,02e-3	2,35e-4	3,23e-6	5,52e-7	5,65e-9	1,14e-9	7,52e-12	1,68e-12

### Beispiel 4.2.

$$y''(x) - 4y(x) = 4 \cosh(1), \quad y(0) = y(1) = 0,$$

mit der exakten Lösung

$$u(x) = \cosh(2x - 1) - \cosh(1).$$

## 5 Fazit und Ausblick

Wir haben eine Näherungslösung bestimmt und die zu erwartenden Eigenschaften des Verfahrens beispielhaft beobachten können. Aufbauend auf den Ergebnissen werden wir die Untersuchung der Aufgabe in folgenden Bereichen weiterführen

1. Erweiterung der Untersuchungen auf stückweise polynomiale Näherungslösungen.
2. Implementierung des Verfahrens mit beliebiger Genauigkeit.
3. Vergleichende Fehlerbetrachtung bei Verwendung gleichabständiger Kollokationspunkte oder von GAUSS-LEGENDRE-Punkten.

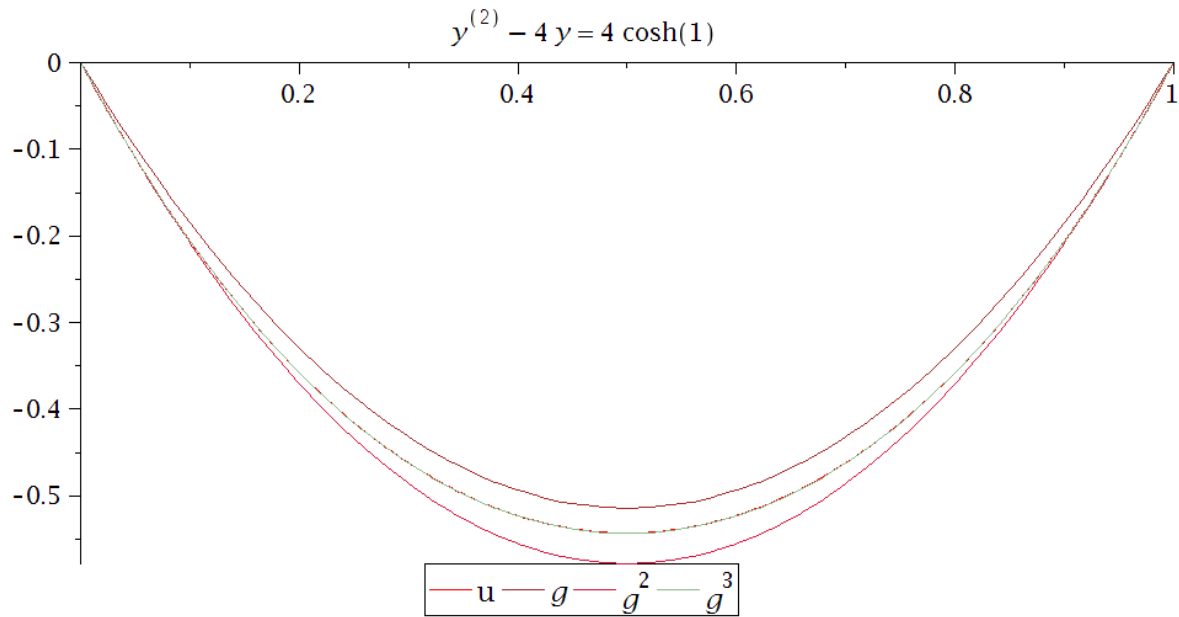


Abbildung 2: Die Graphen der exakten Lösung  $u$  und der Näherungslösungen von Beispiel 4.2 für  $k = 1, \dots, 3$ .

## Literatur

- [1] BOOR, C. DE: *A Practical Guide to Splines*. Springer-Verlag, 1978.
- [2] BOOR, C. DE und B. SCHWARTZ: *Collocation at Gaussian Points*. SIAM Journal on Numerical Analysis, 10(4):582–606, 1973.
- [3] H. PRAUTZSCH, W. BÖHM UND M. PALUSZNY: *Bézier and B-Spline Techniques*. Springer, 2002.
- [4] LINSS, T.: *Numerische Mathematik I*. FernUniversität in Hagen, Kurs 01270, Version Sommersemester 2016.
- [5] MÜLLENHEIM, G.: *Numerische Lösung von Zwei-Punkt Randwertproblemen durch Kollokation mit Splinefunktionen*. Doktorarbeit, Mathematisch-Geographische Fakultät der Katholischen Universität Eichstätt, 1986.
- [6] REINHARDT, H.-J.: *Numerik gewöhnlicher Differentialgleichungen*. De Gruyter, 2012.
- [7] RUSSELL, R. D. und L. F. SHAMPINE: *A collocation method for boundary value problems*. Numerische Mathematik, 19(1):1–28, 1972.
- [8] STRASSER, W.: *Einführung in Computergrafik*. FernUniversität in Hagen, Kurs 01277, Version Wintersemester 2014–2015.
- [9] UNGER, L.: *Lineare Algebra*. FernUniversität in Hagen, Kurs 01143, Version Wintersemester 2016–2017.
- [10] W. BÖHM, G. FARIN, J. KAHMANN: *A survey of curve and surface methods in CAGD*. Computer Aided Geometric Design, 1:1–60, 1984.