Klassische orthogonale Polynome

Bachelorarbeit von Martin Meisrimel
Betreuer: Prof. Dr. W. Koepf

Universität Kassel

1. Juli 2008

Inhaltsverzeichnis

1 Einleitung

Die folgende Bachelorarbeit basiert auf dem Script der Vorlesung „Computeralgebra und Orthogonale Polynome“ [COP] von Prof. Dr. Wolfram Koepf, welche im Wintersemester 2005/2006 an der Universität Kassel gehalten wurde.
Die mathematische Modellierung physikalischer Prozesse erfordert oft das Lösen relativ komplexer Differential- und Integralgleichungen, welche sich nur in einigen Fällen direkt lösen lassen. Aus diesem Grund wird das ursprüngliche Problem vereinfacht, um erste Ergebnisse über die meisten qualitativen Eigenschaften des zu erforschenden Prozesses und die Wirkung der wichtigen Einflussfaktoren zu erhalten. In einigen Fällen erhält man das vereinfachte Problem in einer analytischen Form, was für die weitere Erforschung sehr praktikabel ist. Viele Modellierungsprobleme der Atom-, Molekular-, und Nuklearphysik sowie der Elektrodynamik und Akustik lassen sich auf die so genannte Eigenwertdifferentialgleichung

$$\sigma(x)y''(x) + \tau(x)y'(x) + \lambda_n y(x) = 0$$

reduzieren, wobei $\sigma(x)$ ein Polynom vom Grad ≤ 2, $\tau(x)$ ein Polynom vom Grad ≤ 1 und λ_n eine Konstante ist. In dieser Arbeit werden Polynomlösungen $P_n(x) = k_n x^n + k'_n x^{n-1} + k''_n x^{n-2} \ldots, k_n \neq 0$ dieser Differentialgleichung betrachtet. Diese Polynomlösungen, welche aus den Laguerre-, Hermite-, Bessel- und Jacobipolynomen bestehen, werden die klassischen orthogonalen Polynome genannt. Es wird sich herausstellen, dass alle diese Polynome bis auf die Besselpolynome in einem bestimmten reellen Intervall $[s, t]$ bzgl. einer gegebenen Gewichtsfunktion $w(x)$ die Orthogonalitätsbedingung

$$\int_s^t P_n(x)P_k(x)w(x)dx = 0$$

für $n \neq k$ erfüllen.
Zuvor werden wir jedoch im ersten Abschnitt allgemein Systeme orthogonaler Funktionen φ_n, welche in einem Intervall $[s, t]$ bzgl. einer Gewichtsfunktion $w(x)$ orthogonal sind, definieren und betrachten. Ein solches System werden wir Orthogonalsystem (OS) nennen und mit $(\varphi_n)_{n\geq 0}$ bezeichnen. Als erste Konsequenz wird sich zeigen, dass alle Funktionen φ_n eines solchen Orthogonalsystems linear unabhängig sind. Zum Schluss werden wir das Gram-Schmidtsche Orthogonalisierungsverfahren kennen lernen, welches einen Algorithmus liefert, der aus einem gegebenen linear unabhängigen Funktionensystem $(f_n(x))_{n\geq 0}$ ein Orthogonalsystem bestimmt.
Im darauf folgenden Abschnitt werden wir uns auf OS beschränken, die für $n \geq 0$ aus Polynomen der Form $\varphi_n(x) := P_n(x) = k_n x^n + k'_n x^{n-1} + k''_n x^{n-2} \ldots, k_n \neq 0$ bestehen, welche wir orthogonale Polynomsysteme (OPS) nennen. Diese Einschränkung erlaubt es nun, weitreichende Struktureigenschaften herzuleiten. Eine ganz zentrale Rolle spielt dabei die Dreitermrekursion

$$P_{n+1}(x) = (A_n x + B_n)P_n(x) - C_n P_{n-1}(x) \quad (n \geq 0),$$

welche für alle OPS gilt und die die Beziehung der einzelnen Polynome eines OPS offenlegt sowie eine Möglichkeit bietet, ein Polynom $P_n(x)$ aus seinen Vorgängern zu

bestimmen, und der Satz von Farvard, welcher besagt, dass jedes Polynomsystem $P_n(x)$, das die Dreitermrekursion erfüllt, ein OPS ist.
Im Kapitel *klassische orthogonale Polynome* wird sich zeigen, dass sich die klassischen OPS für den Ansatz $\sigma(x) = ax^2 + bx + c$, $\tau(x) = dx + e$ und $\lambda_n \in \mathbb{R}$ eindeutig anhand der Variablen (a, b, c, d, e) klassifizieren lassen. Dies erhalten wir durch Betrachtung der Nullstellen von $\sigma(x)$. Die vier Möglichkeiten $\sigma(x)$ konstant, einfache Nullstelle, doppelte Nullstelle und zwei Nullstellen liefern dann die einzelnen klassischen OPS. Weiterhin erfüllt die Gewichtsfunktion $w(x)$ die Pearsonsche Differentialgleichung

$$(\sigma(x)w(x))' = \tau(x)w(x).$$

Lösen dieser Differentialgleichung liefert dann die Gewichtsfunktionen $w(x)$ der klassischen orthogonalen Polynome. Im Anschluss werden wir zeigen, dass die Lösungsfamilien $P_n(x)$ der Eigenwertdifferentialgleichung bzgl. der Parameter $\sigma(x)$, $\tau(x)$ und der Gewichtsfunktion $w(x)$ ein OPS darstellen sowie die Umkehrung. Als wichtiges Zwischenresultat des Umkehrungsbeweises erhalten wir die Rekursionsgleichung

$$(dn + a)\mu_{n+1} = -(nb + e)\mu_n - nc\mu_{n-1}$$

der Momente von $w(x)$.
Im darauf folgenden Abschnitt wollen wir explizite Darstellungen der klassischen orthogonalen Polynome herleiten. Dazu betrachten wir Potenzreihenentwicklungen. Es wird sich herausstellen, dass bei geeigneter Wahl des Entwicklungspunktes alle Polynomsysteme hypergeometrische Potenzreihen besitzen. Durch Einsetzen der Darstellungen für $\sigma(x)$ und $\tau(x)$ mittels des Ansatzes $P_n(x) = \sum_{k=0}^{\infty} A_k x^k$ in die Eigenwertdifferentialgleichung erhalten wir mit *Mathematica* unter Verwendung der Funktion *DEtoRE* aus dem *SpecialFunctions* Package die Rekursion

$$-(n - k)(an + d - a + ak)A_k + (k + 1)(bk + e)A_{k+1} + c(k + 1)(k + 2)A_{k+2} = 0.$$

Diese liefert für die Werte (a, b, c, d, e) dann nach einigen algebraischen Umformungen bzw. Transformationen holonome Rekursionen erster Ordnung, woraus wir dann mit der hypergeometrischen Koeffizientenformel explizite Darstellungen für die einzelnen Polynomsysteme herleiten können. Im zweiten Teil werden dann mit den selben Methoden aus der Rekursiongleichung der Momente Darstellungen für die Momente der klassischen orthogonalen Polynome, bis auf die Besselpolynome, hergeleitet.
Das zentrale Thema des nächsten Abschnitts sind Struktureigenschaften. Es wird sich zeigen, dass man sämtliche Strukturbeziehungen wie Ableitung, Stammfunktion usw. mittels Rekursionen beschreiben kann. Besonders vorteilhaft ist dabei, dass man sogar direkte Formeln für die Koeffizienten sämtlicher Rekursionen durch Koeffizientenvergleiche herleiten kann. *Mathematica* spielt hier eine entscheidende Rolle, da die Berechungen aufgrund ihres Umfangs kaum in akzeptabler Zeit per Hand durchgeführt werden könnten. Im letzten Teil werden wir dann noch die Rodriguesformel als eine wichtige Charakterisierung kennen lernen. Sie besagt, dass sich jedes klassische OPS darstellen lässt in der Form:

$$P_n(x) = \frac{E_n}{w(x)} \frac{d^n}{dx^n} \left(\sigma(x)^n w(x)\right).$$

Die Umkehrung soll ebenfalls gezeigt werden.
Im Abschnitt Verbindungskoeffizienten werden wir uns ausgiebig mit der Fragestellung beschäftigen, wie man Informationen aus einem Polynomsystem $(Q_m(x))_{m\geq 0}$ auf ein anderes Polynomsystem $(P_n(x))_{n\geq 0}$ übertragen kann. Aufgrund der linearen Unabhängigkeit der Monome erhalten wir eine Beziehung der Form:

$$P_n(x) = \sum_{k=0}^{n} C_m(n) Q_m(x).$$

Die Koeffizienten $C_m(n)$ nennt man Verbindungskoeffizienten zwischen den Polynomsystemen $P_n(x)$ und $Q_m(x)$. Wir werden sehen, dass man die Beziehungen zwischen den Verbindungskoeffizienten mittels dreier Rekursionsgleichungen, genannt Kreuzregeln, beschreiben kann. Durch Elimination erhält man dann reine Rekursionsgleichungen bzgl. m und n.
Als Nächstes werden wir das spezielle Verbindungsproblem $P_n(x) = x^n$ betrachten. Dieses Verbindungsproblem ist von besonderem Interesse, da man in vielen Anwendungen ein gegebenes Polynom bzgl. eines gegebenen OPS $Q_m(x)$ entwickeln möchte. Für diesen Fall sind einfache Formeln für die Potenzen von x^n als Linearkombinationen des OPS $Q_m(x)$ sehr hilfreich. Es wird sich herausstellen, dass wir für diesen Spezialfall die zuvor hergeleiteten Kreuzregeln modifizieren können und dann wieder Rekursionsgleichungen erhalten. Die Potenzdarstellungen erhalten wir dann als Folgerung aus diesen Rekursionen. Als Konsequenz der Potenzdarstellungen können wir dann wiederum direkte Formeln für die Verbindungskoeffizienten herleiten. Zentrale Bedeutung bei der Herleitung hat hier der Zeilbergeralgorithmus.
Im letzten Abschnitt werden erzeugende Funktionen, gegeben durch

$$F(z) = \sum_{k=0}^{\infty} P_k(x) z^k,$$

mittels Computeralgebra herleitet. Dabei zeigt sich, dass man für alle klassischen OPS solche erzeugenden Funktionen herleiten kann, diese aber in einigen Fällen aufgrund sehr komplexer Differentialgleichungen nicht ohne weitere Überlegungen lösbar sind.
Sämtliche in *Mathematica* durchgeführten Berechnungen befinden sich als *Mathematica Notebooks* auf der beigefügten CD.

2 Systeme orthogonaler Funktionen

2.1 Skalarprodukte

Aus der linearen Algebra ist bekannt, dass zwei Vektoren orthogonal zueinander sind, wenn ihr Skalarprodukt Null ist. Man kann nun auch die integrierbaren Funktionen $R[s,t]$ eines reellen Intervalls $[s,t]$ als einen unendlich-dimensionalen Vektorraum auffassen und in diesem, für ein beliebiges Intervall $[s,t]$, durch

$$\langle f,g\rangle := \int_s^t f(x)g(x)dx \tag{1}$$

oder allgemeiner

$$\langle f,g\rangle := \int_s^t f(x)g(x)w(x)dx \tag{2}$$

ein Skalarprodukt erklären, wobei $w(x)$ eine in $[s,t]$ nichtnegative Funktion darstellt, welche wir die Gewichtsfunktion des Skalarprodukts nennen. Das Intervall $[s,t]$ kann im zweiten Fall, bei geeigneter Wahl der Gewichtsfunktion $w(x)$, entartet sein, also $s = -\infty$, $t = \infty$ oder beides gelten.
Zu jeder Gewichtsfunktion $w(x)$ kann man ebenfalls eine zugehörige Momentenfolge durch

$$\mu_n := \int_s^t x^n w(x)dx = \langle x^n, 1\rangle \quad (n \in \mathbb{N}_{\geq 0})$$

erklären. Falls $s = -\infty$ oder $t = \infty$, so fordern wir zusätzlich $\mu_n < \infty$.

Es stellt sich heraus, dass $\langle f,g\rangle$ in allen Fällen den Eigenschaften eines Skalarprodukts genügt, insbesondere also auch mit

$$\|f\| := \sqrt{\langle f,f\rangle}$$

die Eigenschaft einer Norm besitzt, wobei Funktionen die „fast überall“ miteinander übereinstimmen miteinander identifiziert werden.

2.2 Orthogonalsysteme

Wir wollen nun Orthogonalsysteme betrachten und definieren dazu:

Definition 2.1 (Orthogonalsystem)
Ist $\langle f,g\rangle$ ein Skalarprodukt auf einem Intervall $[s,t]$, so nennen wir eine Familie von Funktionen $\varphi_n : [s,t] \to \mathbb{R}$ $(n \in \mathbb{N}_{\geq 0})$ ein Orthogonalsystem (OS) oder eine orthogonale Familie, wenn $\langle\varphi_n, \varphi_m\rangle = 0$ für $n \neq m$ und $\|\varphi_n\|^2 = \langle\varphi_n, \varphi_n\rangle \neq 0$ für $n \in \mathbb{N}_{\geq 0}$ gilt.

Ist ferner $\|\varphi_n\|^2 = 1$ für $n \in \mathbb{N}_{\geq 0}$, so wird das System ein Orthonormalsystem (ONS) genannt. Für ein ONS gilt

$$\langle \varphi_n, \varphi_m \rangle = \delta_{nm} = \begin{cases} 0 & \textit{falls } n \neq m \\ 1 & \textit{falls } n = m \end{cases},$$

wobei δ_{nm} das Kroneckersymbol genannt wird.

Wir betrachten nun im Folgenden ONS $(\varphi_n(x))_{n\geq 0}$ und ihre Eigenschaften:

Satz 2.2 (Lineare Unabhängigkeit)
Die Funktionen φ_n eines ONS $(\varphi_n(x))_{n\geq 0}$ sind linear unabhängig.

Beweis:
Wir betrachten zunächst eine Linearkombination der Funktionen φ_n. Sei dazu also

$$\sum_{k=1}^{n} \lambda_k \varphi_k = 0$$

gegeben. Wären die Funktionen φ_n linear abhängig, so gäbe es wenigstens einen Koeffizienten $\lambda_k \neq 0$.
Wende das Skalarprodukt auf diese Gleichung an, dann gilt für alle $m = 1, \ldots, n$:

$$\langle \varphi_m, \sum_{k=1}^{n} \lambda_k \varphi_k \rangle = 0$$

Dann folgt aus der Linearität des Skalarprodukts für die linke Seite der Gleichung

$$\langle \varphi_m, \sum_{k=1}^{n} \lambda_k \varphi_k \rangle = \sum_{k=1}^{n} \lambda_k \langle \varphi_m, \varphi_k \rangle = \lambda_m \langle \varphi_m, \varphi_m \rangle = 0$$

Weil $\langle \varphi_m, \varphi_k \rangle = 0$ (für $m \neq k$) und $\langle \varphi_m, \varphi_m \rangle \neq 0$ ist, folgt $\lambda_m = 0$, und da m beliebig gewählt war, existiert kein $\lambda_n \neq 0$. Also sind $\varphi_1, \ldots, \varphi_n$ linear unabhängig und damit folgt die Behauptung. □

Wir wollen nun noch das Gram-Schmidtsche Orthogonalisierungsverfahren betrachten, welches auch auf unendliche Systeme angewendet werden kann.

Satz 2.3 (Gram-Schmidtsches Orthogonalisierungsverfahren)
Sei $(f_n(x))_{n\geq 0}$ eine linear unabhängige Familie von Funktionen $f_n \in R[s,t]$. Diese lässt sich durch den folgenden iterativen Algorithmus orthogonalisieren:

$$g_0 := f_0$$
$$g_n := f_n - \sum_{k=0}^{n-1} \frac{\langle f_n, g_k \rangle}{\langle g_k, g_k \rangle} g_k.$$

In diesem Fall erhalten wir ein Orthogonalsystem OS $(g_n(x))_{n\geq 0}$. Normiert man die erhaltenen Polynome am Ende noch, so liefert der Algorithmus

$$\begin{aligned} g_0 &:= f_0 \\ g_n &:= f_n - \sum_{k=0}^{n-1} \frac{\langle f_n, g_k \rangle}{\langle g_k, g_k \rangle} g_k, \\ \varphi_n &:= \frac{g_n}{\|g_n\|} \end{aligned}$$

ein Orthonormalsystem $(\varphi_n(x))_{n\geq 0}$ *mit* $\langle \varphi_n, \varphi_n \rangle = 1$.

3 Systeme orthogonaler Polynome

3.1 Orthogonale Polynomsysteme

Wir wollen uns nun auf Skalarprodukte der Form (1) bzw. (2) und OS konzentrieren, die für $n \in \mathbb{N}_{\geq 0}$ aus Polynomen

$$\varphi_n(x) := P_n(x) = k_n x^n + k_n' x^{n-1} + k_n'' x^{n-2} + \dots \quad (k_n \neq 0) \tag{3}$$

bestehen, deren Grad genau n ist. Ein solches Polynomsystem nennen wir orthogonales Polynomsystem (OPS).
Da die Polynome paarweise orthogonal sind, gilt

$$\langle P_n, P_m \rangle = \int_s^t P_n(x) P_m(x) w(x) dx = \begin{cases} h_n \neq 0 & \text{falls } n = m \\ 0 & \text{falls } n \neq m \end{cases}. \tag{4}$$

Wir nennen ein Skalarprodukt *positiv definit*, wenn gilt

$$h_n = \langle P_n, P_n \rangle = \|P_n\|^2 > 0.$$

Sei nun ein gewichtetes Skalarprodukt gemäß (2) mit (4) gegeben. Dann können wir das OPS der Monome $1, x, x^2, x^3, \dots$, welches den Vektorraum der Polynome aufspannt, mit dem Gram-Schmidtschen Verfahren orthogonalisieren. Wir werden jedoch die Normierung $\varphi_n := \frac{g_n}{\|g_n\|}$ durch eine Skalierung gemäß (3) ersetzen und erhalten somit ein bis auf einen Skalierungsfaktor eindeutig bestimmtes OPS.

Beispiel 1 (Legendrepolynome)
Das einfachste gewichtete Skalarprodukt ist gegeben durch $w(x) = 1$*, weiterhin wählen wir das symmetrische Intervall* $[s, t] = [-1, 1]$*. Die orthogonalen Polynome bzgl. dieses Skalarprodukts werden Legendrepolynome genannt. Wir berechnen nun die ersten Legendrepolynome mit dem Gram-Schmidtschen Orthogonalisierungsverfahren mit Hilfe von Mathematica.*

Dazu definieren wir zuerst ein Skalarprodukt. Grenzen und Gewichtsfunktion setzen wir gleich ein.

```
In[1]:= Skalarprodukt[f_, g_] := ∫_{-1}^{1} f * g dx
```

Nun definieren wir zwei Funktionen: GramSchmidtOS, welche ein OS liefert und GramSchmidtONS, welche ein ONS liefert.

```
In[2]:= GramSchmidtOS[f_, anz_] :=
          Module[{fliste, gliste = {}, t},
            fliste = Table[f, {n, 0, anz - 1}];
            AppendTo[gliste, fliste[[1]]];
            For[i = 2, i ≤ anz, i ++,
              t = fliste[[i]] -
                  Sum[Skalarprodukt[fliste[[i]], gliste[[k]]] / Skalarprodukt[gliste[[k]], gliste[[k]]] * gliste[[k]], {k, 1, i-1}];
              AppendTo[gliste, t]];
            gliste//Expand
          ]
In[3]:= GramSchmidtONS[f_, anz_] :=
          Module[{g = GramSchmidtOS[f, anz]},
              Table[g[[i]] / Sqrt[Skalarprodukt[g[[i]], g[[i]]]], {i, 1, anz}]]//
            Expand
```

Wir wollen zuerst zeigen, dass sich der höchste Koeffizient k_n von f_n auf g_n im Falle von GramSchmidtOS überträgt. Dazu betrachten wir

$$g_n = f_n - \sum_{k=0}^{n-1} \frac{\langle f_n, g_k \rangle}{\langle g_k, g_k \rangle} g_k.$$

Da f_n ein Polynom vom Grad n ist überträgt sich der höchste Koeffizient k_n genau dann auf g_n, wenn $\sum_{k=0}^{n-1} \frac{\langle f_n, g_k \rangle}{\langle g_k, g_k \rangle} g_k$ ein Polynom vom Grad $\leq n-1$ ist. Mit $\langle g_k, g_k \rangle = h_k \in \mathbb{R}$ mit $h_k \neq 0$ und $\int_s^t f_n(x) g_k(x) w(x) dx := c \in \mathbb{R}$ erhalten wir $\sum_{k=0}^{n-1} \frac{c}{h_k} g_k$. Da g_k ein Polynom vom Grad $\leq k$ ist und c, h_k Konstanten sind, erhalten wir aufgrund der Summationsgrenze $n-1$ für $\sum_{k=0}^{n-1} \frac{\langle f_n, g_k \rangle}{\langle g_k, g_k \rangle} g_k$ ein Polynom vom Grad $\leq n-1$. Der höchste Koeffizient k_n von f_n bleibt also bei der Subtraktion

$$f_n - \sum_{k=0}^{n-1} \frac{\langle f_n, g_k \rangle}{\langle g_k, g_k \rangle} g_k$$

unverändert und überträgt sich somit auf g_n.

Jetzt lassen wir uns die ersten zehn Legendrepolynome ausgeben, indem wir mit $f = x^n$ starten.

```
In[4]:= OSliste = GramSchmidtOS[x^n, 10]
```

Out[4]= $\left\{1, x, x^2 - \frac{1}{3}, x^3 - \frac{3x}{5}, x^4 - \frac{6x^2}{7} + \frac{3}{35}, x^5 - \frac{10x^3}{9} + \frac{5x}{21},\right.$
$x^6 - \frac{15x^4}{11} + \frac{5x^2}{11} - \frac{5}{231}, x^7 - \frac{21x^5}{13} + \frac{105x^3}{143} - \frac{35x}{429},$
$\left. x^8 - \frac{28x^6}{15} + \frac{14x^4}{13} - \frac{28x^2}{143} + \frac{7}{1287}, x^9 - \frac{36x^7}{17} + \frac{126x^5}{85} - \frac{84x^3}{221} + \frac{63x}{2431}\right\}$

Nun überprüfen wir noch die Orthogonalität

```
In[5]:= Skalarprodukt[OSliste[[4]], OSliste[[5]]]
```

Out[5]= 0

und die Norm

```
In[6]:= Skalarprodukt[OSliste[[4]], OSliste[[4]]]
```

Out[6]= $\frac{8}{175}$

Jetzt lassen wir uns die ersten zehn normierten Legendrepolynome ausgeben.

```
In[7]:= ONSliste = GramSchmidtONS[x^n, 10]
```

Out[7]= $\left\{\frac{1}{\sqrt{2}}, \sqrt{\frac{3}{2}}\, x, \frac{3}{2}\sqrt{\frac{5}{2}}\, x^2 - \frac{\sqrt{\frac{5}{2}}}{2}, \frac{5}{2}\sqrt{\frac{7}{2}}\, x^3 - \frac{3}{2}\sqrt{\frac{7}{2}}\, x,\right.$

$\frac{105x^4}{8\sqrt{2}} - \frac{45x^2}{4\sqrt{2}} + \frac{9}{8\sqrt{2}}, \frac{63}{8}\sqrt{\frac{11}{2}}\, x^5 - \frac{35}{4}\sqrt{\frac{11}{2}}\, x^3 + \frac{15}{8}\sqrt{\frac{11}{2}}\, x,$

$\frac{231}{16}\sqrt{\frac{13}{2}}\, x^6 - \frac{315}{16}\sqrt{\frac{13}{2}}\, x^4 + \frac{105}{16}\sqrt{\frac{13}{2}}\, x^2 - \frac{5\sqrt{\frac{13}{2}}}{16},$

$\frac{429}{16}\sqrt{\frac{15}{2}}\, x^7 - \frac{693}{16}\sqrt{\frac{15}{2}}\, x^5 + \frac{315}{16}\sqrt{\frac{15}{2}}\, x^3 - \frac{35}{16}\sqrt{\frac{15}{2}}\, x,$

$\frac{6435}{128}\sqrt{\frac{17}{2}}\, x^8 - \frac{3003}{32}\sqrt{\frac{17}{2}}\, x^6 + \frac{3465}{64}\sqrt{\frac{17}{2}}\, x^4 - \frac{315}{32}\sqrt{\frac{17}{2}}\, x^2 + \frac{35\sqrt{\frac{17}{2}}}{128},$

$\frac{12155}{128}\sqrt{\frac{19}{2}}\, x^9 - \frac{6435}{32}\sqrt{\frac{19}{2}}\, x^7 +$

$\left.\frac{9009}{64}\sqrt{\frac{19}{2}}\, x^5 - \frac{1155}{32}\sqrt{\frac{19}{2}}\, x^3 + \frac{315}{128}\sqrt{\frac{19}{2}}\, x\right\}$

Auch hier überprüfen wir die Orthogonalität

```
In[8]:= Skalarprodukt[ONSliste[[4]], ONSliste[[5]]]
```

Out[8]= 0

und die Norm

```
In[9]:= Skalarprodukt[ONSliste[[4]], ONSliste[[4]]]
```

Out[9]= 1

Die Polynome, die GramSchmidtONS liefert, sind also wirklich normiert.

Zum Schluss vergleichen wir unsere errechneten Legendrepolynome mit denen die die in Mathematica implementierte Prozedur LegendreP liefert. Dazu lassen wir uns die ersten zehn Legendrepolynome ausgeben

```
In[10]:= LeP = Table[LegendreP[n, x], {n, 0, 9}]
```

Out[10]= $\left\{1, x, \frac{3x^2}{2} - \frac{1}{2}, \frac{5x^3}{2} - \frac{3x}{2}, \frac{35x^4}{8} - \frac{15x^2}{4} + \frac{3}{8}, \frac{63x^5}{8} - \frac{35x^3}{4} + \frac{15x}{8}, \frac{231x^6}{16} - \frac{315x^4}{16} + \frac{105x^2}{16} - \frac{5}{16}, \frac{429x^7}{16} - \frac{693x^5}{16} + \frac{315x^3}{16} - \frac{35x}{16}, \frac{6435x^8}{128} - \frac{3003x^6}{32} + \frac{3465x^4}{64} - \frac{315x^2}{32} + \frac{35}{128}, \frac{12155x^9}{128} - \frac{6435x^7}{32} + \frac{9009x^5}{64} - \frac{1155x^3}{32} + \frac{315x}{128}\right\}$

Als nächstes bilden wir den Quotienten mit den Polynomen, welche wir mit GramSchmidtOS und GramSchmidtONS erhalten haben

```
In[11]:= quot1 = LeP/OSliste //Simplify
```

Out[11]= $\left\{1, 1, \frac{3}{2}, \frac{5}{2}, \frac{35}{8}, \frac{63}{8}, \frac{231}{16}, \frac{429}{16}, \frac{6435}{128}, \frac{12155}{128}\right\}$

```
In[12]:= quot2 = LeP/ONSliste //Simplify
```

Out[12]= $\left\{\sqrt{2}, \sqrt{\frac{2}{3}}, \sqrt{\frac{2}{5}}, \sqrt{\frac{2}{7}}, \frac{\sqrt{2}}{3}, \sqrt{\frac{2}{11}}, \sqrt{\frac{2}{13}}, \sqrt{\frac{2}{15}}, \sqrt{\frac{2}{17}}, \sqrt{\frac{2}{19}}\right\}$

Multiplizieren wir nun die Polynome aus GramSchmidtOS und GramSchmidtONS mit den erhaltenen konstanten Faktoren, so sehen wir, dass die Polynomsysteme bis auf diese konstanten Faktoren gleich sind.

```
In[13]:= Expand[quot1 * OSliste] == LeP
Out[13]= True

In[14]:= Expand[quot2 * ONSliste] == LeP
Out[14]= True
```

3.2 Eigenschaften von OPS

Als Nächstes wollen wir Eigenschaften betrachten, die alle OPS besitzen. Dazu beweisen wir

Satz 3.1 (Linearkombination)
Sei $(P_n(x))_{n\in\mathbb{N}_{\geq 0}}$ ein OPS bzgl. eines gewichteten Skalarprodukts und sei $q(x)$ ein Polynom vom Grad n. Dann ist $q(x)$ als Linearkombination

$$q(x) = \sum_{k=0}^{n} c_k P_k(x)$$

des gegebenen OPS darstellbar, wobei die Koeffizienten gegeben sind durch

$$c_j = \frac{1}{h_j}\langle q, P_j\rangle = \frac{1}{h_j}\int_s^t w(x)q(x)P_j(x)dx.$$

Beweis:
Wir rechnen nach:

$$\left\langle q, P_j \right\rangle = \left\langle \sum_{k=0}^{n} c_k P_k, P_j \right\rangle = \sum_{k=0}^{n} c_k \left\langle P_k, P_j \right\rangle = c_j h_j,$$

$$\Rightarrow c_j = \frac{1}{h_j} \left\langle q, P_j \right\rangle \text{und somit die Behauptung}$$

□

Eine erste Folgerung daraus ist

Satz 3.2 (Strukturbeziehung)
Sei $(P_n(x))_{n\in\mathbb{N}_{\geq 0}}$ ein OPS bzgl. eines gewichteten Skalarprodukts, dann ist $P_n(x)$ zu jedem Polynom kleineren Grades orthogonal.

Beweis:
Aus Satz 3.1 wissen wir, dass sich jedes Polynom $q(x)$ vom Grad kleiner n linear darstellen lässt durch

$$q(x) = c_0 P_0(x) + c_1 P_1(x) + \ldots + c_{n-1} P_{n-1}(x)$$

mit konstanten Koeffizienten c_k. Daraus folgt nun

$$\left\langle q, P_n \right\rangle = \sum_{k=0}^{n-1} c_k \left\langle P_k, P_n \right\rangle = 0,$$

da alle P_k zueinander orthogonal sind.

□

Wir können nun weiterhin folgern

Satz 3.3 (Nullstellensatz)
Die Nullstellen jedes OPS bzgl. eines gewichteten Skalarprodukts sind alle reell, einfach und liegen im Inneren des Intervalls $[s, t]$.

Beweis:
Seien $x_1, \ldots, x_m$ $(0 \leq m \leq n)$ die Nullstellen von $P_n(x)$, welche im Inneren von $[s, t]$ liegen und einfach sind. Dann wechselt $P_n(x)$ genau an den Stellen $x_1, \ldots, x_m$ das Vorzeichen. Wir definieren das Polynom

$$q_n(x) := \begin{cases} 1 & \text{falls } m = 0 \\ (x - x_1)...(x - x_m) & \text{falls } m > 0, \end{cases}$$

das an den gleichen Stellen wie $P_n(x)$ das Vorzeichen wechselt. Folglich hat das Produkt $w(x)P_n(x)q_n(x)$ in ganz $[s, t]$ ein einheitliches Vorzeichen. Es gilt also

$$\int_s^t w(x)P_n(x)q_n(x)dx \neq 0.$$

Dies kann aber nur dann der Fall sein, wenn $m = n$ gilt, denn für $m < n$ ist gemäß Satz 3.2 $P_n(x)$ zu jedem Polynom kleineren Grades orthogonal. Weiterhin hat jedes Polynom vom Grad n höchstens n Nullstellen, also muss $n = m$ gelten, also sind die Nullstellen einfach. □

Wir können nun den ersten Trennungssatz formulieren. Allgemein ist es anzumerken, dass es viele Trennungssätze gibt. Insbesondere wenn man spezielle Polynomsysteme wie z.B. Legendrepolynome, Chebychevpolynome usw. einzeln betrachtet, kann man noch wesentlich weitreichendere Aussagen über Nullstellen treffen. Wir wollen uns in dieser Arbeit aber auf den ersten und zweiten Trennungssatz beschränken, welche sehr fundamentale Aussagen über die Lage von Nullstellen, gültig für alle Polynomsysteme, enthalten.

Satz 3.4 (Trennungssatz 1)
Zwischen je zwei aufeinander folgenden Nullstellen eines orthogonalen Polynoms im Intervall $[s, t]$ *liegt genau eine Nullstelle seiner Ableitung.*

Beweis:
Gemäß des Satzes von Rolle liegt zwischen zwei Nullstellen einer differenzierbaren Funktion mindestens eine Nullstelle seiner Ableitung. Da $P_n'(x)$ ein Polynom vom Grad $n-1$ ist, liegen seine Nullstellen genau zwischen den Nullstellen von $P_n(x)$ und sind wegen Satz 3.3 einfach. □

Als ersten wichtigen Satz erhalten wir die folgende Rekursionsgleichung:

Satz 3.5 (Rekursionsgleichung)
Jedes OPS bzgl. eines gewichteten Skalarprodukts erfüllt eine Rekursionsgleichung der Form

$$P_{n+1}(x) = (A_n x + B_n)P_n(x) - C_n P_{n-1}(x) \qquad (n \geq 0) \tag{5}$$

mit Konstanten A_n, B_n, C_n, *die den Gleichungen*

$$A_n = \frac{k_{n+1}}{k_n} \tag{6}$$

$$B_n = A_n\left(\frac{k'_{n+1}}{k_{n+1}} - \frac{k'_n}{k_n}\right) \tag{7}$$

$$C_n = \frac{A_n h_n}{A_{n-1} h_{n-1}} \tag{8}$$

genügen.

Beweis:
Sei $n \geq 1$ und $A_n = \frac{k_{n+1}}{k_n}$. Das Polynom

$$P_{n+1}(x) - A_n x P_n(x)$$

hat einen Grad $\leq n$ und man kann es somit als Linearkombination

$$P_{n+1}(x) - A_n x P_n(x) = c_0 P_0(x) + \ldots + c_n P_n(x).$$

darstellen. Sei $k = 0, \dots, n-2$, dann folgt mit Satz 3.2, der Orthogonalität von $P_k(x)$ und der speziellen Eigenschaft $\langle f, xg\rangle = \langle xf, g\rangle$ des Skalarprodukts

$$\begin{aligned} h_k c_k &= \left\langle c_0 P_0(x) + \dots + c_n P_n(x), P_k \right\rangle = \left\langle P_{n+1}(x) - A_n x P_n(x), P_k \right\rangle \\ &= \left\langle P_{n+1}, P_k \right\rangle - A_n \left\langle x P_n, P_k \right\rangle = -A_n \left\langle P_n, x P_k \right\rangle = 0. \end{aligned}$$

Da $h_k \neq 0$ ist folgt $c_k = 0$. Wir erhalten also eine Darstellung

$$P_{n+1}(x) - A_n x P_n(x) = c_{n-1} P_{n-1}(x) + c_n P_n(x).$$

Mit $c_n = B_n$ und $c_{n-1} = C_n$ erhalten wir dann nach Umformung (5). Da alle Polynome in (5) eine Darstellung gemäß (3) haben, können wir einen Koeffizientenvergleich von x^n durchführen, welcher

$$k'_{n+1} = A_n k'_n + B_n k_n$$

liefert. Durch Umformung erhalten wir dann

$$B_n = \frac{k'_{n+1} - A_n k'_n}{k_n} = A_n \left(\frac{k'_{n+1}}{k_{n+1}} - \frac{k'_n}{k_n}\right).$$

Formen wir nun (5) um, so erhalten wir

$$C_n P_{n-1}(x) = (A_n x + B_n) P_n(x) - P_{n+1}(x)$$

und somit auch

$$\begin{aligned} C_n h_{n-1} &= C_n \left\langle P_{n-1}, P_{n-1} \right\rangle = A_n \left\langle P_{n-1}, x P_n \right\rangle + B_n \left\langle P_n, P_{n-1} \right\rangle - \left\langle P_{n+1}, P_{n-1} \right\rangle \\ &= A_n \left\langle P_n, k_{n-1} x^n \right\rangle = A_n \frac{k_{n-1}}{k_n} \left\langle P_n, P_n \right\rangle = A_n \frac{k_{n-1}}{k_n} h_n. \end{aligned}$$

Auflösen nach C_n liefert dann (8). Definieren wir zusätzlich noch $P_{-1}(x) := 1$, so bleibt unsere Rekursionsgleichung auch für $n = 0$ gültig. □

Es gilt ebenfalls die Umkehrung dieses Satzes

Satz 3.6 (Satz von Farvard)
Gilt für ein Polynomsystem $P_n(x)$ eine Dreitermrekursion der Form

$$P_{n+1}(x) = (A_n x + B_n) P_n(x) - C_n P_{n-1}(x) \quad (n \geq 0), P_{-1}(x) = 0, P_0(x) = 1,$$

dann ist $P_n(x)$ orthogonal. Ferner ist das zugehörige Skalarprodukt ist genau dann positiv definit, wenn $C_n A_n A_{n-1} > 0$ für $n \geq 1$ gilt.

Beweis:
Sei o.B.d.A $Q_n(x) := \dfrac{P_n(x)}{k_n}$. Dann ist $Q_n(x)$ ein Polynomsystem bei dem für den höchsten Koeffizienten $k_n = 1$ gilt. Als Folgerung erhalten wir dann

$$A_n = 1, \quad B_n = -(k'_n - k'_{n+1}) := c_n, \quad C_n = \frac{h_n}{h_{n-1}} := \lambda_n.$$

Dies liefert uns die Gleichung

$$Q_n(x) = (x - c_n)Q_{n-1}(x) - \lambda_n Q_{n-2}(x). \tag{9}$$

Weiterhin erklären wir die Gewichtsfunktion $w(x)$ genau so, dass gilt

$$\int_s^t Q_0(x)Q_n(x)w(x)dx = \int_s^t Q_n(x)w(x)dx = 0 \tag{10}$$

für $n = 1, 2, 3, \ldots$ und

$$\int_s^t Q_0(x)Q_0(x)w(x)dx = \int_s^t 1w(x)dx = \mu_0 = \lambda_1.$$

Integrieren wir nun (9) nach Multiplikation mit $w(x)$, so erhalten wir für $n = 1$

$$\int_s^t Q_1(x)w(x)dx = \int_s^t xw(x)dx - c_1 \int_s^t Q_0(x)w(x)dx - \lambda_1 \int_s^t Q_{-1}w(x)dx = 0$$

$$\Rightarrow \int_s^t Q_1(x)w(x)dx = \mu_1 - c_1\mu_0 = 0.$$

Für $n = 2$ erhalten wir

$$\begin{aligned}
\int_s^t Q_2(x)w(x)dx &= \int_s^t xQ_1(x)w(x)dx - c_2 \int_s^t Q_1(x)w(x)dx - \lambda_2 \int_s^t Q_0(x)w(x)dx \\
&= \int_s^t xQ_1(x)w(x)dx - c_2(\mu_1 - c_1\mu_0) - \lambda_2\mu_0 \\
&= \int_s^t x\left(xQ_0(x) - c_1Q_0(x) - \lambda_1 Q_{-1}(x)\right) - c_2(\mu_1 - c_1\mu_0) - \lambda_2\mu_0 \\
&= \int_s^t x^2 w(x)dx - c_1 \int_s^t xw(x)dx - c_2(\mu_1 - c_1\mu_0) - \lambda_2\mu_0 \\
&= \mu_2 - (c_1 + c_2)\mu_1 + (\lambda_2 - c_1c_2)\mu_0 = 0
\end{aligned}$$

Auf analoge Weise können wir auch die folgenden Darstellungen für $n = 3, 4, 5, \ldots$ herleiten.
Umformung von (9) und Shift von n liefert

$$xQ_n(x) = Q_{n+1}(x) + c_{n+1}Q_n(x) + \lambda_{n+1}Q_{n-1} \tag{11}$$

Multiplikation von (11) mit $w(x)$ und Integration liefert dann zusammen mit (10)

$$\int_s^t xQ_n(x)w(x)dx = 0, \quad n \geq 2.$$

Multiplikation von (11) mit $xw(x)$ und Integration liefert zusammen mit (10)

$$\int_s^t x^2 Q_n(x)w(x)dx = 0, \quad n \geq 3.$$

Führen wir dies fort, so erhalten wir

$$\int_s^t x^k Q_n(x)w(x)dx = 0, \quad 0 \leq k < n.$$

Daraus folgt die Orthogonalität von $Q_n(x)$ bzgl. der Gewichtsfunktion $w(x)$.

Wir müssen nun noch zeigen, dass gilt

$$C_n A_n A_{n-1} > 0 \Leftrightarrow \langle P_n(x), P_n(x) \rangle = \int_s^t P_n^2(x)w(x)dx > 0.$$

Setzen wir (6) und (8) in $C_n A_n A_{n-1}$ ein, so erhalten wir

$$C_n A_n A_{n-1} = \frac{A_n h_n}{A_{n-1} h_{n-1}} \frac{k_{n+1}}{k_n} \frac{k_n}{k_{n-1}} = \frac{k_{n+1}^2 h_n}{k_n^2 h_{n-1}}$$

Da k_n und k_{n+1} als führende Koeffizienten von $P_n(x)$ bzw. $P_{n+1}(x)$ ungleich Null sind gilt $k_n^2 > 0$ und $k_{n+1}^2 > 0$.
„$\Leftarrow$“ Sei $P_n(x)$ positiv definit, dann gilt $h_n > 0$ und $h_{n-1} > 0$ für $n \geq 1$. Mit $C_n A_n A_{n-1} = \frac{k_{n+1}^2 h_n}{k_n^2 h_{n-1}}$ folgt dann $C_n A_n A_{n-1} > 0$ für $n \geq 1$.
„$\Rightarrow$“ Wir zeigen diese Aussage induktiv. Wegen $h_0 = 1$ erhalten wir für $n = 1$

$$C_1 A_1 A_0 = \frac{k_2^2 h_1}{k_1^2}.$$

Also kann $C_1 A_1 A_0 > 0$ nur gelten, wenn $h_1 > 0$ erfüllt ist.
Für $n = 2$ gilt

$$C_2 A_2 A_1 = \frac{k_3^2 h_2}{k_2^2 h_1}.$$

Wenn $h_1 > 0$, so folgt, dass $C_2 A_2 A_1 > 0$ nur dann erfüllt ist, wenn $h_2 > 0$ gilt.
Für $n = 3$ gilt

$$C_3 A_3 A_2 = \frac{k_4^2 h_3}{k_3^2 h_2}.$$

Wenn $h_2 > 0$, so folgt, dass $C_3 A_3 A_2 > 0$ nur dann erfüllt ist, wenn $h_3 > 0$ gilt. Führen wir dies fort, so erhalten wir für n

$$C_n A_n A_{n-1} = \frac{k_{n+1}^2 h_n}{k_n^2 h_{n-1}}.$$

Wenn $h_{n-1} > 0$, so folgt, dass $C_n A_n A_{n-1} > 0$ nur dann erfüllt ist, wenn $h_n > 0$ gilt. Wir sehen das $h_n > 0$, also die positive Definitheit von $P_n(x)$, genau dann folgt, wenn $C_n A_n A_{n-1} > 0$ für $n \geq 1$ erfüllt ist.

□

Wir wollen nun eine weitere wichtige Formel für OPS einführen und beweisen.

Satz 3.7 (Summationsformel von Christoffel-Darboux)
Jedes OPS bzgl. eines gewichteten Skalarprodukts erfüllt die Summationsformel von Christoffel-Darboux

$$\sum_{k=0}^{n} \frac{1}{h_k} P_k(x) P_k(y) = \frac{1}{h_n} \frac{k_n}{k_{n+1}} \frac{P_{n+1}(x)P_n(y) - P_n(x)P_{n+1}(y)}{x-y}. \tag{12}$$

Hieraus folgt im Grenzfall $y \to x$

$$\sum_{k=0}^{n} \frac{1}{h_k} P_k^2(x) = \frac{1}{h_n} \frac{k_n}{k_{n+1}} (P'_{n+1}(x)P_n(x) - P'_n(x)P_{n+1}(x)). \tag{13}$$

Beweis:
Wir multiplizieren (5) bzgl. x mit $P_n(y)$ sowie (5) bzgl y mit $P_n(x)$ und erhalten:

$$\begin{aligned}
\text{I. } P_{k+1}(x)P_k(y) &= (A_k x + B_k)P_k(x)P_k(y) - C_k P_{k-1}(x)P_k(y) \\
\text{II. } P_k(x)P_{k+1}(y) &= (A_k y + B_k)P_k(x)P_k(y) - C_k P_k(x)P_{k+1}(y)
\end{aligned}$$

Wir bilden nun I – II und erhalten mit Umformungen:

$$\begin{aligned}
P_{k+1}(x)P_k(y) - P_k(x)P_{k+1}(y) &= (A_k x + B_k)P_k(x)P_k(y) - C_k P_{k-1}(x)P_k(y) \\
&\quad -(A_k y + B_k)P_k(x)P_k(y) + C_k P_k(x)P_{k+1}(y) \\
&= A_k x P_k(x)P_k(y) + B_k P_k(x)P_k(y) - C_k P_{k-1}(x)P_k(y) \\
&\quad -A_k y P_k(x)P_k(y) - B_k P_k(x)P_k(y) - C_k P_{k-1}(y)P_k(x) \\
&= (x-y)(A_k P_k(x)P_k(y)) + C_k(P_k(x)P_{k-1}(y) - P_{k-1}(x)P_k(y))
\end{aligned}$$

$$\begin{aligned}
\Rightarrow P_{k+1}(x)P_k(y) - P_k(x)P_{k+1}(y) &= (x-y)(A_k P_k(x)P_k(y)) + C_k(P_k(x)P_{k-1}(y) - P_{k-1}(x)P_k(y)) \\
\Leftrightarrow -(x-y)(A_k P_k(x)P_k(y)) &= -(P_{k+1}(x)P_k(y) - P_k(x)P_{k+1}(y)) \\
&\quad +C_k(P_k(x)P_{k-1}(y) - P_{k-1}(x)P_k(y))
\end{aligned}$$

Einsetzen von (6) und (8) liefert:

$$\begin{aligned}
\frac{(x-y)}{h_k} P_k(x)P_k(y) &= \frac{1}{A_k h_k}(P_{k+1}(x)P_k(y) - P_k(x)P_{k+1}(y)) \\
&\quad - \frac{1}{A_{k-1}h_{k-1}}(P_k(x)P_{k-1}(y) - P_{k-1}(x)P_k(y))
\end{aligned}$$

Summieren wir nun von $k = 0, \ldots, n$, so erhalten wir:

$$\begin{aligned}
(x-y)\sum_{k=0}^{n}\frac{1}{h_k}P_k(x)P_k(y) &= \sum_{k=0}^{n}\Big(\frac{1}{A_kh_k}(P_{k+1}(x)P_k(y)-P_k(x)P_{k+1}(y)) \\
&\quad -\frac{1}{A_{k-1}h_{k-1}}(P_k(x)P_{k-1}(y)-P_{k-1}(x)P_k(y))\Big) \\
&\overset{\text{Teleskopsumme}}{=} -\frac{1}{h_{-1}A_{-1}}(P_0(x)P_{-1}(y)-P_{-1}(x)P_0(y)) \\
&\quad +\frac{1}{A_nh_n}(P_{n+1}(x)P_n(y)-P_n(x)P_{n+1}(y))
\end{aligned}$$

Mit $-\frac{1}{h_{-1}A_{-1}}(P_0(x)P_{-1}(y) - P_{-1}(x)P_0(y)) = 0$ erhalten wir dann (12). Für den Grenzfall substituieren wir $y = x - z$ und setzen dies in (12) ein. Damit erhalten wir

$$\sum_{k=0}^{n}\frac{1}{h_k}P_k(x)P_k(x-z) = \frac{1}{h_n}\frac{k_n}{k_{n+1}}\frac{(P_{n+1}(x)P_n(x-z)-P_n(x)P_{n+1}(x-z))}{z}.$$

Führen wir nun auf beiden Seiten den Grenzübergang $z \to 0$ durch, was auf der rechten Seite nach der Regel von L'Hospital zu erfolgen hat, so erhalten wir

$$\sum_{k=0}^{n}\frac{1}{h_k}P_k^2(x) = \frac{1}{h_n}\frac{k_n}{k_{n+1}}\left(P'_{n+1}(x)P_n(x)-P'_n(x)P_{n+1}(x)\right).$$

□

Mithilfe der Summationsformel von Christoffel-Darboux können wir nun den zweiten Trennungssatz beweisen.

Satz 3.8 (Trennungssatz 2)
Zwischen je zwei aufeinander folgenden Nullstellen eines orthogonalen Polynoms $P_n(x)$ im Intervall $[s, t]$ liegt genau eine Nullstelle von $P_{n+1}(x)$ und umgekehrt.

Beweis:
Sei o.B.d.A. $k_n = 1$, dann gilt für jede Nullstelle x_k von $P_n(x)$ wegen (13) die Ungleichung

$$P'_n(x_k)P_{n+1}(x_k) < 0.$$

Seien weiterhin x_k und x_{k+1} zwei aufeinander folgende Nullstellen von $P_n(x)$, dann ist das Vorzeichen von $P'_n(x)$ und $P'_n(x_{k+1})$ verschieden und aufgrund von $P'_n(x_k)P_{n+1}(x_k) < 0$ auch von $P'_{n+1}(x_k)$ und $P'_{n+1}(x_{k+1})$. Folglich liegt zwischen x_k und x_{k+1} eine Nullstelle von $P_{n+1}(x)$, welche wegen Satz 3.3 einfach ist. Die Rückrichtung kann analog gezeigt werden. □

4 Klassische orthogonale Polynome

4.1 Einleitung

Wir wollen nun Polynomsysteme der Form (3) mit der Eigenschaft (4), welche die Eigenwertdifferentialgleichung

$$\sigma(x)y''(x) + \tau(x)y'(x) + \lambda_n y(x) = 0, \tag{14}$$

erfüllen, betrachten. Seien dazu $\sigma(x)$ und $\tau(x)$ zunächst beliebige stetige Funktionen und λ_n eine beliebige Folge reeller Zahlen.
Wir setzen nun $y(x) = P_1(x)$ in die Eigenwertdifferentialgleichung ein und erhalten

$$\tau(x)P_1'(x) + \lambda_1 P_1(x) = 0.$$

Da $P_1(x)$ ein lineares Polynom ist, muss auch $\tau(x) = dx + e$ ein lineares Polynom sein.
Wir benutzen weiterhin den Ansatz $y(x) = P_2(x)$ und erhalten

$$\sigma(x)P_2''(x) + \tau(x)P_2'(x) + \lambda_2 P_2(x) = 0.$$

Da $P_2(x)$ ein quadratisches Polynom ist, muss auch $\sigma(x) = ax^2 + bx + c$ ein quadratisches Polynom sein. Die Eigenwertdifferentialgleichung ist also in jedem Fall eine Polynomidentität. Koeffizientenvergleich des höchsten Koeffizienten liefert uns

$$\lambda_n = -(an(n-1) + dn).^1$$

Die Polynomlösungen der Eigenwertdifferentialgleichung können nun vollständig klassifiziert werden. Aufgrund der Invarianz der Eigenwertdifferentialgleichung unter einer linearen Abbildung $x \mapsto fx + g$ ist das Polynomsystem $Q_n(x) := P_n(fx + g)$ eine Lösung von (14) genau dann, wenn $P_n(x)$ eine Lösung ist. Die Invarianz ermöglicht es uns nun, die Nullstellen von $\sigma(x)$ beliebig zu verschieben. Wir definieren also

- Ist $\sigma(x)$ konstant, so setzen wir $\sigma(x) = 1$
- Hat $\sigma(x)$ eine einfache Nullstelle, so setzen wir $\sigma(x) = x$
- Hat $\sigma(x)$ eine doppelte Nullstelle, so setzen wir $\sigma(x) = x^2$
- Hat $\sigma(x)$ zwei verschiedene reelle Nullstellen, so seien die Nullstellen bei $1, -1$ und wir setzen $\sigma(x) = x^2 - 1$

und erhalten folgende Fallunterscheidung:

[1] Beweis folgt später

	$\sigma(x)$	$\tau(x)$	$P_n(x)$	Familie
1	0	x	x^n	
2	1	$-2x$	$H_n(x)$	Hermitepolynome
3	x	$-x+\alpha+1$	$L_n^{(\alpha)}(x)$	Laguerrepolynome
4	x^2	$(\alpha+2)x+2$	$B_n^{(\alpha)}(x)$	Besselpolynome
5	x^2-1	$(\alpha+\beta+2)x+\alpha-\beta$	$P_n^{(\alpha,\beta)}(x)$	Jacobipolynome

Tabelle 1: Klassifizierung der klassischen orthogonalen Polynome

Diese Polynome werden die klassischen orthogonalen Polynome genannt.

4.2 Orthogonalität

Wir wollen nun beweisen, dass alle diese Polynomsysteme, bis auf die Besselpolynome, in einem reellen Intervall orthogonal sind. Dazu müssen wir zuerst die zugehörige Gewichtsfunktion suchen. Sei also $w(x)$ zunächst eine beliebige stetig differenzierbare Funktion auf dem Intervall $[s,t]$. Wir betrachten dazu

$$(w(x)\sigma(x)y'(x))' = w(x)\sigma(x)y''(x) + (w(x)\sigma(x))'y'(x) \tag{15}$$

sowie mit (14)

$$\begin{aligned} -\lambda_n y(x) &= \tau(x)y'(x) + \sigma(x)y''(x) \\ \Leftrightarrow -w(x)\lambda_n y(x) &= w(x)\sigma(x)y''(x) + w(x)\tau(x)y'(x). \end{aligned} \tag{16}$$

Genügt die Funktion $w(x)$ zusätzlich der Pearsonschen Differentialgleichung

$$(\sigma(x)w(x))' = \tau(x)w(x),$$

so erhalten wir mit (15) und (16)

$$(w(x)\sigma(x)y'(x))' + w(x)\lambda_n y(x) = 0. \tag{17}$$

Durch Lösen der Pearsonschen Differentialgleichung erhalten wir eine bis auf einen konstanten Faktor eindeutig bestimmte Darstellung für $w(x)$, sofern $\sigma(x)$ im Intervall $[s,t]$ nicht verschwindet, denn sonst würde der Fall einer Null im Nenner doppelt auftreten:

$$w(x) = \frac{C}{\sigma(x)} e^{\int \frac{\tau(x)}{\sigma(x)} dx}$$

Somit erhalten wir für die Hermite-, Laguerre-, Bessel- und Jacobipolynome folgende Tabelle, wobei C so gewählt ist, dass die Beziehung $w(x) > 0$ in dem zugehörigen Intervall $[s,t]$ gilt.

$\sigma(x)$	$P_n(x)$	Familie	$w(x)$	Intervall	Bedingungen
1	$H_n(x)$	Hermitepolynome	e^{-x^2}	$(-\infty,\infty)$	
x	$L_n^{(\alpha)}(x)$	Laguerrepolynome	$x^\alpha e^{-x}$	$(0,\infty)$	$\alpha > -1$
x^2	$B_n^{(\alpha)}$	Besselpolynome	$x^\alpha e^{-\frac{2}{x}}$	-	
x^2-1	$P_n^{(\alpha,\beta)}$	Jacobipolynome	$(1-x)^\alpha(1+x)^\beta$	$(-1,1)$	$\alpha > -1, \beta > -1$

Tabelle 2: Gewichtsfunktionen der klassischen orthogonalen Polynome

Umformung der Pearsonschen Differentialgleichung liefert

$$\begin{aligned}(\sigma(x)w(x))' &= \tau(x)w(x)\\ \Leftrightarrow \sigma'(x)w(x)+\sigma(x)w'(x) &= \tau(x)w(x)\\ \Leftrightarrow \frac{w'(x)}{w(x)} &= \frac{\tau(x)-\sigma'(x)}{\sigma(x)}.\end{aligned}$$

Mit Partialbruchzerlegung erhalten wir dann für geeignete $\alpha,\beta \in \mathbb{R}$

$$\frac{w'(x)}{w(x)} = \frac{\tau(x)-\sigma'(x)}{\sigma(x)} = \frac{\alpha}{x-x_1} + \frac{\beta}{x-x_2},$$

wobei $x_1, x_2 \in \mathbb{R}$ die Nullstellen von $\sigma(x)$ und somit genau die Intervallgrenzen darstellen. Integration liefert uns dann

$$w(x) = C(x-x_1)^\alpha (x-x_2)^\beta.$$

Nach Einsetzen der Intervallgrenzen erhalten wir für $C = (-1)^\beta$ die Gewichtsfunktion der Jacobipolynome.
Wir müssen nun zeigen, dass die orthogonalen Polynome, bis auf die Besselpolynome, gemäß den Spezifikationen aus Tabelle 2 orthogonal sind.

Satz 4.1 (Klassische OPS)
Die Lösungsfamilien $P_n(x)$ der Eigenwertdifferentialgleichung mit $\sigma(x) = ax^2 + bx + c$ ($abc \neq 0$) und $\tau(x) = dx + e, d \neq 0$) stellen OPS dar, welche die klassischen OPS genannt werden. Die jeweilige Gewichtsfunktion ist durch die Pearsonsche Differentialgleichung gegeben. Es ergibt sich modulo linearer Transformationen eine Klassifizierung der klassischen OPS gemäß Tabelle 1.

Beweis:
Wir setzen $y(x) = P_n(x)$ und erhalten mit (17):

$$\left(w(x)\sigma(x)P_n'(x)\right)' + \lambda_n w(x)P_n(x) = 0.$$

Multiplizieren wir die zu n gehörige obige Gleichung mit $P_m(x)$ und die zu m gehörige obige Gleichung mit $P_n(x)$, so erhalten wir durch Gleichsetzen dieser beiden Gleichungen:

$$\begin{aligned}\left(w(x)\sigma(x)P_n'(x)\right)' P_m(x) + \lambda_n w(x)P_n(x)P_m(x) &= \left(w(x)\sigma(x)P_m'(x)\right)' P_n(x)\\ &\quad +\lambda_m w(x)P_m(x)P_n(x)\\ \Leftrightarrow (\lambda_m-\lambda_n)P_n(x)P_m(x)w(x) &= \left(w(x)\sigma(x)P_n'(x)\right)' P_m(x)\\ &\quad -\left(w(x)\sigma(x)P_m'(x)\right)' P_n(x)\end{aligned}$$

Integration über das Intervall $[s,t]$ liefert:

$$\begin{aligned}(\lambda_m-\lambda_n)\int_s^t P_n(x)P_m(x)w(x)dx &= \int_s^t \left(w(x)\sigma(x)P_n'(x)\right)' P_m(x)dx - \\ &\quad \int_s^t \left(w(x)\sigma(x)P_m'(x)\right)' P_n(x)dx\end{aligned}$$

Mit partieller Integration der rechten Seite erhalten wir dann:

$$
\begin{aligned}
&= \left(\Big(w(x)\sigma(x)P_n'(x)P_m(x)\Big)_s^t - \int_s^t w(x)\sigma(x)P_n'(x)P_m'(x)dx \right) \\
&- \left(\Big(w(x)\sigma(x)P_m'(x)P_n(x)\Big)_s^t - \int_s^t w(x)\sigma(x)P_m'(x)P_n'(x)dx \right) \\
&= \Big(w(x)\sigma(x)P_n'(x)P_m(x)\Big)_s^t - \Big(w(x)\sigma(x)P_m'(x)P_n(x)\Big)_s^t
\end{aligned}
$$

Stellen wir nun sicher, dass die Randbedingungen

$$\lim_{x\to s} x^n\sigma(x)w(x) = \lim_{x\to t} x^n\sigma(x)w(x) = 0 \quad n \in \mathbb{N}_{\geq 0}$$

für alle Familien orthogonaler Polynome gelten, so ist deren Orthogonalität gezeigt. Dazu verwenden wir die in Tabelle 2 gegebenen Intervallgrenzen, Gewichtsfunktionen und Zusatzbedingungen.

Für die Hermitepolynome gilt

$$\lim_{x\to -\infty} x^n e^{-x^2} = \lim_{x\to\infty} x^n e^{-x^2} = 0.$$

Die Grenzbedingung ist erfüllt, da e^{-x^2} schneller gegen Null konvergiert als x^n gegen $-\infty$ bzw. ∞.

Bei den Laguerrepolynomen müssen wir die Zusatzbedingung $\alpha > -1$ mit betrachten. Für $s = 0$ und $\alpha > -1$ erhalten wir

$$\lim_{x\to 0} x^n x x^\alpha e^{-x} = 0.$$

Für $s = 0$ und $\alpha \leq -1$ erhalten wir

$$\lim_{x\to 0} x^n x \frac{1}{x^\alpha} e^{-x} = \lim_{x\to 0} \frac{x^{n+1}}{x^\alpha} e^{-x}.$$

Betrachten wir nun den Term $\frac{x^{n+1}}{x^\alpha}$, so erhalten wir im Fall $n + 1 < (-1)\alpha$ den Term $\frac{1}{x^{\alpha-n-1}}$, dessen Grenzwert ∞ ist. Die Randbedingung ist also im Falle $\alpha \leq -1$ nicht mehr sichergestellt.

Für $t = \infty$ und erhalten wir:

$$\lim_{x\to\infty} x^n x x^\alpha e^{-x} = \lim_{x\to\infty} x^{n+1+\alpha} e^{-x} = 0,$$

da e^{-x} schneller gegen Null konvergiert als $x^{n+1+\alpha}$ gegen ∞. Im Falle $n+1+\alpha < 0$ würde $x^{n+1+\alpha}$ dann auch gegen Null konvergieren.

Die Integrale der Besselpolynome existieren wegen

$$\lim_{x\to-\infty} x^{\alpha+n+2}e^{-\frac{2}{x}} = \begin{cases} \infty, & \text{wenn } \alpha+n+2 \text{ gerade} \\ -\infty, & \text{wenn } \alpha+n+2 \text{ ungerade} \end{cases}$$

sowie

$$\lim_{x\to\infty} x^{\alpha+n+2}e^{-\frac{2}{x}} = \infty$$

in keinem reellen Intervall. Die Besselpolynome sind also in keinem reellen Intervall orthogonal.

Für die Jacobipolynome gilt für $s = -1$ und $\alpha > -1, \beta > -1$

$$\lim_{x\to-1} x^n(x^2-1)(1-x)^\alpha(1+x)^\beta = 0.$$

Im Falle $s = -1$ und $\alpha < -1, \beta < -1$ erhalten wir jedoch

$$\lim_{x\to-1} \frac{x^n(x^2-1)}{(1-x)^\alpha(1+x)^\beta} = \lim_{x\to-1} \frac{-x^n}{(1-x)^{\alpha-1}(1+x)^{\beta-1}} = \begin{cases} -\infty, & \text{wenn } n \text{ gerade} \\ \infty, & \text{wenn } n \text{ ungerade,} \end{cases}$$

und im Falle $s = -1$ und $\alpha = -1, \beta = -1$

$$\lim_{x\to-1} \frac{x^n(x^2-1)}{(1-x)^\alpha(1+x)^\beta} = \lim_{x\to-1} -x^n = \begin{cases} -1, & \text{wenn } n \text{ gerade} \\ 1, & \text{wenn } n \text{ ungerade.} \end{cases}$$

Die Randbedingung ist also für $\alpha \leq -1, \beta \leq -1$ nicht erfüllt. Für $t = 1$ und $\alpha > -1, \beta > -1$ erhalten wir

$$\lim_{x\to 1} x^n(x^2-1)(1-x)^\alpha(1+x)^\beta = 0.$$

Im Falle $t = 1$ und $\alpha < -1, \beta < -1$ erhalten wir jedoch

$$\lim_{x\to 1} \frac{x^n(x^2-1)}{(1-x)^\alpha(1+x)^\beta} = \lim_{x\to 1} \frac{-x^n}{(1-x)^{\alpha-1}(1+x)^{\beta-1}} = -\infty,$$

und im Falle $t = 1$ und $\alpha = -1, \beta = -1$

$$\lim_{x\to 1} \frac{x^n(x^2-1)}{(1-x)^\alpha(1+x)^\beta} = \lim_{x\to 1} -x^n = -1.$$

Die Randbedingung ist also für $\alpha \leq -1, \beta \leq -1$ nicht erfüllt. Insgesamt sind die Randbedingungen also nur erfüllt, wenn $\alpha > -1$ und $\beta > -1$ gilt. □

Als Nächstes wollen wir nun die Umkehrung des vorherigen Satzes zeigen.

Satz 4.2
Sei $(P_n(x))_{n\geq 0}$ ein OPS bzgl. einer Gewichtsfunktion $w(x)$, welche die Pearsonsche Differentialgleichung mit einem quadratischen Polynom $\sigma(x)$ und einem linearen Polynom $\tau(x)$ sowie den Randbedingungen

$$\lim_{x\to s} x^n\sigma(x)w(x) = \lim_{x\to t} x^n\sigma(x)w(x) = 0, \quad n \in \mathbb{N}_{\geq 0}$$

erfüllt. Dann handelt es sich bei $(P_n(x))_{n\geq 0}$ um ein klassisches OPS, das der Eigenwertdifferentialgleichung genügt.

Beweis:
Sei o.B.d.A. k_n der höchste Koeffizient von $P_n(x)$ mit $k_n = 1$. Wir definieren

$$Q_n(x) := (ax^2 + bx + c)P_n''(x) + (dx + e)P_n'(x)$$

für alle $n \in \mathbb{N}_{\geq 1}$ und $Q_0(x) := 1$. Wenden wir nach zweimaliger partieller Integration die Pearsonsche Differentialgleichung sowie die Randbedingungen an, so erhalten wir

$$\int_s^t w(x)x^m Q_n(x)dx = m(d + (m-1)a) \int_s^t w(x)x^m P_n(x)dx.$$

Es gilt also:

$$\begin{aligned}
\int_s^t w(x)x^m Q_n(x)dx &= 0 \quad (0 \leq m \leq n-1) \\
\int_s^t w(x)x^n Q_n(x)dx &= n(d + (n-1)a)h_n \quad (n \in \mathbb{N}) \\
\int_s^t w(x)Q_0(x)dx &= h_0.
\end{aligned}$$

Wegen (4) gilt $h_n \neq 0$. Wir müssen also noch zeigen, dass $d+(n-1)a \neq 0$ gilt. Zusammen mit den Randbedingungen erhalten wir

$$\begin{aligned}
(\sigma(x)w(x))' &= \tau(x)w(x) \\
\Rightarrow (\sigma(x)w(x))' x^n &= \tau(x)w(x)x^n \\
\Rightarrow \int_s^t (\sigma(x)w(x))' x^n dx &= \int_s^t \tau(x)w(x)x^n dx \\
\Rightarrow (\sigma(x)w(x)x^n)_s^t - n\int_s^t \sigma(x)w(x)x^{n-1}dx &= \int_s^t \tau(x)w(x)x^n dx \\
\Rightarrow -n\int_s^t (ax^2 + bx + c)x^{n-1}w(x)dx &= \int_s^t (dx + e)w(x)x^n dx \\
\Rightarrow -na\mu_{n+1} - nb\mu_n - nc\mu_{n-1} &= d\mu_{n+1} + e\mu_n \\
\Rightarrow (d + na)\mu_{n+1} = -(nb + e)\mu_n - nc\mu_{n-1}, &
\end{aligned} \tag{18}$$

dabei sind μ_n die n-ten Momente von $w(x)$. Damit wir das n-te Moment in eindeutiger Weise aus seinen Vorgängern berechnen können, muss $d + (n-1)a \neq 0$ gelten. Somit ist $d + (n-1)a \neq 0$ gezeigt.
$Q_n(x)$ ist wegen $Q_n(x) = n(d + (n-1)a)x^n + \ldots$ ein Polynom vom Grad n. Demnach ist $(Q_n(x))_{n\geq 0}$ ein OPS bzgl. der Gewichtsfunktion $w(x)$, weswegen $Q_n(x)$ für jedes $n \in \mathbb{N}$ ein Vielfaches von $P_n(x)$ sein muss:

$$Q_n(x) = -\lambda_n P_n(x) \quad (n \in \mathbb{N}_{\geq 0}).$$

Wegen $k_n = 1$ folgt, dass $\lambda_n = -n(d + (n-1)a)$ gilt. □

4.3 Darstellung

Im Folgenden wollen wir Potenzreihenentwicklungen der klassischen orthogonalen Polynome betrachten. Wir werden sehen, dass bei geeigneter Wahl des Entwicklungspunktes alle diese Polynomsysteme hypergeometische Potenzreihen besitzen, welche uns explizite Darstellungen liefern. Dazu definieren wir zuerst:

Definition 4.3 (Pochhammer-Symbol)
Wir bezeichnen

$$(a)_k := \underbrace{a(a+1)(a+2)\ldots(a+k-1)}_{k-Faktoren} = \frac{\Gamma(a+k)}{\Gamma(a)}$$

als Pochhammer-Symbol.

Definition 4.4 (Hypergeometrische Reihe)
Die Potenzreihe

$$ {}_pF_q\left(\begin{array}{c} a_1, \ldots, a_p \\ \\ b_1, \ldots, b_q \end{array} \middle| \, z \right) = \sum_{k=0}^{\infty} A_k z^k,$$

deren Summanden $\alpha_k = A_k z^k$ *das rationale Termverhältnis*

$$\frac{\alpha_{k+1}}{\alpha_k} = \frac{A_{k+1}z^{k+1}}{A_k z^k} = \frac{(k+a_1)\ldots(k+a_p)}{(k+b_1)\ldots(k+b_q)} \frac{z}{k+1}$$

besitzen, nennt man hypergeometrische Reihe.
Die Summanden $\alpha_k = A_k z^k$ *einer hypergeometrischen Reihe nennt man hypergeometrischen Term.*
Für die Koeffizienten der hypergeometrischen Reihe erhalten wir die Formel

$$ {}_pF_q\left(\begin{array}{c} a_1, \ldots, a_p \\ \\ b_1, \ldots, b_q \end{array} \middle| \, z \right) = \sum_{k=0}^{\infty} \frac{(a_1)_k \ldots (a_p)_k}{(b_1)_k \ldots (b_q)_k} \frac{z^k}{k!}.$$

Wir wollen nun hypergeometrische Darstellungen von klassischen orthogonalen Polynomen herleiten. Dazu verwenden wir *Mathematica* und das *SpecialFunctions* Package von Prof. Dr. Wolfram Koepf,[2] welches die in [CompAlg] Kapitel 10 (Potenzreihen) und Kapitel 11 (Algorithmische Summation) vorgestellten sowie einige weitere Algorithmen zur Verfügung stellt.

Zuerst laden wir das *SpecialFunctions* Package.

```
In[15]:= Needs[„SpecialFunctions‘"]

SpecialFunctions, (C) Wolfram Koepf, version 2.03, 2008

Fast Zeilberger, (C) Peter Paule and Markus Schorn (V 2.2) loaded
```

[2]Erhältlich unter http://www.mathematik.uni-kassel.de/~koepf/Publikationen/index.html#down

Entwickeln wir am Ursprung, so erhalten wir die Reihe $P_n(x) = \sum_{k=0}^{\infty} A_k x^k$. Einsetzen in die Eigenwertdifferentialgleichung liefert uns:

```
In[16]:= DE := (a*x^2 + b*x + c)P''[x] + (d*x + e)P'[x] - (n(a*n + d - a))P[x].
```

Unter Verwendung der Funktion *DEtoRE*[3]

```
In[17]:= req = DEtoRE[DE, P[x], A[k]]
```

$Out[17]=$ $(k-n)(k\,a + n\,a - a + d)\,A(k) + (k+1)(e + b\,k)\,A(k+1) + c\,(k+1)(k+2)\,A(k+2) == 0$

Für die Laguerrepolynome entnehmen wir ($a = 0, b = 1, c = 0, d = -1, e = \alpha + 1$) aus Tabelle 1 und setzen dies in die Rekursion ein. Dies liefert uns

```
In[18]:= req/.{a → 0, b → 1, c → 0, d → -1, e → α + 1}
```

$Out[18]=$ $(k+1)(k+\alpha+1)\,A(k+1) - (k-n)\,A(k) == 0$

Somit erhalten wir eine hypergeometrische Rekursion, Umformung liefert uns

$$\frac{A_{k+1}}{A_k} = \frac{k-n}{(k+1)(k+\alpha+1)},$$

daraus erhalten wir dann die hypergeometrische Darstellung

$$L_n^{(\alpha)}(x) = \binom{n+\alpha}{n} {}_1F_1\left(\begin{array}{c} -n \\ \\ \alpha+1 \end{array}\middle|\, x\right) = \sum_{k=0}^{n} \binom{n+\alpha}{n-k} \frac{(-1)^k}{k!} x^k. \tag{19}$$

Dabei ist $\binom{n+\alpha}{n}$ ein historisch gegebener Standardisierungsfaktor. Mit (19) setzen wir die Standardisierung der Laguerrepolynome fest. Um zu beweisen, dass die Gleichheit (19) gilt, reicht es zu zeigen, dass das Termverhältnis $\frac{B_{k+1}}{B_k}$, $B_k = \binom{n+\alpha}{n-k}\frac{(-1)^k}{k!}$ mit dem von $\frac{A_{k+1}}{A_k}$ übereinstimmt. In diesen Fällen hätten sie die gleiche hypergeometrische Darstellung und wären folglich gleich. Wir definieren dazu

```
In[19]:= Bk = Binomial[n + α, n - k] (-1)^k / k!;
```

und erhalten

```
In[20]:= (Bk/. k → k + 1) / Bk //FunctionExpand
```

$Out[20]=$ $-\frac{n-k}{(k+1)(k+\alpha+1)}$

Somit ist die Gleichheit gezeigt.

Für die Hermitepolynome erhalten wir mit ($a = 0, b = 0, c = 1, d = -2, e = 0$) die Rekursion

```
In[21]:= req/.{a → 0, b → 0, c → 1, d → -2, e → 0}
```

$Out[21]=$ $(k+1)(k+2)\,A(k+2) - 2\,(k-n)\,A(k) == 0$

[3] Siehe Abschnitt „Erzeugende Funktionen“ für mehr Details zu den Funktionen *DEtoRE* und *RETODE*.

vom hypergeometrischen Typ, welche nach Umformung die Gestalt

$$\frac{A_{k+2}}{A_k} = \frac{2(k-n)}{(k+1)(k+2)}$$

besitzt. Da die obige Rekursion eine Rekursion zweiten Grades ist, erhalten wir für gerade bzw. ungerade n die Darstellungen

$$H_{2m}(x) = (-1)^m \frac{(2m)!}{m!}\ {}_1F_1\left(\left.\begin{array}{c} -m \\ \\ 1/2 \end{array}\right| x^2\right)$$

und

$$H_{2m+1}(x) = (-1)^m \frac{(2m+1)!}{m!} 2x\ {}_1F_1\left(\left.\begin{array}{c} -m \\ \\ 3/2 \end{array}\right| x^2\right).$$

Schreiben wir nun $H_n(x) = x^n F(\frac{1}{x^2})$, also $F(y) = x^{-n} H_n(x)|_{x=1/\sqrt{y}}$, so liefert dies eine echte hypergeometrische Reihe, welche algebraisch hergeleitet werden kann [CompAlg]. Dazu leiten wir die Differentialgleichung der Komposition her und konvertieren diese wieder in eine Rekursionsgleichung. Als Resultat erhalten wir die Rekursion

$$(n-2k)(n-1-2k)A_k + 4(k+1)A_{k+1} = 0.$$

Umformung liefert

$$\frac{A_{k+1}}{A_k} = -\frac{(n-2k)(n-1-2k)}{4(k+1)},$$

daraus erhalten wir dann

$$H_n(x) = (2x)^n\ {}_2F_0\left(\left.\begin{array}{c} -n/2, -(n-1)/2 \\ \\ - \end{array}\right| -\frac{1}{x^2}\right) = n! \sum_{k=0}^{\lfloor n/2 \rfloor} \frac{(-1)^k}{k!(n-2k)!} (2x)^{n-2k}, \qquad (20)$$

was wir als die Standardisierung der Hermitepolynome festsetzen. Der Standardisierungsfaktor 2^n ist wieder historisch bedingt.

Um die Gültigkeit der letzten Gleichheit zu verifizieren überprüfen wir wieder die Gleichheit der Termverhältnisse $\frac{B_{k+1}}{B_k}$. Dazu definieren wir

```
In[22]:= Bk = (-1)^k (2)^(n - 2k) / (k! (n - 2k)!);
```

und erhalten

```
In[23]:= (Bk /. k -> k + 1) / Bk // Simplify
```

Out[23]= $-\frac{(2\,k-n)\,(2\,k-n+1)}{4\,(k+1)}$

Aus der Gleichheit der Termverhältnisse folgt dann die Gleichheit und die Gültigkeit der Darstellung (20).

Für die Besselpolynome erhalten wir mit ($a = 1, b = 0, c = 0, d = \alpha + 2, e = 2$) die Rekursion

```
In[24]:= req/.{a → 1, b → 0, c → 0, d → α + 2, e → 2}
```

Out[24]= $(k - n)(k + n + \alpha + 1) A(k) + 2(k + 1) A(k + 1) == 0$

vom hypergeometrischen Typ, welche nach Umformung die Gestalt

$$\frac{A_{k+1}}{A_k} = \frac{(k-n)(k+n+\alpha+1)}{2(k+1)}$$

besitzt. Somit folgt die hypergeometrische Darstellung

$$B_n^{(\alpha)}(x) = {}_2F_0\left(\left.\begin{array}{c} -n, n+\alpha+1 \\ - \end{array}\right| -\frac{x}{2}\right) = \frac{(n+\alpha+1)_n}{2^n} x^n \, {}_1F_1\left(\left.\begin{array}{c} -n \\ -2n-\alpha \end{array}\right| \frac{2}{x}\right), \tag{21}$$

welche wir als die Standardisierung der Besselpolynome festsetzen.

Für die Jacobipolynome müssen wir um eine hypergeometrische Lösung zu erhalten an den Stellen $x_0 = \pm 1$ entwickeln.
Mit ($a = 1, b = 0, c = -1, d = \alpha + \beta + 2, e = \alpha - \beta$) erhalten wir dann die Rekursion

```
In[25]:= req2 = req/.{a → 1, b → 0, c → -1, d → α + β + 2, e → α - β}
```

Out[25]= $(k - n)(k + n + \alpha + \beta + 1) A(k) + (k + 1)(\alpha - \beta) A(k + 1) - (k + 1)(k + 2) A(k + 2) == 0$

Man sieht, das im Falle der Jacobipolynome, im Gegensatz zu den anderen klassischen OPS, keiner der Terme A_k, A_{k+1}, A_{k+2} wegfällt. Um nun eine hypergeometrische Rekursion zu erhalten, die nur aus zwei Termen besteht, müssen wir die Entwicklungspunkte -1 und $+1$ betrachten. Dazu transformieren wir die obige Rekursion mittels der Funktion *RETODE* aus dem *SpecialFunctions* Package in eine Differentialgleichung

```
In[26]:= DE2 = RETODE[req2, A, k, F, x]
```

Out[26]= $-n(n + \alpha + \beta + 1) F(x) +$
$(\alpha x + \beta x + 2x + \alpha - \beta) F'(x) + (x - 1)(x + 1) F''(x) == 0$

und substituieren $x \to z + 1$

```
In[27]:= DE3 =
    -n (n + α + β + 1) F[z] +
      (α (z + 1) + β (z + 1) + 2 (z + 1) + α - β) F′[z] +
      ((z + 1) - 1) ((z + 1) + 1) F″[z] == 0
```

Out[27]= $-n(n + \alpha + \beta + 1) F(z) +$
$(\alpha(z + 1) + \beta(z + 1) + 2(z + 1) + \alpha - \beta) F'(z) + z(z + 2) F''(z) == 0$

Anschließend transformieren wir die Differentialgleichung mittels *DEtoRE* zurück in eine Rekursionsgleichung. Dies liefert uns

```
In[28]:= DEtoRE[DE3, F[z], A[k]]
```

Out[28]= $(k - n)(k + n + \alpha + \beta + 1) A(k) + 2(k + 1)(k + \alpha + 1) A(k + 1) == 0$

Wir erhalten also eine Rekursion erster Ordnung, welche das Termverhältnis

$$\frac{A_{k+1}}{A_k} = -\frac{(k-n)(k+n+\alpha+\beta+1)}{2(k+1)(k+\alpha+1)}$$

liefert. Daraus folgt dann mit $z = x - 1$ die hypergeometrische Darstellung

$$P_n^{(\alpha,\beta)}(x) = \binom{n+\alpha}{n}\, {}_2F_1\left(\left.\begin{array}{c} -n, n+\alpha+\beta+1 \\ \\ \alpha+1 \end{array}\right| \frac{1-x}{2}\right) \tag{22}$$

$$= \frac{\Gamma(\alpha+n+1)}{n!\Gamma(\alpha+\beta+n+1)} \sum_{k=0}^{n} \binom{n}{k} \frac{\Gamma(\alpha+\beta+n+k+1)}{2^k\Gamma(\alpha+k+1)}(1-x)^k.$$

Um die obige Gleichheit zu zeigen betrachten wir wieder die Termverhältnisse. Dies liefert

```
In[29]:= Bk = Binomial[n, k] Gamma[α + β + n + k + 1] / (2^k Gamma[α + k + 1])

In[30]:= (Bk /. k → k + 1) / Bk // FunctionExpand
```

Out[30]= $\frac{(n-k)\,(k+n+\alpha+\beta+1)}{2\,(k+1)\,(k+\alpha+1)}$

Somit ist die Gleichheit gezeigt.
Als Nächstes wollen wir nun noch den Fall $x \to z - 1$ betrachten. Dazu substituieren wir dies in die zuvor bestimmte Differentialgleichung und erhalten

```
In[31]:= DE4 =
           -n (n + α + β + 1) F[z] +
              (α (z - 1) + β (z - 1) + 2 (z - 1) + α - β) F′[z] +
              ((z - 1) - 1) ((z - 1) + 1) F″[z] == 0
```

Out[31]= $-n\,(n+\alpha+\beta+1)\,F(z)+$
$(\alpha\,(z-1)+\beta\,(z-1)+2\,(z-1)+\alpha-\beta)\,F'(z)+(z-2)\,z\,F''(z)==0$

Rücktransformation mittels *DEtoRE* in eine Rekursionsgleichung liefert

```
In[32]:= DEtoRE[DE4, F[z], A[k]]
```

Out[32]= $(k-n)\,(k+n+\alpha+\beta+1)\,A(k)-2\,(k+1)\,(k+\beta+1)\,A(k+1)==0$

mit dem Termverhältnis

$$\frac{A_{k+1}}{A_k} = \frac{(k-n)(k+n+\alpha+\beta+1)}{2(k+1)(k+\beta+1)}.$$

Daraus folgt dann mit $z = 1 + x$ die hypergeometrische Darstellung

$$P_n^{(\alpha,\beta)}(x) = (-1)^n\binom{n+\beta}{n}\, {}_2F_1\left(\left.\begin{array}{c} -n, n+\alpha+\beta+1 \\ \\ \beta+1 \end{array}\right| \frac{1+x}{2}\right). \tag{23}$$

Mit (22) setzen wir im Folgenden die Standardisierung der Jacobipolynome fest.

Als Nächstes wollen wir explizite Darstellungen für die Momente der Laguerre-, Hermite- und Jacobipolynome herleiten. Die Momente der Besselpolynome können wir nicht bestimmen, da die Besselpolynome in keinem reellen Intervall existieren und somit das Integral $\int_s^t x^n w(x)dx$ nicht existiert. Prinzipiell kann man die Momente auf analytische Weise durch Bestimmung des Integrals $\int_s^t x^n w(x)dx$ für die jeweiligen Intervalle und Gewichtsfunktionen der klassischen OPS bestimmmen oder auf algebraische Weise aus der Rekursiongleichung

$$cn\mu_{n-1} + (e + bn)\mu_n + (d + an)\mu_{n+1} = 0,$$

welche wir bereits hergeleitet haben (18). Wir werden für die Laguerre- und Hermitepolynome die algebraische Herleitung verwenden, da in diesen Fällen die Rekursion ein hypergeometrisches Termverhältnis aufweist und wir somit besonders einfache Darstellungen erhalten. Im Falle der Jacobipolynome ist die Herleitung aus der Rekursion wesentlich schwieriger, weswegen wir darauf verzichten und die analytische Variante wählen.

Für die Laguerrepolynome setzen wir also $(a = 0, b = 1, c = 0, d = -1, c = \alpha + 1)$ und erhalten die Rekursion

$$(\alpha + 1 + n)\mu_n - \mu_{n+1} = 0.$$

Umformung liefert

$$\frac{\mu_{n+1}}{\mu_n} = \alpha + 1 + n = \frac{(\alpha + 1 + n)(n + 1)}{n + 1}.$$

Somit erhalten wir

$$\mu_n = \frac{(\alpha + 1)_n (1)_n}{n!} \mu_0.$$

Mit *Mathematica* erhalten wir für μ_0

```
In[33]:= μ0 = Integrate[Exp[-x]x^α, {x, 0, ∞}, Assumptions → α > -1]
Out[33]= Γ(α + 1)
```

Einsetzen liefert uns dann μ_n

```
In[34]:= Pochhammer[α + 1, n]Pochhammer[1, n] / n! * μ0//FunctionExpand
Out[34]= Γ(n + α + 1)
```

Ein Vergleich mit der Integralherleitung bestätigt unser Resultat

```
In[35]:= Integrate[x^n Exp[-x]x^α, {x, 0, ∞},
           Assumptions → {α > -1 && n ≥ 0 }]
Out[35]= Γ(n + α + 1)
```

Für die Hermitepolynome setzen wir $(a = 0, b = 0, c = 1, d = -2, e = 0)$ und erhalten die Rekursion

$$n\mu_{n-1} - 2\mu_{n+1} = 0.$$

Umformung und Shift von n liefert uns

$$\frac{\mu_{n+2}}{\mu_n} = \frac{n+1}{2}.$$

Da die Rekursion die Symmetriezahl 2 hat, müssen wir sie in einen geraden und einen ungeraden Anteil zerlegen. Wegen

```
In[36]:= μ1 = Integrate[x ⅇ^(-x^2), {x, -∞, ∞},
              Assumptions → { n ≥ 0}]
Out[36]= 0
```

verschwindet aber der ungerade Anteil. Setzen wir nun $a_k = \mu_{2n}$, so erhalten wir

$$a_k = \frac{(\frac{1}{2})_n (1)_k}{k!} \mu_0.$$

Wir bestimmen μ_0

```
In[37]:= μ0 = Integrate[ ⅇ^(-x^2), {x, -∞, ∞},
              Assumptions → { n ≥ 0}]
Out[37]= √π
```

und erhalten somit für a_k

```
In[38]:= Pochhammer[1/2, k]Pochhammer[1, k]/k! μ0 //FunctionExpand
Out[38]= Γ(k + 1/2)
```

für gerade $k = 2n$.

Insgesamt können wir also auch definieren

$$\mu_n = \frac{1}{2}(1 + (-1)^n)\Gamma\left(\frac{n+1}{2}\right) = \begin{cases} 0 & , n \text{ gerade} \\ \Gamma(\frac{n+1}{2}) & , n \text{ ungerade.} \end{cases}$$

Der Vergleich mit der Integralherleitung bestätigt unser Resultat.

```
In[39]:= Integrate[x^n ⅇ^(-x^2), {x, -∞, ∞},
            Assumptions → { n ≥ 0}]
Out[39]= 1/2 (1 + (-1)^n) Γ((n + 1)/2)
```

Für die Jacobipolynome erhalten wir mit ($a = 1, b = 0, c = -1, d = \alpha+\beta+2, e = \alpha-\beta$) die Rekursion

$$-n\mu_{n-1} + (\alpha - \beta)\mu_n + (\alpha + \beta + 2 + n)\mu_{n+1} = 0.$$

Wir sehen, dass im Gegensatz zu den anderen Polynomsystemen keiner der Terme μ wegfällt. Es können also keine der obigen Methoden angewendet werden. Die Rekursion bietet natürlich immer noch die Möglichkeit die jeweiligen Momente aus deren Vorgängern zu berechnen. Dazu benötigen wir lediglich zwei Startwerte, welche wir leicht finden können.

Um jedoch eine explizite Darstellung zu finden betrachten wir das Integral

$$\int_{-1}^{1} (1-x)^\alpha (1+x)^\beta x^n dx,$$

welches ein Beta-Integral ist. Siehe dazu z.B. [Tab] S.1089.
Lösen mit *Mathematica* liefert dann die explizite Darstellung der Jacobipolynome

```
In[40]:= Jacobiμ =
           Integrate[x^n (1 + x)^β (1 - x)^α, {x, -1, 1},
              Assumptions → { n ≥ 0&&α > -1&&β > -1}]//FunctionExpand
```

Out[40]= $$\frac{(-1)^n\,\Gamma(n+1)\,\Gamma(\beta+1)\,{}_2F_1(n+1,-\alpha;n+\beta+2;-1)}{\Gamma(n+\beta+2)}+\frac{\Gamma(n+1)\,\Gamma(\alpha+1)\,{}_2F_1(n+1,-\beta;n+\alpha+2;-1)}{\Gamma(n+\alpha+2)}$$

4.4 Struktureigenschaften

Im folgenden Abschnitt sollen weitere Eigenschaften klassischer OPS untersucht werden. Dazu betrachten wir zunächst eine weitere Rekursion.

Satz 4.5 (Rekursion)
Erfüllt ein Polynomsystem $P_n(x)$ die Dreitermrekursion (5), dann erfüllt es auch die Rekursion

$$xP_n(x) = a_nP_{n+1}(x) + b_nP_n(x) + c_nP_{n-1}(x) \tag{24}$$

mit

$$a_n = \frac{1}{A_n}, \qquad b_n = -\frac{B_n}{A_n}, \qquad c_n = \frac{C_n}{A_n}. \tag{25}$$

Beweis:
Sei also $P_n(x)$ ein Polynomsystem, welches die Dreitermrekursion (5) erfüllt. Dann gilt für $P_n(x)$:

$$P_{n+1}(x) = (A_nx + B_n)P_n(x) - C_nP_{n-1}(x).$$

Durch Umformung erhalten wir

$$\begin{aligned}
\Leftrightarrow P_{n+1}(x) &= A_nxP_n(x) + B_nP_n(x) - C_nP_{n-1}(x)\\
\Leftrightarrow -A_nxP_n(x) &= -P_{n+1}(x) + B_nP_n(x) - C_nP_{n-1}(x)\\
\Leftrightarrow xP_n(x) &= \frac{1}{A_n}P_{n+1}(x) + \left(-\frac{B_n}{A_n}\right)P_n(x) + \frac{C_n}{A_n}P_{n-1}(x)
\end{aligned}$$

Setzen wir nun

$$a_n = \frac{1}{A_n}, \qquad b_n = -\frac{B_n}{A_n}, \qquad c_n = \frac{C_n}{A_n},$$

so erhalten wir die obige Rekursion. Da $P_n(x)$ (5) erfüllt erfüllt es somit auch (24). □

Wir werden nun sehen, dass man für die Koeffizienten A_n, B_n, C_n und damit auch a_n, b_n, c_n explizite Darstellungen herleiten kann.

Satz 4.6 (Koeffizienten der Dreitermrekursion)
Sei $P_n(x) = k_nx^n + k'_nx^{n-1} + k''_nx^{n-2} + ...(n \in \mathbb{N}_{\geq 0})$ ein klassisches OPS, welches die Eigenwertdifferentialgleichung mit einem quadratischen Polynom $\sigma(x)$ und einem linearen Polynom $\tau(x)$ erfüllt. Dann erfüllt $P_n(x)$ die Rekursionsgleichung (5) $P_{n+1}(x) =$

$(A_n x + B_n)P_n(x) - C_n P_{n-1}(x) \quad (n \geq 0)$ *mit den generischen Formeln*

$$A_n = \frac{k_{n+1}}{k_n}$$

sowie

$$B_n = \frac{2bn(an + d - a) - e(-d + 2a)}{(d + 2an)(d - 2a + 2an)} \frac{k_{n+1}}{k_n}$$

und

$$C_n = \frac{\left((an + d - 2a)n(4ca - b^2) + 4a^2c - ab^2 + ae^2 - 4acd + db^2 - bed + d^2c\right)(an + d - 2a)n}{(d - 2a + 2an)^2(2an - 3a + d)(2an - a + d)} \frac{k_{n+1}}{k_{n-1}}.$$

Die Koeffizienten a_n, b_n und c_n ergeben sich dann gemäß (25).

Beweis:
Die Koeffizienten erhalten wir durch Koeffizientenvergleich. Dazu verwenden wir *Mathematica*. Wir betrachten die drei höchsten Koeffizienten

In[41]:= $\mathbf{p = k_n x^n + kstr_n x^{n-1} + kstrstr_n x^{n-2}}$
Out[41]= $\mathrm{kstrstr}_n\, x^{n-2} + \mathrm{kstr}_n\, x^{n-1} + k_n\, x^n$

und setzen diese in die Eigenwertdifferentialgleichung ein

In[42]:= **EDGL = (a * x^2 + b * x + c) * D[p, {x, 2}] + (d * x + e) * D[p, x] + λ_n p**
Out[42]= $(a\, x^2 + b\, x + c)$
$\left((n-3)\,(n-2)\,\mathrm{kstrstr}_n\, x^{n-4} + (n-2)\,(n-1)\,\mathrm{kstr}_n\, x^{n-3} + (n-1)\, n\, k_n\, x^{n-2}\right) +$
$(e + d\, x)\left((n-2)\,\mathrm{kstrstr}_n\, x^{n-3} + (n-1)\,\mathrm{kstr}_n\, x^{n-2} + n\, k_n\, x^{n-1}\right) +$
$\left(\mathrm{kstrstr}_n\, x^{n-2} + \mathrm{kstr}_n\, x^{n-1} + k_n\, x^n\right)\lambda_n$

Als Nächstes teilen wir durch x^{n-4} und multiplizieren aus

In[43]:= **liste = Together[EDGL/x^(n-4)]**
Out[43]= $a\, n^2\, k_n\, x^4 - a\, n\, k_n\, x^4 + d\, n\, k_n\, x^4 + k_n\, \lambda_n\, x^4 + b\, n^2\, k_n\, x^3 - b\, n\, k_n\, x^3 +$
$e\, n\, k_n\, x^3 + a\, n^2\, \mathrm{kstr}_n\, x^3 + 2\, a\, \mathrm{kstr}_n\, x^3 - d\, \mathrm{kstr}_n\, x^3 - 3\, a\, n\, \mathrm{kstr}_n\, x^3 +$
$d\, n\, \mathrm{kstr}_n\, x^3 + \mathrm{kstr}_n\, \lambda_n\, x^3 + c\, n^2\, k_n\, x^2 - c\, n\, k_n\, x^2 + b\, n^2\, \mathrm{kstr}_n\, x^2 + 2\, b\, \mathrm{kstr}_n\, x^2 -$
$e\, \mathrm{kstr}_n\, x^2 - 3\, b\, n\, \mathrm{kstr}_n\, x^2 + e\, n\, \mathrm{kstr}_n\, x^2 + a\, n^2\, \mathrm{kstrstr}_n\, x^2 + 6\, a\, \mathrm{kstrstr}_n\, x^2 -$
$2\, d\, \mathrm{kstrstr}_n\, x^2 - 5\, a\, n\, \mathrm{kstrstr}_n\, x^2 + d\, n\, \mathrm{kstrstr}_n\, x^2 + \mathrm{kstrstr}_n\, \lambda_n\, x^2 + c\, n^2\, \mathrm{kstr}_n\, x +$
$2\, c\, \mathrm{kstr}_n\, x - 3\, c\, n\, \mathrm{kstr}_n\, x + b\, n^2\, \mathrm{kstrstr}_n\, x + 6\, b\, \mathrm{kstrstr}_n\, x - 2\, e\, \mathrm{kstrstr}_n\, x -$
$5\, b\, n\, \mathrm{kstrstr}_n\, x + e\, n\, \mathrm{kstrstr}_n\, x + c\, n^2\, \mathrm{kstrstr}_n + 6\, c\, \mathrm{kstrstr}_n - 5\, c\, n\, \mathrm{kstrstr}_n$

Jetzt legen wir eine Koeffizientenliste bzgl. x an

In[44]:= **Coeff = CoefficientList[liste, x]**
Out[44]= $\{c\, \mathrm{kstrstr}_n\, n^2 - 5\, c\, \mathrm{kstrstr}_n\, n + 6\, c\, \mathrm{kstrstr}_n,\ c\, \mathrm{kstr}_n\, n^2 + b\, \mathrm{kstrstr}_n\, n^2 -$
$3\, c\, \mathrm{kstr}_n\, n - 5\, b\, \mathrm{kstrstr}_n\, n + e\, \mathrm{kstrstr}_n\, n + 2\, c\, \mathrm{kstr}_n + 6\, b\, \mathrm{kstrstr}_n - 2\, e\, \mathrm{kstrstr}_n,$
$c\, k_n\, n^2 + b\, \mathrm{kstr}_n\, n^2 + a\, \mathrm{kstrstr}_n\, n^2 - c\, k_n\, n - 3\, b\, \mathrm{kstr}_n\, n + e\, \mathrm{kstr}_n\, n - 5\, a\, \mathrm{kstrstr}_n\, n +$
$d\, \mathrm{kstrstr}_n\, n + 2\, b\, \mathrm{kstr}_n - e\, \mathrm{kstr}_n + 6\, a\, \mathrm{kstrstr}_n - 2\, d\, \mathrm{kstrstr}_n + \mathrm{kstrstr}_n\, \lambda_n,$
$b\, k_n\, n^2 + a\, \mathrm{kstr}_n\, n^2 - b\, k_n\, n + e\, k_n\, n - 3\, a\, \mathrm{kstr}_n\, n + d\, \mathrm{kstr}_n\, n +$
$2\, a\, \mathrm{kstr}_n - d\, \mathrm{kstr}_n + \mathrm{kstr}_n\, \lambda_n,\ a\, k_n\, n^2 - a\, k_n\, n + d\, k_n\, n + k_n\, \lambda_n\}$

Koeffizientenvergleich liefert uns nun λ_n

```
In[45]:= sub1 = Solve[Coeff[[5]] == 0, λn][[1]]
```

Out[45]= $\{\lambda_n \to -a\,n^2 + a\,n - d\,n\}$

Einsetzen von λ_n in die Koeffizientenliste und Koeffizientenvergleich liefert uns k_n'

```
In[46]:= sub2 =
          Map[Factor, (Solve[Coeff[[4]] == 0, kstrn]/.sub1[[1]])[[1]],
           {2}]
```

Out[46]= $\left\{\text{kstr}_n \to \frac{n\,(n\,b - b + e)\,k_n}{2\,n\,a - 2\,a + d}\right\}$

Einsetzen von k_n' in die Koeffizientenliste und Koeffizientenvergleich liefert uns k_n''

```
In[47]:= sub3 =
          Map[Factor,
           (Solve[Coeff[[3]] == 0, kstrstrn]/.{sub1[[1]], sub2[[1]]})[[
            1]], {2}]
```

Out[47]= $\left\{\text{kstrstr}_n \to \frac{(n-1)\,n\,(n^2\,b^2 - 3\,n\,b^2 + 2\,b^2 - 3\,e\,b + 2\,e\,n\,b + e^2 - 2\,a\,c + c\,d + 2\,a\,c\,n)\,k_n}{2\,(2\,n\,a - 3\,a + d)\,(2\,n\,a - 2\,a + d)}\right\}$

Als Nächstes wollen wir a_n, b_n und c_n aus (24) bestimmen. Dazu definieren wir zuerst

```
In[48]:= pn = p /.{sub2[[1]], sub3[[1]]};

         pn+1 = pn/.n → n + 1;

         pn−1 = pn/.n → n − 1;
```

Einsetzen liefert

```
In[49]:= rekgl1 = x * pn − an * pn+1 − bnpn − cnpn−1
```

```
Out[49]= -c_n (((((n-1)^2 b^2 - 3 (n-1) b^2 + 2 b^2 - 3 e b + 2 e (n-1) b +
            e^2 - 2 a c + c d + 2 a c (n-1)) (n-2) (n-1) k_{n-1} x^{n-3}) /
        (2 (2 (n-1) a - 3 a + d) (2 (n-1) a - 2 a + d)) +
        ((n-1) b - b + e) (n-1) k_{n-1} x^{n-2} / (2 (n-1) a - 2 a + d) + k_{n-1} x^{n-1}) +
   x (((n-1) n (n^2 b^2 - 3 n b^2 + 2 b^2 - 3 e b + 2 e n b + e^2 - 2 a c + c d + 2 a c n)
        k_n x^{n-2}) / (2 (2 n a - 3 a + d) (2 n a - 2 a + d)) +
        n (n b - b + e) k_n x^{n-1} / (2 n a - 2 a + d) + k_n x^n) -
   b_n (((n-1) n (n^2 b^2 - 3 n b^2 + 2 b^2 - 3 e b + 2 e n b + e^2 - 2 a c + c d + 2 a c n)
        k_n x^{n-2}) / (2 (2 n a - 3 a + d) (2 n a - 2 a + d)) +
        n (n b - b + e) k_n x^{n-1} / (2 n a - 2 a + d) + k_n x^n) -
   a_n ((n (n+1) ((n+1)^2 b^2 - 3 (n+1) b^2 + 2 b^2 - 3 e b + 2 e (n+1) b +
            e^2 - 2 a c + c d + 2 a c (n+1)) k_{n+1} x^{n-1}) /
        (2 (2 (n+1) a - 3 a + d) (2 (n+1) a - 2 a + d)) +
        (n+1) ((n+1) b - b + e) k_{n+1} x^n / (2 (n+1) a - 2 a + d) + k_{n+1} x^{n+1})
```

Als Nächstes erstellen wir wieder eine Koeffizientenliste

```
In[50]:= coeff4 = CoefficientList[rekgl1/x^(n-3), x];
```

Koeffizientenvergleich liefert uns dann a_n

```
In[51]:= sub4 = Solve[coeff4[[5]] == 0, a_n][[1]]
```

$$\text{Out[51]}= \left\{a_n \to \frac{k_n}{k_{n+1}}\right\}$$

Einsetzen von a_n und Koeffizientenvergleich liefert

```
In[52]:= sub5 = Map[Factor, Solve[coeff4[[4]] == 0, b_n]/.sub4, {3}][[
           1]]
```

$$\text{Out[52]}= \left\{b_n \to -\frac{2\,a\,b\,n^2 - 2\,a\,b\,n + 2\,b\,d\,n - 2\,a\,e + d\,e}{(d + 2\,a\,n)\,(2\,n\,a - 2\,a + d)}\right\}$$

Einsetzen von b_n und Koeffizientenvergleich liefert

```
In[53]:= sub6 =
          Map[Factor, Solve[coeff4[[3]] == 0, c_n]/.
              {sub4[[1]], sub5[[1]]}, {3}][[1]]
```

$$\text{Out[53]}= \left\{c_n \to \left(n\,(n\,a - 2\,a + d)\left(-4\,c\,n^2\,a^2 - 4\,c\,a^2 + 8\,c\,n\,a^2 + b^2\,a - e^2\,a + b^2\,n^2\,a + 4\,c\,d\,a - 2\,b^2\,n\,a - 4\,c\,d\,n\,a - c\,d^2 - b^2\,d + b\,d\,e + b^2\,d\,n\right)k_n\right) \Big/ \left((2\,n\,a - 3\,a + d)\,(2\,n\,a - 2\,a + d)^2\,(2\,n\,a - a + d)\,k_{n-1}\right)\right\}$$

Mit Hilfe von (25) können wir dann noch A_n, B_n und C_n bestimmen

```
In[54]:= A_n = 1/a_n /.sub4
```

```
Out[54]= k_{n+1}/k_n
In[55]:= B_n = -b_n A_n /. sub5
Out[55]= ((2 a b n^2 - 2 a b n + 2 b d n - 2 a e + d e) k_{n+1}) / ((d + 2 a n) (2 n a - 2 a + d) k_n)
In[56]:= C_n = c_n A_n /. sub6
Out[56]= (n (n a - 2 a + d) (-4 c n^2 a^2 - 4 c a^2 + 8 c n a^2 + b^2 a - e^2 a + b^2 n^2 a +
            4 c d a - 2 b^2 n a - 4 c d n a - c d^2 - b^2 d + b d e + b^2 d n) k_{n+1}) /
         ((2 n a - 3 a + d) (2 n a - 2 a + d)^2 (2 n a - a + d) k_{n-1})
```

□

Als Folgerung aus Satz 4.6 erhalten wir auch eine explizite Formel für die Quadratnorm.

Satz 4.7 (Quadratnorm)
Sei $P_n(x) = k_n x^n + ...(n \in \mathbb{N}_{\geq 0})$ ein klassisches OPS, welches die Eigenwertdifferentialgleichung mit einem quadratischen Polynom $\sigma(x)$ und einem linearen Polynom $\tau(x)$ erfüllt. Dann gilt die Beziehung:

$$\frac{h_{n+1}}{h_n} = \frac{(n+1)(a(n-1)+d)(dbe - cd^2 + db^2 n - 4dcan - 4n^2ca^2 - e^2a + n^2b^2a)}{(d+2an)^2(2an+a+d)(2an-a+d)} \left(\frac{k_{n+1}}{k_n}\right)^2$$

Beweis:
Gemäß (6) und (8) gelten für A_n und C_n die Gleichungen

$$C_n = \frac{A_n}{A_{n-1}} \frac{h_n}{h_{n-1}} \text{und } A_n = \frac{k_{n+1}}{k_n}.$$

Formt man nun (8) um, führt eine Indexverschiebung durch und setzt (6) ein, so erhält man

$$\frac{h_{n+1}}{h_n} = C_{n+1} \frac{A_n}{A_{n+1}} = C_{n+1} \frac{k_{n+1}^2}{k_n k_{n+2}}.$$

Setzen wir nun die in Satz 4.6 gewonnene Formel für C_n ein, so erhalten wir die obige Darstellung für $\frac{h_{n+1}}{h_n}$. □

4.4.1 Ableitung klassischer OPS

Als Nächstes wollen wir Ableitungen klassischer OPS genauer betrachten. Dazu zeigen wir

Satz 4.8 (Ableitungsregel)
Ein klassisches OPS, das Lösung der Eigenwertdifferentialgleichung ist, erfüllt eine Ableitungsregel der Form:

$$\sigma(x)P_n'(x) = \alpha_n P_{n+1}(x) + \beta_n P_n(x) + \gamma_n P_{n-1}(x) \quad (n \in \mathbb{N}) \tag{26}$$

Beweis:
Wir zeigen, dass eine Ableitungsregel der Form

$$\sigma(x)P_n'(x) = (\widetilde{\alpha_n}x + \widetilde{\beta_n})P_n(x) + \widetilde{\gamma_n}P_{n-1}(x) \quad (n \in \mathbb{N}) \tag{27}$$

gilt. (26) erhalten wir dann, indem wir die Rekursionsgleichung (24) anwenden. Setzen wir nun $P_n(x) = k_n x^n + k_n' x^{n-1} + k_n'' x^{n-2} + ...(n \in \mathbb{N}_{\geq 0})$ ein und vergleichen den höchsten Koeffizienten x^{n+1}, so erhalten wir

$$\widetilde{\alpha_n} = an.$$

Nach Vorraussetzung ist $P_n(x)$ bzgl. $w(x)$ orthogonal. Als Nächstes betrachten wir die selbstadjungierte Form (17)

$$\left(w(x)\sigma(x)P_n'(x)\right)' + \lambda_n w(x)P_n(x) = 0$$

und zeigen, dass daraus die Beziehung

$$\int_s^t w(x)\sigma(x)P_n'(x)f(x)dx = 0 \tag{28}$$

für alle Polynome $f(x)$ vom Grad $\leq n-2$ folgt. Sei $F(x)$ die Stammfunktion von $f(x)$, welche den Grad $\leq n-1$ hat. Dann erhalten wir mit partieller Integration

$$\int_s^t w(x)\sigma(x)P_n'(x)f(x)dx = \left(w(x)\sigma(x)P_n'(x)F(x)\right)_s^t - \int_s^t \left(w(x)\sigma(x)P_n'(x)\right)' F(x)dx$$

$$= \lambda_n \int_s^t w(x)P_n(x)F(x)dx = 0.$$

Der letzte Schritt folgt aus der Orthogonalität von $P_n(x)$ und $F(x)$. Aufgrund von $\widetilde{\alpha_n} = an$ ist der Grad des Polynoms $\sigma(x)P_n'(x) - \widetilde{\alpha_n}x \leq n$. Es gilt also

$$\sigma(x)P_n'(x) - \widetilde{\alpha_n}xP_n(x) = \sum_{j=0}^{n} e_j P_n(x).$$

Mit (28) folgt dann $e_j = 0$ für $0 \leq j \leq n-2$. Setzen wir $e_j = \widetilde{\beta_n}$ und $e_{j-1} = \widetilde{\gamma_n}$, so erhalten wir (27). □

In Satz 4.6 haben wir gesehen, dass man mittels linearer Algebra Darstellungen für die Koeffizienten finden kann. Dies wollen wir nun auch für α_n, β_n und γ_n tun.

Satz 4.9 (Koeffizienten der Ableitungsregel)
Sei $P_n(x) = k_n x^n + ...(n \in \mathbb{N}_{\geq 0})$ ein klassisches OPS, welches die Eigenwertdifferentialgleichung mit einem quadratischen Polynom $\sigma(x)$ und einem linearen Polynom $\tau(x)$ erfüllt. Dann erfüllt $P_n(x)$ die Ableitungsregel mit folgenden generischen Koeffizienten:

$$\alpha_n = an\frac{k_n}{k_{n+1}}$$

sowie

$$\beta_n = \frac{n(a(n-1)+d)(bd-2ae)}{(2a(n-1)+d)(2an+d)}$$

und

$$\gamma_n = \frac{n(a(n-1)+d)(a(n-2)+d)(n(an+d)(4ac-b^2)+ae^2+cd^2-bde)}{(a(2n-1)+d)(a(2n+3)+d)(2a(n-1)+d)^2}\frac{k_n}{k_{n-1}}.$$

Beweis:

Auch hier werden wir die Koeffizienten wieder mit Koeffizientenvergleich und *Mathematica* herleiten. Dazu verwenden wir die bereits hergeleiteten Formeln für λ_n, k'_n und k''_n. Diese setzen wir in die Ableitungregel ein, legen eine Koeffizientenliste an, nachdem wir durch x^{n-3} geteilt haben, und erhalten[4]

```
In[57]:= ablregel = αn pn+1 + βn pn + γn pn-1 -
            (a x^2 + b x + c) D[pn, x];
```

```
In[58]:= coeff5 = CoefficientList[ablregel/x^(n - 3), x] ;
```

Koeffizientenvergleich liefert

```
In[59]:= sub7 = Solve[coeff5[[5]] == 0, αn] [[1]]
```

Out[59]= $\left\{\alpha_n \to \frac{a\,n\,k_n}{k_{n+1}}\right\}$

Einsetzen von α_n und Koeffizientenvergleich liefert

```
In[60]:= sub8 = Map[Factor, Solve[coeff5[[4]] == 0, βn]/.sub7, {3}][[1]]
```

Out[60]= $\left\{\beta_n \to \frac{(b\,d-2\,a\,e)\,n\,(n\,a-a+d)}{(d+2\,a\,n)\,(2\,n\,a-2\,a+d)}\right\}$

Einsetzen von β_n und Koeffizientenvergleich liefert

```
In[61]:= sub9 =
          Map[Factor, Solve[coeff5[[3]] == 0, γn]/.
              {sub7[[1]], sub8[[1]]}, {3}][[1]]
```

Out[61]= $\Big\{\gamma_n \to \Big(n\,(n\,a-2\,a+d)\,(n\,a-a+d)$
$\big(4\,c\,n^2\,a^2+4\,c\,a^2-8\,c\,n\,a^2-b^2\,a+e^2\,a-b^2\,n^2\,a-4\,c\,d\,a+$
$2\,b^2\,n\,a+4\,c\,d\,n\,a+c\,d^2+b^2\,d-b\,d\,e-b^2\,d\,n\big)\,k_n\Big)\Big/$
$\Big((2\,n\,a-3\,a+d)\,(2\,n\,a-2\,a+d)^2\,(2\,n\,a-a+d)\,k_{n-1}\Big)\Big\}$

□

Zum Abschluss wollen wir zeigen, dass die Ableitungen von klassischen OPS wieder klassische OPS darstellen.

Satz 4.10 (Ableitung klassischer OPS)
Sei $P_n(x) = k_n x^n + ...(n \in \mathbb{N}_{\geq 0})$ ein klassisches OPS, welches die Eigenwertdifferentialgleichung mit einem quadratischen Polynom $\sigma(x)$ und einem linearen Polynom $\tau(x)$ erfüllt. Dann ist das Polynomsystem $(P'_{n+1}(x))_{n\geq 0}$ wieder ein klassisches OPS. Hiervon gilt auch die Umkehrung: Ein OPS $P_n(x)$, für welches die Folge der Ableitungen $P'_{n+1}(x)$ wieder ein OPS darstellt, ist klassisch.

[4] Der Output wird hier und im Folgenden unterdrückt, wenn er sehr lang ist.

Beweis:
Ableiten der Eigenwertdifferentialgleichung liefert uns

$$\sigma(x)y'''(x) + \sigma'(x)y''(x) + \tau(x)y''(x) + \tau'(x)y'(x) + \lambda_n y'(x) = 0.$$

Daher erfüllt $y(x) = P_n'(x)$ auch die Eigenwertdifferentialgleichung mit gleichem $\sigma(x)$ wie $P_n(x)$. Somit ist auch die Folge $(P_{n+1}'(x))_{n\geq 0}$ wieder ein klassisches OPS.

Sei $P_n'(x)$ nun ein klassisches OPS mit

$$\sigma(x)P_n'''(x) + \overline{\tau}(x)P_n''(x) + \rho_n P_n'(x) = 0,$$

$\overline{\tau}(x) = \tau(x) + \sigma'(x)$ und $\rho_n = \tau'(x) + \lambda_n$. Sei weiterhin $Q_n(x)$ ein klassisches OPS mit

$$\sigma(x)Q_n''(x) + \tau(x)Q_n'(x) + \lambda_n Q_n(x) = 0.$$

Wir definieren

$$Q_n(x) := -\frac{1}{\lambda_n}\left(\sigma(x)P_n''(x) + \tau(x)P_n'(x)\right).$$

Ableiten liefert dann

$$\begin{aligned} -\lambda_n Q_n'(x) &= \sigma(x)P_n'''(x) + \sigma'(x)P_n''(x) + \tau'(x)P_n'(x) + \tau(x)P_n''(x) \\ &= \sigma(x)P_n'''(x) + \left(\tau(x) + \sigma'(x)\right)P_n''(x) + \tau'(x)P_n'(x) \\ &= -\lambda_n P_n'(x). \end{aligned}$$

Folglich muss $Q_n(x) = P_n(x)$ gelten, weswegen auch $P_n(x)$ ein klassisches OPS ist, da es wieder die Eigenwertdifferentialgleichung mit quadratischem $\sigma(x)$, linearem $\tau(x) = \sigma'(x) - \overline{\tau}(x)$ und konstantem $\lambda_n = \rho_n - \tau'(x)$ erfüllt. □

4.4.2 Stammfunktionen klassischer OPS

Als Nächstes wollen wir Stammfunktionen klassischer OPS betrachten. Dazu leiten wir zuerst eine wichtige Strukturformel her.

Satz 4.11 (Strukturformel)
Sei $P_n(x) = k_n x^n + ...(n \in \mathbb{N}_{\geq 0})$ ein klassisches OPS, welches die Eigenwertdifferentialgleichung mit einem quadratischen Polynom $\sigma(x)$ und einem linearen Polynom $\tau(x)$ erfüllt. Dann erfüllt $P_n(x)$ die Strukturformel

$$P_n(x) = \widehat{a}_n P_{n+1}'(x) + \widehat{b}_n P_n'(x) + \widehat{c}_n P_{n-1}'(x) \tag{29}$$

Beweis:
Sei $(P_n(x))_{n\geq 0}$ ein klassisches OPS, dann gilt für $P_{n+1}'(x)$ wegen Satz 4.10 eine Rekursionsgleichung der Form

$$xP_n'(x) = \alpha_n^* P_{n+1}'(x) + \beta_n^* P_n'(x) + \gamma_n^* P_{n-1}'(x). \tag{30}$$

Als Nächstes leiten wir (26) ab

$$\sigma(x)P_n''(x) + \sigma'(x)P_n'(x) = \alpha_n P_{n+1}'(x) + \beta_n P_n'(x) + \gamma_n P_{n-1}'(x)$$

und ersetzen $xP_n'(x)$ mit Hilfe von (30), dies liefert uns

$$\sigma(x)P_n''(x) = a_n' P_{n+1}'(x) + b_n' P_n'(x) + c_n' P_{n-1}'(x). \tag{31}$$

Jetzt substituieren wir (31) in die Eigenwertdifferentialgleichung und erhalten

$$a_n' P_{n+1}'(x) + b_n' P_n'(x) + c_n' P_{n-1}'(x) + \tau(x)P_n'(x) + \lambda_n P_n(x) = 0.$$

Ersetzen wir nun $xP_n'(x)$ mit Hilfe von (30), so erhalten wir

$$P_n(x) = \widehat{a}_n P_{n+1}'(x) + \widehat{b}_n P_n'(x) + \widehat{c}_n P_{n-1}'(x)$$

□

Auch hier können explizite Formeln für die Koeffizienten mittels linearer Algebra hergeleitet werden.

Satz 4.12 (Koeffizienten der Strukturformel)
Sei $P_n(x) = k_n x^n + ...(n \in \mathbb{N}_{\geq 0})$ ein klassisches OPS, welches die Eigenwertdifferentialgleichung mit einem quadratischen Polynom $\sigma(x)$ und einem linearen Polynom $\tau(x)$ erfüllt. Dann erfüllt $P_n(x)$ die Strukturformel $P_n(x) = \widehat{a}_n P_{n+1}'(x) + \widehat{b}_n P_n'(x) + \widehat{c}_n P_{n-1}'(x)$ mit generischen Koeffizienten:

$$\widehat{a}_n = \frac{1}{n+1}\frac{k_n}{k_{n+1}}$$

sowie

$$\widehat{b}_n = \frac{2ea - db}{(d + 2an)(d - 2a + 2an)}$$

und

$$\widehat{c}_n = \frac{((n-1)(an + d - a)(4ca - b^2) + ae^2 + d^2c - bed)an}{(d - 2a + 2an)^2(2an - 3a + d)(2an - a + d)}\frac{k_n}{k_{n-1}}.$$

Beweis:
Auch hier werden wir die Koeffizienten wieder mit Koeffizientenvergleich und *Mathematica* herleiten. Dazu verwenden wir die bereits hergeleiteten Formeln für λ_n, k_n' und k_n''. Diese setzen wir in die Strukturformel ein, legen eine Koeffizientenliste an, nachdem wir durch x^{n-4} geteilt haben, und erhalten

```
In[62]:= struktformel = â_n D[p_{n+1}, x] + b̂_n D[p_n, x] + ĉ_n D[p_{n-1}, x] - p_n;

In[63]:= coeff6 = CoefficientList[struktformel/x^(n - 4), x] ;
```

Koeffizientenvergleich liefert

```
In[64]:= sub10 = Solve[coeff6[[5]] == 0, â_n][[1]]
```

Out[64]= $\left\{\hat{a}_n \to \frac{k_n}{(n+1)\,k_{n+1}}\right\}$

Einsetzen von $\widehat{a_n}$ und Koeffizientenvergleich liefert

```
In[65]:= sub11 = Map[Factor, Solve[coeff6[[4]] == 0, b̂_n]/.sub10, {3}][[1]]
```

Out[65]= $\left\{\hat{b}_n \to -\frac{b\,d - 2\,a\,e}{(d + 2\,a\,n)\,(2\,n\,a - 2\,a + d)}\right\}$

Einsetzen von $\widehat{b_n}$ und Koeffizientenvergleich liefert

```
In[66]:= sub12 =
      Map[Factor, Solve[coeff6[[3]] == 0, ĉ_n]/.
          {sub10[[1]], sub11[[1]]}, {3}][[1]]
```

Out[66]= $\Big\{\hat{c}_n \to -\Big(a\,n\,\big(-4\,c\,n^2\,a^2 - 4\,c\,a^2 + 8\,c\,n\,a^2 + b^2\,a - e^2\,a + b^2\,n^2\,a + 4\,c\,d\,a - 2\,b^2\,n\,a - 4\,c\,d\,n\,a - c\,d^2 - b^2\,d + b\,d\,e + b^2\,d\,n\big)\,k_n\Big)\Big/ \Big((2\,n\,a - 3\,a + d)\,(2\,n\,a - 2\,a + d)^2\,(2\,n\,a - a + d)\,k_{n-1}\Big)\Big\}$

□

Als eine Folge der Strukturformel (29) finden wir nun explizite Darstellungen für die Stammfunktionen klassischer OPS.

Satz 4.13 (Stammfunktion klassischer OPS)
Sei $P_n(x) = k_n x^n + \ldots (n \in \mathbb{N}_{\geq 0})$ ein klassisches OPS, welches die Eigenwertdifferentialgleichung mit einem quadratischen Polynom $\sigma(x)$ und einem linearen Polynom $\tau(x)$ erfüllt. Dann hat die Stammfunktion von $P_n(x)$ die Darstellung:

$$\int P_n(x)dx = \widehat{a}_n P_{n+1}(x) + \widehat{b}_n P_n(x) + \widehat{c}_n P_{n-1}(x)$$

Für die klassischen OPS ergibt sich insbesondere

$$\int H_n(x)dx = \frac{1}{2(n+1)} H_{n+1}(x),$$

$$\int L_n^{(\alpha)}(x)dx = L_n^{(\alpha)}(x) - L_{n+1}^{(\alpha)}(x),$$

$$\begin{aligned}\int B_n^{(\alpha)}dx &= \frac{2(n+1+\alpha)}{(n+1)(2n+\alpha+1)(2n+\alpha+2)} B_{n+1}^{(\alpha)}(x)\\ &+ \frac{4}{(2n+\alpha)(2n+\alpha+2)} B_n^{(\alpha)}(x)\\ &+ \frac{2n}{(n+\alpha)(2n+\alpha)(2n+\alpha+1)} B_{n-1}^{(\alpha)}(x),\end{aligned}$$

$$\begin{aligned}\int P_n^{(\alpha,\beta)}(x)dx &= \frac{2(n+\alpha+\beta+1)}{(2n+\alpha+\beta+1)(2n+\alpha+\beta+2)}P_{n+1}^{(\alpha,\beta)}(x)\\ &+ \frac{2(\alpha-\beta)}{(2n+\alpha+\beta)(2n+\alpha+\beta+2)}P_n^{(\alpha,\beta)}(x)\\ &- \frac{2(n+\alpha)(n+\beta)}{(n+\alpha+\beta)(2n+\alpha+\beta)(2n+\alpha+\beta+1)}P_{n-1}^{(\alpha,\beta)}(x).\end{aligned}$$

Beweis:
Integrieren wir die Strukturformel (29), so liefert dies die Stammfunktion. Setzen wir dann die Daten für die jeweiligen Polynomsysteme ein, so erhalten wir die obigen Darstellungen. □

4.4.3 Rodriguesformel

Als Nächstes wollen wir die Rodriguesformel betrachen, welche eine wichtige Charakerisierung der klassischen orthogonalen Polynome darstellt.

Satz 4.14 (Rodriguesformel)
Sei $P_n(x) = k_n x^n + \ldots (n \in \mathbb{N}_{\geq 0})$ ein klassisches OPS, welches die Eigenwertdifferentialgleichung mit einem quadratischen Polynom $\sigma(x)$ und einem linearen Polynom $\tau(x)$ erfüllt. Dann hat $P_n(x)$ die Darstellung

$$P_n(x) = \frac{E_n}{w(x)} \frac{d^n}{dx^n} \left(w(x)\sigma(x)^n \right), \tag{32}$$

welche die Rodriguesformel für $P_n(x)$ genannt wird.
Umgekehrt sind die durch die Rodriguesformel gegebenen Funktionen für quadratisches $\sigma(x)$ und lineares $\tau(x)$ immer klassische OPS.

Beweis:
Wir definieren
$$Q_n(x) := \frac{1}{w(x)} \frac{d^n}{dx^n} \left(w(x)\sigma(x)^n \right).$$
Um die Korrektheit der Rodriguesformel zu beweisen, müssen wir zeigen, dass es sich bei $Q_n(x)$ um Polynome vom Grad n handelt und dass diese bzgl. $w(x)$ orthogonal sind. Die Eindeutigkeit folgt daraus, dass die OPS bis auf Standardisierung gleich sind.
Zuerst wollen wir nun überlegen, wann $Q_n(x)$ ein Polynom ist. Dazu betrachten wir den Fall $n = 1$. Es soll

$$Q_1(x) = \frac{1}{w(x)} \left(w(x)\sigma(x) \right)' = \frac{w'(x)}{w(x)}\sigma(x) + \sigma'(x) = Ax + B$$

ein lineares Polynom sein. Dann erhalten wir für geeignete $\alpha, \beta \in \mathbb{R}$ mittels Partialbruchzerlegung
$$\frac{w'(x)}{w(x)} = \frac{Ax + B - \sigma'(x)}{\sigma(x)} = \frac{\alpha}{x - x_1} + \frac{\beta}{x - x_2},$$

wobei $x_1, x_2 \in \mathbb{R}$ die Nullstellen von $\sigma(x)$ sind. Integration liefert dann die Gewichtsfunktion der Jacobipolynome. Betrachten wir auf analoge Weise die Fälle, dass $\sigma(x)$ linear oder konstant ist, so erhalten wir die linear transformierten Gewichtsfunktionen der Laguerre- bzw. Hermitepolynome. Die Funktionen die wir für quadratisches $\sigma(x)$ und lineares $\tau(x)$ aus der Rodriguesformel erhalten sind also immer klassische OPS.
Wir wollen nun zeigen, dass es sich bei $Q_n(x)$ um ein Polynom vom Grad n handelt. Dazu setzen wir voraus, dass $\sigma(x)$ ein Polynom vom Grad ≤ 2 und $\tau(x)$ ein Polynom vom Grad ≤ 1 ist. Weiterhin sei $w(x)$ eine Gewichtsfunktion, welche die Pearsonsche Differentialgleichung erfüllt. Wir werden den Nachweis des Grades für $n = 0, 1, 2\dots$ führen. Den allgemeinen Fall erhalten wir dann induktiv.
Für $n = 0$ ist nichts zu zeigen.
Für $n = 1$ erhalten wir

$$Q_1(x) = \frac{1}{w(x)} (w(x)\sigma(x))' = \tau(x),$$

also ein lineares Polynom.
Für $n = 2$ erhalten wir

$$\begin{aligned}
w(x)Q_2(x) &= \left(w(x)\sigma(x)^2\right)'' = \left((w(x)\sigma(x))'\,\sigma(x) + w(x)\sigma(x)\sigma'(x)\right)' \\
&= \left(w(x)\sigma(x)\left(\tau(x) + \sigma'(x)\right)\right)' \\
&= (w(x)\sigma(x))'\left(\tau(x) + \sigma'(x)\right) + w(x)\sigma(x)\left(\tau(x) + \sigma'(x)\right)' \\
&= w(x)\tau(x)\left(\tau(x) + \sigma'(x)\right) + w(x)\sigma(x)\left(\tau(x) + \sigma'(x)\right)' \\
&= w(x)\Big(\underbrace{\tau(x)\left(\tau(x) + \sigma'(x)\right)}_{linear*linear} + \sigma(x)\left(\tau(x) + \sigma'(x)\right)'\Big).
\end{aligned}$$

Somit ist $Q_2(x)$ ein quadratisches Polynom. Leiten wir n-mal ab und wenden n-mal die Pearsonsche Differentialgleichung an, so enthält auch $w(x)Q_n(x)$ den Faktor $w(x)$. Der n-te Koeffizient ist immer ein Produkt aus Termen der Form $d + na, (n \in \mathbb{N})$, welche wegen Satz 4.2 ungleich Null sind. Daraus folgt, dass $Q_n(x)$ wirklich ein Polynom vom Grad n ist.
Wir müssen nun noch zeigen, dass die Polynome $Q_n(x)$ bzgl. $w(x)$ orthogonal sind. Dazu genügt es zu zeigen, dass $Q_n(x)$ orthogonal zu x^m ist für $m < n$. Denn wegen Satz 3.1 kann man x^n als Linearkombination darstellen. Mit n-facher partieller Integration erhalten wir

$$\int_s^t Q_n(x)x^m w(x)dx = \int_s^t \frac{d^n}{dx^n} (w(x)\sigma(x)^n)\, x^m dx = (-1)^n \int_s^t (x^m)^{(n)} w(x)\sigma(x)^n dx.$$

Wegen $m < n$ gilt $(x^m)^{(n)} = 0$ und somit

$$\int_s^t Q_n(x)x^m w(x)dx = 0,$$

daraus folgt dann die Behauptung. □

Als Nächstes wollen wir noch einige Aussagen zu den Standardisierungskoeffizienten der Rodriguesformel betrachten.

Satz 4.15 (Standardisierungskoeffizienten der Rodriguesformel)
Sei $P_n(x) = k_n x^n + ...(n \in \mathbb{N}_{\geq 0})$ ein klassisches OPS, welches die Eigenwertdifferentialgleichung mit einem quadratischen Polynom $\sigma(x)$ und einem linearen Polynom $\tau(x)$ erfüllt. Die Standardisierungskoeffizienten E_n der Rodriguesformel

$$P_n(x) = \frac{E_n}{w(x)} \frac{d^n}{dx^n} (w(x)\sigma(x)^n),$$

haben das Termverhältnis

$$\frac{E_{n+1}}{E_n} = \frac{(a(n-1)+d)}{(a(2n-1)+d)(2an+d)} \frac{k_{n+1}}{k_n}.$$

Für $a \neq 0$ erhalten wir für E_n selbst

$$E_n = \frac{k_n}{a^n (n-1+\frac{d}{a})_n}.$$

<u>**Beweis:**</u>
Sei $y(x)$ eine Lösung der Eigenwertdifferentialgleichung mit quadratischem $\sigma(x)$, linearem $\tau(x)$ und konstantem $\lambda_n \in \mathbb{R}$. Durch n-maliges Ableiten erhalten wir wegen Satz 4.10 mit $v_n(x) = y^{(n)}(x)$ das klassische OPS

$$\sigma(x)v_n''(x) + \tau_n v_n'(x) + \rho_n v(x) = 0.$$

Für die Koeffizienten $\tau_n(x)$ und ρ_n gilt dann

$$\tau_n(x) = \tau(x) + n\sigma'(x),$$

und

$$\rho_n = \lambda + n\tau' + \frac{1}{2}n(n-1)\sigma''.$$

Setzen wir $\rho_n = 0$, so erhalten wir

$$\lambda = \lambda_n = -n\tau' - \frac{1}{2}n(n-1)\sigma''.$$

Als Nächstes betrachten wir die selbstadjungierten Formen (17)

$$(\sigma(x)w(x)y(x))' + \lambda_n w(x)y(x) = 0$$

und

$$\left(\sigma(x)w_n(x)v_n(x)\right)' + \lambda_n w_n(x)v_n(x) = 0$$

für $y(x)$ und $v_n(x)$ mit den zugehörigen Gewichtfunktionen $w(x), w_n(x)$.
Weiterhin erfüllen $w(x)$ und $w_n(x)$ die Pearsonsche Differentialgleichung mit

$$(\sigma(x)w(x))' = \tau(x)w(x)$$

und

$$\left(\sigma(x)w_n(x)\right)' = \tau_n(x)w_n(x).$$

Mit der Darstellung von $\tau_n(x)$ können wir nun eine Verbindung zwischen $w_n(x)$ und $w_0(x) := w(x)$ herstellen mittels

$$\frac{\left(\sigma(x)w_n(x)\right)'}{w_n(x)} = \tau_n(x) = \tau(x) + n\sigma'(x) = \frac{(\sigma(x)w(x))'}{w(x)} + n\sigma'(x)$$

$$\Rightarrow \frac{\left(\sigma(x)w_n(x)\right)'}{w_n(x)} = \frac{(\sigma(x)w(x))'}{w(x)} + n\sigma'(x)$$

Lösen dieser Differentialgleichung liefert

$$w_n(x) = \sigma^n(x)w(x), \quad n = 0, 1, \ldots$$

Damit gilt dann auch $\sigma(x)w_n(x) = w_{n+1}(x)$. Mit diesem Resultat und $v_n'(x) = (y^{(n)})'(x) = y^{(n+1)}(x) = v_{n+1}(x)$ erhalten wir nach Umformung der selbstadjungierten Form bzgl. $w_n(x)$ die Darstellung

$$w_n(x)v_n(x) = -\frac{1}{\rho_n}\left(v_{n+1}(x)w_{n+1}(x)\right)'.$$

Hieraus erhalten wir sukzessive

$$w(x)y(x) = w_0(x)v_0(x) = -\frac{1}{\rho_0}\left(v_1(x)w_1(x)\right)' = -\frac{1}{\rho_0}\frac{1}{\rho_1}\left(v_2(x)w_2(x)\right)'' = \ldots = \frac{1}{A_n}\left(v_n(x)w_n(x)\right)^{(n)}$$

mit

$$A_n = (-1)^n \prod_{k=0}^{n-1}\rho_k, \quad A_0 = 1.$$

Dies liefert uns

$$y(x) = \frac{1}{A_n w(x)}\left(\sigma^n(x)w(x)y^{(n)}(x)\right)^{(n)}.$$

Ist $y(x)$ ein Polynom vom Grad n, so erhalten wir $y^{(n)}(x) = k_n n(n-1)(n-2)\ldots 3\cdot 2\cdot 1 = k_n n!$. Daraus folgt

$$y_n(x) = P_n(x) = \frac{E_n}{w(x)}\,(\sigma^n(x)w(x))^{(n)}$$

mit $E_n = A_n^{-1}P_n^{(n)}(x)$.
Einsetzen von ρ_k in A_n liefert uns zusammen mit *Mathematica*

In[67]:= $\mathbf{En = \dfrac{k_n\, n!}{(-1)\hat{}n\, \prod_{k=0}^{n-1}\left(-n*d - \frac{1}{2}n(n-1)2a + k*d + \frac{1}{2}k(k-1)2a\right)};}$

In[68]:= $\mathbf{\dfrac{En/.\, n \to n+1}{En}//FunctionExpand}$

Out[68]= $\dfrac{(n\,a - a + d)\,k_{n+1}}{(d + 2\,a\,n)\,(2\,n\,a - a + d)\,k_n}$

also das gesuchte Termverhältnis $\dfrac{E_{n+1}}{E_n}$. Nun verifizieren wir noch die Gültigkeit der Darstellung für E_n

In[69]:= $\mathbf{Bn = \dfrac{k_n}{a\hat{}n\, Pochhammer\left[n - 1 + \frac{d}{a}, n\right]};}$

```
In[70]:= (Bn/.n → n+1)/Bn //FunctionExpand
```

$$\texttt{Out[70]=}\ \frac{(n\,a - a + d)\,k_{n+1}}{(d + 2\,a\,n)\,(2\,n\,a - a + d)\,k_n}$$

Die Termverhältnisse sind identisch. Daraus folgt die Gültigkeit der Darstellung von E_n.

□

4.5 Verbindungskoeffizienten

In diesem Abschnitt wollen wir uns überlegen, wie Informationen, die für ein Polynomsystem $(Q_m(x))_{m\geq 0}$ gegeben sind, auf ein anderes Polynomsystem $(P_n(x))_{n\geq 0}$ transferiert werden können. Seien dazu $P_n(x)$ und $Q_n(x)$ Polynomsysteme vom Grad n, welche aufgrund der linearen Unabhängigkeit eine Beziehung der Form

$$P_n(x) = \sum_{m=0}^{n} C_m(n) Q_m(x) \tag{33}$$

besitzen. Die Koeffizienten $C_m(n) (n \in \mathbb{N}_{\geq 0}, m = 0, ...n)$ heißen Verbindungskoeffizienten zwischen $P_n(x)$ und $Q_m(x)$. Im Folgenden sei $C_m(n)$ nur für ganzzahlige m, n erklärt und es gelte $C_m(n) = 0$ außerhalb der $m \times n$-Region. Weiterhin beschränken wir uns auf Polynomsysteme $\widetilde{P}_n(x)$ und $\widetilde{Q}_m(x)$, bei denen die höchsten Koeffizienten 1 sind. Das folgern wir so: Wenn k_n und $\overline{k}_m$ die höchsten Koeffizienten von $P_n(x)$ und $Q_m(x)$ sind sowie $\widetilde{C}_m(n)$ die Verbindungskoeffizienten zwischen $\widetilde{P}_n(x)$ und $\widetilde{Q}_m(x)$ mit

$$\widetilde{P}_n(x) = \sum_{m=0}^{n} \widetilde{C}_m(n) \widetilde{Q}_m(x),$$

dann gilt zwischen $C_m(n)$ und $\widetilde{C}_m(n)$ die Beziehung

$$C_m(n) = \frac{k_n}{\overline{k}_m} \widetilde{C}_m(n).$$

Im Folgenden wollen wir für $C_m(n)$ Rekursionsgleichungen herleiten, da diese in einigen Fällen hypergeometrische Terme als Lösungen besitzen. Weil $C_m(n)$ jedoch von m und n abhängig ist, werden wir im ersten Schritt gemischte Rekursionsgleichungen erhalten. Ziel ist es dann, im Weiteren reine Rekursionsgleichungen bzgl. m und n für die Verbindungskoeffizienten zu finden.
Sei nun $P_n(x)$ ein OPS, welches die Eigenwertdifferentialgleichung erfüllt mit $\sigma(x) = ax^2 + bx + c$ und $\tau(x) = dx + e$ und sei $Q_m(x)$ ein OPS, welches die Eigenwertdifferentialgleichung erfüllt mit $\overline{\sigma}(x) = \overline{a}x^2 + \overline{b}x + \overline{c}$ und $\overline{\tau}(x) = \overline{d}x + \overline{e}$.
Dann erfüllen beide OPS die Rekursion (24) mit den Koeffizienten (25). Die zu $Q_m(x)$ gehörigen Koeffizienten werden im Folgenden durch Überstriche notiert. Wir erhalten also zusammen mit (24)

$$xP_n(x) = a_n P_{n+1}(x) + b_n P_n(x) + c_n P_{n-1}(x)$$

und

$$xQ_m(x) = \overline{a}_m Q_{m+1}(x) + \overline{b}_m Q_m(x) + \overline{c}_m Q_{m-1}(x),$$

wobei alle Koeffizenten $a_n, b_n, c_n, \overline{a}_m, \overline{b}_m, \overline{c}_m$ explizit gegeben sind.
Als Nächstes wollen wir unabhängige Rekursionsgleichungen für die Verbindungskoeffizienten $C_m(n)$ herleiten. Dazu betrachten wir $xP_n(x)$. Anwenden von (33) auf $xP_n(x)$ liefert

$$xP_n(x) = \sum_{m=0}^{n} xC_m(n)Q_m(x) = \sum_{m=0}^{n} C_m(n)\left(\overline{a}_m Q_{m+1}(x) + \overline{b}_m Q_m(x) + \overline{c}_m Q_{m-1}(x)\right).$$

Als Nächstes betrachten wir $a_nP_{n+1}(x) + b_nP_n(x) + c_nP_{n-1}(x)$. Anwenden von (33) liefert

$$a_nP_{n+1}(x)+b_nP_n(x)+c_nP_{n-1}(x) = \sum_{m=0}^{n}\left(a_nC_m(n+1)Q_m(x) + b_nC_m(n)Q_m(x) + c_nC_m(n-1)Q_m(x)\right).$$

Wegen $xP_n(x) = a_nP_{n+1}(x) + b_nP_n(x) + c_nP_{n-1}(x)$ erhalten wir dann

$$\sum_{m=0}^{n}\left(a_nC_m(n+1)Q_m(x) + b_nC_m(n)Q_m(x) + c_nC_m(n-1)Q_m(x)\right) =$$

$$\sum_{m=0}^{n} C_m(n)\left(\overline{a}_m Q_{m+1}(x) + \overline{b}_m Q_m(x) + \overline{c}_m Q_{m-1}(x)\right).$$

Indexverschiebung und Koeffizientenvergleich liefert uns dann die *erste Kreuzregel*

$$a_nC_m(n+1) + b_nC_m(n) + c_nC_m(n-1) = \overline{a}_{m-1}C_{m-1}(n) + \overline{b}_mC_m(n) + \overline{c}_{m+1}C_{m+1}(n). \qquad (34)$$

Um die zweite Kreuzregel zu erhalten, die von denselben Variablen $C_m(n+1), C_m(n), C_m(n-1), C_{m-1}(n)$ und $C_{m+1}(n)$ wie die erste Kreuzregel abhängt, betrachten wir den Term $xP_n'(x)$. Für $xP_n'(x)$ gelten dann Rekursionsgleichungen

$$xP_n'(x) = \alpha_n^* P_{n+1}'(x) + \beta_n^* P_n'(x) + \gamma_n^* P_{n-1}'(x)$$

und

$$xQ_m'(x) = \overline{\alpha}_m^* Q_{m+1}'(x) + \overline{\beta}_m^* Q_m'(x) + \overline{\gamma}_m^* Q_{m-1}'(x).$$

Mit den obigen Rekursionsgleichungen und (33) erhalten wir dann

$$\begin{aligned}
xP_n'(x) &= \alpha_n^* P_{n+1}'(x) + \beta_n^* P_n'(x) + \gamma_n^* P_{n-1}'(x)\\
&= \sum_{m=0}^{n}\left(\alpha_n^* C_m(n+1)Q_m'(x) + \beta_n^* C_m(n)Q_m'(x) + \gamma_n^* C_m(n-1)Q_m'(x)\right)\\
&= \sum_{m=0}^{n} C_m(n)xQ_m'(x)\\
&= \sum_{m=0}^{n} C_m(n)\left(\overline{\alpha}_m^* Q_{m+1}'(x) + \overline{\beta}_m^* Q_m'(x) + \overline{\gamma}_m^* Q_{m-1}'(x)\right).
\end{aligned}$$

Also

$$\sum_{m=0}^{n}\left(\alpha_n^* C_m(n+1)Q_m'(x) + \beta_n^* C_m(n)Q_m'(x) + \gamma_n^* C_m(n-1)Q_m'(x)\right) =$$

$$\sum_{m=0}^{n} C_m(n)\left(\overline{\alpha}^*_m Q'_{m+1}(x) + \overline{\beta}^*_m Q'_m(x) + \overline{\gamma}^*_m Q'_{m-1}(x)\right).$$

Durch Indexverschiebung und Koeffizientenvergleich erhalten wir dann die *zweite Kreuzregel*

$$\alpha^*_n C_m(n+1) + \beta^*_n C_m(n) + \gamma^*_m C_m(n-1) = \overline{\alpha}^*_{m-1} C_{m-1}(n) + \overline{\beta}^*_m C_m(n) + \overline{\gamma}^*_{m+1} C_{m+1}(n). \quad (35)$$

Analog kann man aus der Strukturformel die Kreuzregel

$$\widehat{\alpha}_n C_m(n+1) + \widehat{b}_n C_m(n) + \widehat{c}_n C_m(n-1) = \overline{\overline{\widehat{a}}}_{m-1} C_{m-1}(n) + \overline{\overline{\widehat{b}}}_m C_m(n) + \overline{\overline{\widehat{c}}}_{m+1} C_{m+1}(n). \quad (36)$$

herleiten. Diese ist aber linear von der ersten und zweiten Kreuzregel abhängig und liefert somit keine neuen Resultate.
Zur Herleitung einer dritten unabhängigen Kreuzregel betrachten wir den Spezialfall $\sigma(x) = \overline{\sigma}(x)$. Mit der Ableitungsregel (26) erhalten wir die beiden Rekursionsgleichungen

$$\sigma(x)P'_n(x) = \alpha_n P_{n+1}(x) + \beta_n P_n(x) + \gamma_n P_{n-1}(x)$$

und

$$\overline{\sigma}(x)Q'_m(x) = \overline{\alpha}_m Q_{m+1}(x) + \overline{\beta}_m Q_m(x) + \overline{\gamma}_m Q_{m-1}(x) = \sigma(x)Q'_m(x).$$

Aus den Rekursionsgleichungen und (33) folgt dann

$$\begin{aligned}
\sigma(x)P'_n(x) &= \alpha_n P_{n+1}(x) + \beta_n P_n(x) + \gamma_n P_{n-1}(x) \\
&= \sum_{m=0}^{n} \left(\alpha_n C_{n+1}(n+1)Q_m(x) + \beta_n C_m(n)Q_m(x) + \gamma_n C_m(n-1)Q_m(x)\right) \\
&= \sum_{m=0}^{n} C_m(n)\sigma(x)Q'_m(x) \\
&= \sum_{m=0}^{n} C_m(n)\left(\overline{\alpha}_m Q_{m+1}(x) + \overline{\beta}_m Q_m(x) + \overline{\gamma}_m Q_{m-1}(x)\right).
\end{aligned}$$

Also

$$\sum_{m=0}^{n} \left(\alpha_n C_{n+1}(n+1)Q_m(x) + \beta_n C_m(n)Q_m(x) + \gamma_n C_m(n-1)Q_m(x)\right) =$$

$$\sum_{m=0}^{n} C_m(n)\left(\overline{\alpha}_m Q_{m+1}(x) + \overline{\beta}_m Q_m(x) + \overline{\gamma}_m Q_{m-1}(x)\right)$$

Durch Indexverschiebung und Koeffizientenvergleich erhalten wir dann die *dritte Kreuzregel*

$$\alpha_n C_m(n+1) + \beta_n C_m(n) + \gamma_n C_m(n-1) = \overline{\alpha}_{m-1} C_{m-1}(n) + \overline{\beta}_m C_m(n) + \overline{\gamma}_{m+1} C_{m+1}(n). \quad (37)$$

Um nun reine Rekursionsgleichungen bzgl. n und m zu erhalten, müssen wir mittels linearer Algebra aus den drei Kreuzregeln die Variablen $C_{m-1}(n)$ und $C_{m+1}(n)$ bzw. $C_m(n+1)$ und $C_m(n-1)$ eliminieren. Dies liefert dann den folgenden Satz:

Satz 4.16 (Verbindungskoeffizienten klassischer OPS)
Sei $P_n(x) = x^n + \dots \quad (n \in \mathbb{N}_{\geq 0})$ ein klassisches OPS mit dem höchsten Koeffizienten $k_n = 1$, welches die Eigenwertdifferentialgleichung mit einem quadratischen Polynom $\sigma(x) = ax^2 + bx + c$ und einem linearen Polynom $\tau(x) = dx + e$ erfüllt. Sei ferner $Q_m(x) = x^m + \dots \quad (m \in \mathbb{N}_{\geq 0})$ ebenfalls ein klassisches OPS mit dem höchsten Koeffizienten $\overline{k}_m = 1$, welches die Eigenwertdifferentialgleichung mit einem quadratischen Polynom $\overline{\sigma}(x) = \sigma(x)$ und einem linearen Polynom $\overline{\tau}(x) = \overline{d}x + \overline{e}$ erfüllt. Dann gilt die Beziehung (33), wobei $C_m(n)$ die Dreitermrekursion

$$\begin{gathered}
-(m-n)(am+d-a+an)(\overline{d}+2am)(\overline{d}+a+2am)(\overline{d}+3a+2am) \\
(\overline{d}+2am+2a)^2C_m(n) + (-dbn\overline{d} + 2dam^2b + db\overline{d} + 2damb \\
+ 2d\overline{e}na + d\overline{d}\overline{e} + 2d\overline{d}bm - mb\overline{d}^2 - e\overline{d}^2 - 4a^2m^2e - m^2ab\overline{d} + bn\overline{d}a - 2e\overline{d}a \\
- 4a^2me - 4e\overline{d}am + 2m^2a^2\overline{e} + 2\overline{e}a^2n^2 - 2\overline{e}a^2n - mab\overline{d} + 2m\overline{d}\overline{e}a + 2m\overline{e}a^2 \\
- bn^2\overline{d}a)(\overline{d}+2am+2a)(m+1)(\overline{d}+a+2am)(\overline{d}+3a+2am)C_{m+1}(n) - (\overline{d}+2am)(m+1) \\
(-am-2a+an-\overline{d}+d)(am+an+a+\overline{d})(ab^2m^2 - 4a^2m^2c - 8a^2mc + 2amb^2 - 4a\overline{d}mc \\
+ mb^2\overline{d} - 4a\overline{d}c - a\overline{e}^2 + ab^2 - c\overline{d}^2 + b\overline{e}\overline{d} - 4a^2c + b^2\overline{d})(m+2)C_{m+2}(n) = 0
\end{gathered}$$

mit Anfangswerten $C_n(n) = 1, C_{n+1}(n) \equiv 0$ bzgl. m erfüllt. Außerdem gilt die Dreitermrekursion

$$\begin{gathered}
-(d+2an)^2(d-a+2an)(d+2an+2a)(d+a+2an)(-m+n+2) \\
(\overline{d}+am+a+an)C_m(n+2) + (d-a+2an)(d+a+2an)(n+2)(d+2an) \\
(-2\overline{e}ad - bd^2n - 2ma^2e + 2m^2a^2e + 2a^2en - 4\overline{e}a^2n^2 + 2\overline{d}bd - 2\overline{d}ea \\
+ mbda - bd^2 + 2eda - 4\overline{e}adn + 2edan - abdn^2 - bdna + 2a^2en^2 - m^2abd \\
- 4\overline{e}a^2n - \overline{d}mbd + 2\overline{d}anb + 2\overline{d}an^2b + 2\overline{d}dbn + 2\overline{d}mea \\
- \overline{e}d^2 + \overline{d}de)C_m(n+1) + (d+2an+2a)(n+2)(n+1)(an-am+d-\overline{d}) \\
(an+am-a+d)(bed - ae^2 - d^2c - 4acnd - 4a^2cn^2 + ab^2n^2 + nb^2d)C_m(n) = 0
\end{gathered}$$

bzgl. n.

Beweis:
Zuerst müssen wir die Koeffizienten α^*, β^* und γ^* der zweiten Kreuzregel herleiten. Diese erhalten wir aus der Rekursion (30), welche wir analog wie die anderen Koeffizienten zuvor mit Mathematica herleiten. Wir geben also die Rekursionsformel ein und teilen diese durch x^{n-4}

```
In[71]:= rekgl2 = αsn D[pn+1, x] + βsn D[pn, x] + γsn D[pn-1, x] - x D[pn, x];
```

```
In[72]:= coeff7 = CoefficientList[rekgl2/x^(n - 4), x];
```

Mit Koeffizientenvergleich erhalten wir dann die Koeffizienten.

```
In[73]:= sub13 = Solve[coeff7[[5]] == 0, αsn][[1]]
```

Out[73]= $\left\{\alpha s_n \to \frac{n\, k_n}{(n+1)\, k_{n+1}}\right\}$

```
In[74]:= sub14 = Map[Factor, Solve[coeff7[[4]] == 0, βsn]/.sub13, {3}][[1]]
```

Out[74]= $\left\{\beta s_n \to -\frac{2\,a\,b\,n^2 - 2\,a\,b\,n + 2\,b\,d\,n - b\,d + d\,e}{(d + 2\,a\,n)\,(2\,n\,a - 2\,a + d)}\right\}$

```
In[75]:= sub15 =
           Map[Factor, Solve[coeff7[[3]] == 0, γsn]/.
               {sub13[[1]], sub14[[1]]}, {3}][[1]]
```

Out[75]= $\Big\{\gamma s_n \to \Big(n\,(n\,a - a + d)\,\big(-4\,c\,n^2\,a^2 - 4\,c\,a^2 + 8\,c\,n\,a^2 + b^2\,a - e^2\,a + b^2\,n^2\,a + 4\,c\,d\,a - 2\,b^2\,n\,a - 4\,c\,d\,n\,a - c\,d^2 - b^2\,d + b\,d\,e + b^2\,d\,n\big)\,k_n\Big)\Big/\Big((2\,n\,a - 3\,a + d)\,(2\,n\,a - 2\,a + d)^2\,(2\,n\,a - a + d)\,k_{n-1}\Big)\Big\}$

Als Nächstes definieren wir nun die drei Kreuzregeln.

```
In[76]:= cr1 = cr1 = anCm[n+1] + bnCm[n] + cnCm[n-1] ==
             oam-1Cm-1[n] + obmCm[n] + ocm+1Cm+1[n];
```

```
In[77]:= cr2 = αsnCm[n+1] + βsnCm[n] + γsnCm[n-1] ==
           oαsm-1Cm-1[n] + oβsmCm[n] + oγsm+1Cm+1[n];
```

```
In[78]:= cr3 = αnCm[n+1] + βnCm[n] + γnCm[n-1] ==
           oαm-1Cm-1[n] + oβmCm[n] + oγm+1Cm+1[n];
```

Nun müssen wir $C_m(n-1)$ und $C_m(n+1)$ sowie $C_{m-1}(n)$ und $C_{m+1}(n)$ eliminieren. Dies liefert uns die beiden neuen Rekursionen

```
In[79]:= rek1 = Eliminate[{cr1, cr2, cr3}, {Cm[n+1], Cm[n-1]}]
```

Out[79]= $-c_n\,\mathrm{o}\alpha\mathrm{s}_{m-1}\,\alpha_n\,\mathrm{C}_{m-1}(n) + c_n\,\mathrm{o}\alpha_{m-1}\,\alpha\mathrm{s}_n\,\mathrm{C}_{m-1}(n) +$
$a_n\,\mathrm{o}\alpha\mathrm{s}_{m-1}\,\gamma_n\,\mathrm{C}_{m-1}(n) - a_n\,\mathrm{o}\alpha_{m-1}\,\gamma\mathrm{s}_n\,\mathrm{C}_{m-1}(n) - c_n\,\mathrm{o}\beta\mathrm{s}_m\,\alpha_n\,\mathrm{C}_m(n) +$
$c_n\,\mathrm{o}\beta_m\,\alpha\mathrm{s}_n\,\mathrm{C}_m(n) - c_n\,\alpha\mathrm{s}_n\,\beta_n\,\mathrm{C}_m(n) + c_n\,\alpha_n\,\beta\mathrm{s}_n\,\mathrm{C}_m(n) + a_n\,\mathrm{o}\beta\mathrm{s}_m\,\gamma_n\,\mathrm{C}_m(n) +$
$b_n\,\alpha\mathrm{s}_n\,\gamma_n\,\mathrm{C}_m(n) - \mathrm{ob}_m\,\alpha\mathrm{s}_n\,\gamma_n\,\mathrm{C}_m(n) - a_n\,\beta\mathrm{s}_n\,\gamma_n\,\mathrm{C}_m(n) - a_n\,\mathrm{o}\beta_m\,\gamma\mathrm{s}_n\,\mathrm{C}_m(n) -$
$b_n\,\alpha_n\,\gamma\mathrm{s}_n\,\mathrm{C}_m(n) + \mathrm{ob}_m\,\alpha_n\,\gamma\mathrm{s}_n\,\mathrm{C}_m(n) + a_n\,\beta_n\,\gamma\mathrm{s}_n\,\mathrm{C}_m(n) - c_n\,\mathrm{o}\gamma\mathrm{s}_{m+1}\,\alpha_n\,\mathrm{C}_{m+1}(n) +$
$c_n\,\mathrm{o}\gamma_{m+1}\,\alpha\mathrm{s}_n\,\mathrm{C}_{m+1}(n) + a_n\,\mathrm{o}\gamma\mathrm{s}_{m+1}\,\gamma_n\,\mathrm{C}_{m+1}(n) - \mathrm{oc}_{m+1}\,\alpha\mathrm{s}_n\,\gamma_n\,\mathrm{C}_{m+1}(n) -$
$a_n\,\mathrm{o}\gamma_{m+1}\,\gamma\mathrm{s}_n\,\mathrm{C}_{m+1}(n) + \mathrm{oc}_{m+1}\,\alpha_n\,\gamma\mathrm{s}_n\,\mathrm{C}_{m+1}(n) == \mathrm{oa}_{m-1}\,(\alpha\mathrm{s}_n\,\gamma_n - \alpha_n\,\gamma\mathrm{s}_n)\,\mathrm{C}_{m-1}(n)$

und

```
In[80]:= rek2 = Eliminate[{cr1, cr2, cr3}, {Cm+1[n], Cm-1[n]}]
```

Out[80]= $-\mathrm{oc}_{m+1}\,\mathrm{o}\alpha\mathrm{s}_{m-1}\,\gamma_n\,\mathrm{C}_m(n-1) + \mathrm{oa}_{m-1}\,\mathrm{o}\gamma\mathrm{s}_{m+1}\,\gamma_n\,\mathrm{C}_m(n-1) + \mathrm{oc}_{m+1}\,\mathrm{o}\alpha_{m-1}\,\gamma\mathrm{s}_n\,\mathrm{C}_m(n-1) -$
$\mathrm{oa}_{m-1}\,\mathrm{o}\gamma_{m+1}\,\gamma\mathrm{s}_n\,\mathrm{C}_m(n-1) + \mathrm{oc}_{m+1}\,\mathrm{o}\alpha\mathrm{s}_{m-1}\,\mathrm{o}\beta_m\,\mathrm{C}_m(n) - \mathrm{oc}_{m+1}\,\mathrm{o}\alpha_{m-1}\,\mathrm{o}\beta\mathrm{s}_m\,\mathrm{C}_m(n) +$
$b_n\,\mathrm{o}\alpha\mathrm{s}_{m-1}\,\mathrm{o}\gamma_{m+1}\,\mathrm{C}_m(n) - \mathrm{ob}_m\,\mathrm{o}\alpha\mathrm{s}_{m-1}\,\mathrm{o}\gamma_{m+1}\,\mathrm{C}_m(n) + \mathrm{oa}_{m-1}\,\mathrm{o}\beta\mathrm{s}_m\,\mathrm{o}\gamma_{m+1}\,\mathrm{C}_m(n) -$
$b_n\,\mathrm{o}\alpha_{m-1}\,\mathrm{o}\gamma\mathrm{s}_{m+1}\,\mathrm{C}_m(n) + \mathrm{ob}_m\,\mathrm{o}\alpha_{m-1}\,\mathrm{o}\gamma\mathrm{s}_{m+1}\,\mathrm{C}_m(n) - \mathrm{oa}_{m-1}\,\mathrm{o}\beta_m\,\mathrm{o}\gamma\mathrm{s}_{m+1}\,\mathrm{C}_m(n) -$
$\mathrm{oc}_{m+1}\,\mathrm{o}\alpha\mathrm{s}_{m-1}\,\beta_n\,\mathrm{C}_m(n) + \mathrm{oa}_{m-1}\,\mathrm{o}\gamma\mathrm{s}_{m+1}\,\beta_n\,\mathrm{C}_m(n) + \mathrm{oc}_{m+1}\,\mathrm{o}\alpha_{m-1}\,\beta\mathrm{s}_n\,\mathrm{C}_m(n) -$
$\mathrm{oa}_{m-1}\,\mathrm{o}\gamma_{m+1}\,\beta\mathrm{s}_n\,\mathrm{C}_m(n) + a_n\,\mathrm{o}\alpha\mathrm{s}_{m-1}\,\mathrm{o}\gamma_{m+1}\,\mathrm{C}_m(n+1) - a_n\,\mathrm{o}\alpha_{m-1}\,\mathrm{o}\gamma\mathrm{s}_{m+1}\,\mathrm{C}_m(n+1) -$
$\mathrm{oc}_{m+1}\,\mathrm{o}\alpha\mathrm{s}_{m-1}\,\alpha_n\,\mathrm{C}_m(n+1) + \mathrm{oa}_{m-1}\,\mathrm{o}\gamma\mathrm{s}_{m+1}\,\alpha_n\,\mathrm{C}_m(n+1) + \mathrm{oc}_{m+1}\,\mathrm{o}\alpha_{m-1}\,\alpha\mathrm{s}_n\,\mathrm{C}_m(n+1) -$
$\mathrm{oa}_{m-1}\,\mathrm{o}\gamma_{m+1}\,\alpha\mathrm{s}_n\,\mathrm{C}_m(n+1) == c_n\,(\mathrm{o}\alpha_{m-1}\,\mathrm{o}\gamma\mathrm{s}_{m+1} - \mathrm{o}\alpha\mathrm{s}_{m-1}\,\mathrm{o}\gamma_{m+1})\,\mathrm{C}_m(n-1)$

Da wir alle auftretenden Koeffizienten bereits zuvor hergeleitet haben, können wir diese in unsere Rekursionen einsetzen. Nach einem Shift von m erhalten wir

```
In[81]:= rek3 = rek1/.m → m + 1;
         rek4 = rek2/.n → n + 1;
```

```
In[82]:= rek5 =
           Map[Factor,
             Collect[Simplify[Numerator[Together[rek3[[1]] - rek3[[2]]]]],
               {C_m[n], C_{m+1}[n], C_{m+2}[n]}]]
```

Out[82]= $-n\,(n-m)\,(m\,a+n\,a-a+d)\left(4\,c\,n^2\,a^2+4\,c\,a^2-8\,c\,n\,a^2-b^2\,a+e^2\,a-b^2\,n^2\,a-4\,c\,d\,a+2\,b^2\,n\,a+4\,c\,d\,n\,a+c\,d^2+b^2\,d-b\,d\,e-b^2\,d\,n\right)(2\,a\,m+\mathrm{od})\,(2\,m\,a+a+\mathrm{od})\,(2\,m\,a+3\,a+\mathrm{od})\,\mathrm{C}_m(n)\,(2\,m\,a+2\,a+\mathrm{od})^2-(m+1)\,n\left(4\,c\,n^2\,a^2+4\,c\,a^2-8\,c\,n\,a^2-b^2\,a+e^2\,a-b^2\,n^2\,a-4\,c\,d\,a+2\,b^2\,n\,a+4\,c\,d\,n\,a+c\,d^2+b^2\,d-b\,d\,e-b^2\,d\,n\right)(2\,m\,a+a+\mathrm{od})\,(2\,m\,a+3\,a+\mathrm{od})\left(-4\,e\,m^2\,a^2-4\,e\,m\,a^2+2\,m^2\,\mathrm{oe}\,a^2+2\,n^2\,\mathrm{oe}\,a^2+2\,m\,\mathrm{oe}\,a^2-2\,n\,\mathrm{oe}\,a^2+2\,b\,d\,m^2\,a+2\,b\,d\,m\,a-b\,m^2\,\mathrm{od}\,a-b\,n^2\,\mathrm{od}\,a-2\,e\,\mathrm{od}\,a-b\,m\,\mathrm{od}\,a-4\,e\,m\,\mathrm{od}\,a+b\,n\,\mathrm{od}\,a+2\,d\,n\,\mathrm{oe}\,a+2\,m\,\mathrm{od}\,\mathrm{oe}\,a-e\,\mathrm{od}^2-b\,m\,\mathrm{od}^2+b\,d\,\mathrm{od}+2\,b\,d\,m\,\mathrm{od}-b\,d\,n\,\mathrm{od}+d\,\mathrm{od}\,\mathrm{oe}\right)\mathrm{C}_{m+1}(n)\,(2\,m\,a+2\,a+\mathrm{od})+(m+1)\,(m+2)\,n\left(4\,c\,n^2\,a^2+4\,c\,a^2-8\,c\,n\,a^2-b^2\,a+e^2\,a-b^2\,n^2\,a-4\,c\,d\,a+2\,b^2\,n\,a+4\,c\,d\,n\,a+c\,d^2+b^2\,d-b\,d\,e-b^2\,d\,n\right)(2\,a\,m+\mathrm{od})\,(m\,a-n\,a+2\,a-d+\mathrm{od})\,(m\,a+n\,a+a+\mathrm{od})\left(4\,c\,m^2\,a^2+4\,c\,a^2+8\,c\,m\,a^2-b^2\,a-b^2\,m^2\,a+\mathrm{oe}^2\,a-2\,b^2\,m\,a+4\,c\,\mathrm{od}\,a+4\,c\,m\,\mathrm{od}\,a+c\,\mathrm{od}^2-b^2\,\mathrm{od}-b^2\,m\,\mathrm{od}-b\,\mathrm{od}\,\mathrm{oe}\right)\mathrm{C}_{m+2}(n)$

und

```
In[83]:= rek6 =
           Map[Factor,
             Collect[Simplify[Numerator[Together[rek4[[1]] - rek4[[2]]]]],
               {C_m[n + 2], C_m[n + 1], C_m[n]}]]
```

Out[83]= $(m+1)\,(m-n-2)\,(2\,n\,a-a+d)\,(2\,n\,a+a+d)\,(2\,n\,a+2\,a+d)\,(m\,a+n\,a+a+\mathrm{od})\left(-a\,m^2\,b^2-m\,\mathrm{od}\,b^2-\mathrm{od}\,\mathrm{oe}\,b+4\,a^2\,c\,m^2+c\,\mathrm{od}^2+a\,\mathrm{oe}^2+4\,a\,c\,m\,\mathrm{od}\right)\mathrm{C}_m(n+2)\,(d+2\,a\,n)^2-(m+1)\,(n+2)\,(2\,n\,a-a+d)\,(2\,n\,a+a+d)\left(-2\,e\,m^2\,a^2-2\,e\,n^2\,a^2+2\,e\,m\,a^2-2\,e\,n\,a^2+4\,n^2\,\mathrm{oe}\,a^2+4\,n\,\mathrm{oe}\,a^2+b\,d\,m^2\,a+b\,d\,n^2\,a-2\,d\,e\,a-b\,d\,m\,a+b\,d\,n\,a-2\,d\,e\,n\,a-2\,b\,n^2\,\mathrm{od}\,a+2\,e\,\mathrm{od}\,a-2\,e\,m\,\mathrm{od}\,a-2\,b\,n\,\mathrm{od}\,a+2\,d\,\mathrm{oe}\,a+4\,d\,n\,\mathrm{oe}\,a+b\,d^2+b\,d^2\,n-2\,b\,d\,\mathrm{od}-d\,e\,\mathrm{od}+b\,d\,m\,\mathrm{od}-2\,b\,d\,n\,\mathrm{od}+d^2\,\mathrm{oe}\right)\left(-a\,m^2\,b^2-m\,\mathrm{od}\,b^2-\mathrm{od}\,\mathrm{oe}\,b+4\,a^2\,c\,m^2+c\,\mathrm{od}^2+a\,\mathrm{oe}^2+4\,a\,c\,m\,\mathrm{od}\right)\mathrm{C}_m(n+1)\,(d+2\,a\,n)+(m+1)\,(n+1)\,(n+2)\,(m\,a+n\,a-a+d)\,(2\,n\,a+2\,a+d)\left(-a\,n^2\,b^2-d\,n\,b^2-d\,e\,b+c\,d^2+a\,e^2+4\,a^2\,c\,n^2+4\,a\,c\,d\,n\right)(-d+a\,m-a\,n+\mathrm{od})\left(-a\,m^2\,b^2-m\,\mathrm{od}\,b^2-\mathrm{od}\,\mathrm{oe}\,b+4\,a^2\,c\,m^2+c\,\mathrm{od}^2+a\,\mathrm{oe}^2+4\,a\,c\,m\,\mathrm{od}\right)\mathrm{C}_m(n)$

Nach Elimination der Vielfachen erhalten wir die gesuchten Rekursionen

```
In[84]:= Map[Factor,
           Collect[
             Simplify[
               rek5/PolynomialGCD[
                   Coefficient[rek5, {C_m[n], C_{m+1}[n], C_{m+2}[n]}][[1]],
                   Coefficient[rek5, {C_m[n], C_{m+1}[n], C_{m+2}[n]}][[2]],
                   Coefficient[rek5, {C_m[n], C_{m+1}[n], C_{m+2}[n]}][[3]]]],
             {C_m[n], C_{m+1}[n], C_{m+2}[n]}]]
```

Out[84]= $(m-n)\,(m\,a+n\,a-a+d)\,(2\,a\,m+\text{od})\,(2\,m\,a+a+\text{od})\,(2\,m\,a+3\,a+\text{od})$
$\text{C}_m(n)\,(2\,m\,a+2\,a+\text{od})^2+(m+1)\,(2\,m\,a+a+\text{od})\,(2\,m\,a+3\,a+\text{od})$
$\big(4\,e\,m^2\,a^2+4\,e\,m\,a^2-2\,m^2\,\text{oe}\,a^2-2\,n^2\,\text{oe}\,a^2-2\,m\,\text{oe}\,a^2+2\,n\,\text{oe}\,a^2-$
$2\,b\,d\,m^2\,a-2\,b\,d\,m\,a+b\,m^2\,\text{od}\,a+b\,n^2\,\text{od}\,a+2\,e\,\text{od}\,a+b\,m\,\text{od}\,a+$
$4\,e\,m\,\text{od}\,a-b\,n\,\text{od}\,a-2\,d\,n\,\text{oe}\,a-2\,m\,\text{od}\,\text{oe}\,a+e\,\text{od}^2+b\,m\,\text{od}^2-$
$b\,d\,\text{od}-2\,b\,d\,m\,\text{od}+b\,d\,n\,\text{od}-d\,\text{od}\,\text{oe}\big)\,\text{C}_{m+1}(n)\,(2\,m\,a+2\,a+\text{od})+$
$(m+1)\,(m+2)\,(2\,a\,m+\text{od})\,(m\,a-n\,a+2\,a-d+\text{od})\,(m\,a+n\,a+a+\text{od})$
$\big(4\,c\,m^2\,a^2+4\,c\,a^2+8\,c\,m\,a^2-b^2\,a-b^2\,m^2\,a+\text{oe}^2\,a-2\,b^2\,m\,a+$
$4\,c\,\text{od}\,a+4\,c\,m\,\text{od}\,a+c\,\text{od}^2-b^2\,\text{od}-b^2\,m\,\text{od}-b\,\text{od}\,\text{oe}\big)\,\text{C}_{m+2}(n)$

```
In[85]:= Map[Factor,
           Collect[
             Simplify[
               rek6/PolynomialGCD[
                   Coefficient[rek6, {Cm[n + 2], Cm[n + 1], Cm[n]}][[1]],
                   Coefficient[rek6, {Cm[n + 2], Cm[n + 1], Cm[n]}][[2]],
                   Coefficient[rek6, {Cm[n + 2], Cm[n + 1], Cm[n]}][[3]]]],
             {Cm[n + 2], Cm[n + 1], Cm[n]}]]
```

Out[85]= $(m-n-2)\,(2\,n\,a-a+d)\,(2\,n\,a+a+d)$
$(2\,n\,a+2\,a+d)\,(m\,a+n\,a+a+\text{od})\,\text{C}_m(n+2)\,(d+2\,a\,n)^2-$
$(n+2)\,(2\,n\,a-a+d)\,(2\,n\,a+a+d)\,\big(-2\,e\,m^2\,a^2-2\,e\,n^2\,a^2+2\,e\,m\,a^2-2\,e\,n\,a^2+$
$4\,n^2\,\text{oe}\,a^2+4\,n\,\text{oe}\,a^2+b\,d\,m^2\,a+b\,d\,n^2\,a-2\,d\,e\,a-b\,d\,m\,a+b\,d\,n\,a-$
$2\,d\,e\,n\,a-2\,b\,n^2\,\text{od}\,a+2\,e\,\text{od}\,a-2\,e\,m\,\text{od}\,a-2\,b\,n\,\text{od}\,a+2\,d\,\text{oe}\,a+$
$4\,d\,n\,\text{oe}\,a+b\,d^2+b\,d^2\,n-2\,b\,d\,\text{od}-d\,e\,\text{od}+b\,d\,m\,\text{od}-2\,b\,d\,n\,\text{od}+d^2\,\text{oe}\big)$
$\text{C}_m(n+1)\,(d+2\,a\,n)-(n+1)\,(n+2)\,(m\,a+n\,a-a+d)\,(2\,n\,a+2\,a+d)$
$\big(-a\,n^2\,b^2-d\,n\,b^2-d\,e\,b+c\,d^2+a\,e^2+4\,a^2\,c\,n^2+4\,a\,c\,d\,n\big)$
$(d-a\,m+a\,n-\text{od})\,\text{C}_m(n)$

□

In manchen Fällen können einfache Formeln für $C_m(n)$ mit hypergeometrischen Termen gefunden werden. Dies ist in den folgenden Fällen möglich: $L_n^{(\alpha)}(x)$ und $L_m^{(\beta)}(x)$, $B_n^{(\alpha)}(x)$ und $B_m^{(\beta)}(x)$, $P_n(x)=P_n^{(\alpha,\beta)}(x)$ und $Q_m(x)=P_m^{(\alpha,\delta)}(x)$, $P_n(x)=P_n^{(\alpha,\beta)}(x)$ und $Q_m(x)=P_m^{(\gamma,\beta)}(x)$, $P_n(x)=P_n^{(\alpha,\alpha)}(x)$ und $Q_m(x)=P_m^{(\beta,\beta)}(x)$.

Der letzte Fall stellt einen Spezialfall der Jacobipolynome dar, welcher uns für $\alpha=\beta=\lambda-\dfrac{1}{2}$ die sogenannten ultrasphärischen Polynome oder Gegenbauerpolynome liefert. Diese sind gegeben durch

$$\begin{aligned} C_n^{\lambda}(x) &:= \frac{(2\lambda)_n}{(\lambda+1/2)_n}P_n^{(\lambda-1/2,\lambda-1/2)}(x) \\ &= \frac{(2\lambda)_n}{n!}\,{}_2F_1\left(\left.\begin{array}{c} -n,n+2\lambda \\ \lambda+\frac{1}{2}\end{array}\right|\frac{1-x}{2}\right) \\ &= \frac{(\lambda)_n(2x)^n}{n!}\,{}_2F_1\left(\left.\begin{array}{c} -n/2,-(n-1)/2 \\ 1-n-\lambda\end{array}\right|\frac{1}{x^2}\right). \end{aligned}$$

4.6 Potenzdarstellungen klassischer OPS

Ziel des nächsten Abschnitts ist es Verbindungskoeffizienten für den Fall $P_n(x) = x^n$ zu finden. Dieser Fall ist von besonderem Interesse, da man in vielen Anwendungen ein gegebenes Polynom bzgl. eines gegebenen klassischen OPS $Q_m(x)$ entwickeln möchte.

Satz 4.17
Sei $Q_m(x)$ ein OPS mit dem höchsten Koeffizienten $\overline{k}_m = 1$, welches durch die Eigenwertdifferentialgleichung mit $\overline{\sigma}(x) = \overline{a}x^2 + \overline{b}x + \overline{c}$ und $\overline{\tau}(x) = \overline{d}x + \overline{e}$ gegeben sei. Dann erfüllen die Koeffizienten $C_m(x)$ der Potenzdarstellung

$$x^n = \sum_{m=0}^{n} C_m(n) Q_m(x)$$

die Rekursionsgleichung

$$\begin{aligned}
&(n-m)(\overline{d}+2\overline{a}m)(\overline{d}+3\overline{a}+2\overline{a}m)(\overline{d}+\overline{a}+2\overline{a}m)(\overline{d}+2\overline{a}m+2\overline{a})^2 C_m(n) \\
&\quad + (\overline{d}\overline{e}+\overline{b}\overline{d}+2\overline{d}\overline{b}m+2\overline{a}m^2\overline{b}+2\overline{a}m\overline{b}+2\overline{e}\overline{a}n-\overline{d}\overline{b}n)(\overline{d}+2\overline{a}m+2\overline{a}) \\
&\quad (m+1)(\overline{d}+3\overline{a}+2\overline{a}m)(\overline{d}+\overline{a}+2\overline{a}m)C_{m+1}(n) - (m+2)(-4\overline{a}^2\overline{c}m^2 \\
&\quad + \overline{a}\overline{b}^2 m^2 + 2\overline{a}\overline{b}^2 m - 4\overline{a}\overline{c}m\overline{d} - 8\overline{a}^2\overline{c}m + m\overline{b}^2\overline{d} - \overline{a}\overline{e}^2 - \overline{d}^2\overline{c} + \overline{b}\overline{e}\overline{d} - 4\overline{a}^2\overline{c} \\
&\quad - 4\overline{a}\overline{c}\overline{d} + \overline{a}\overline{b}^2 + \overline{b}^2\overline{d})(\overline{a}m+\overline{a}n+\overline{a}+\overline{d})(m+1)(\overline{d}+2\overline{a}m)C_{m+2}(n) = 0 \qquad (38)
\end{aligned}$$

Ist $\overline{c} = 0$, dann ist die Rekursion

$$(n-m)(\overline{d}+2\overline{a}m)(\overline{d}+\overline{a}+2\overline{a}m)C_m(n) + (m+1)(\overline{b}m+\overline{e})(\overline{a}m+n\overline{a}+\overline{d})C_{m+1}(n) = 0 \quad (39)$$

gültig und wir erhalten ($\overline{a}\overline{b} \neq 0$)

$$C_m(n) = \frac{\left(\frac{\overline{e}}{\overline{b}}\right)_n}{\left(\frac{\overline{d}}{\overline{a}}\right)_n}\left(-\frac{\overline{b}}{\overline{a}}\right)^n \cdot \frac{(-n)_m \left(\frac{\overline{d}}{2\overline{a}}\right)_m \left(\frac{\overline{a}+\overline{d}}{2\overline{a}}\right)_m}{\left(\frac{\overline{e}}{\overline{b}}\right)_m \left(\frac{\overline{a}n+\overline{d}}{\overline{a}}\right)_m m!}\left(\frac{4\overline{a}}{\overline{b}}\right)^m . \quad (40)$$

Daher gelten die folgenden Potenzdarstellungen für die klassischen OPS:

$$(1-x)^n = 2^n\Gamma(\alpha+n+1)\sum_{m=0}^{n}\frac{(\alpha+\beta+2m+1)\Gamma(\alpha+\beta+m+1)}{\Gamma(\alpha+m+1)\Gamma(\alpha+\beta+n+m+2)}(-n)_m P_m^{(\alpha,\beta)}(x),$$

$$(1+x)^n = 2^n\Gamma(\beta+n+1)\sum_{m=0}^{n}(-1)^m\frac{(\alpha+\beta+2m+1)\Gamma(\alpha+\beta+m+1)}{\Gamma(\beta+m+1)\Gamma(\alpha+\beta+n+m+2)}(-n)_m P_m^{(\alpha,\beta)}(x),$$

$$\begin{aligned}
x^n &= \frac{n!}{(\alpha)_n 2^n}\sum_{k=0}^{\lfloor n/2\rfloor}\frac{(-n/2-\alpha/2+1)_k(-n-\alpha)_k}{(-n/2-\alpha/2)_k k!}(-1)^k C_{n-2k}^{\alpha}(x) \\
&= \frac{n!}{2^n}\sum_{k=0}^{\lfloor n/2\rfloor}\frac{n+\alpha-2k}{k!(\alpha)_{n+1-k}}C_{n-2k}^{\alpha}(x),
\end{aligned}$$

$$x^n = (1+\alpha)_n \sum_{m=0}^{n} \frac{(-n)_m}{(1+\alpha)_m} L_m^{(\alpha)}(x) = n! \sum_{m=0}^{n} \binom{n+\alpha}{n-m} (-1)^m L_m^{(\alpha)}(x),$$

$$x^n = \sum_{k=0}^{\lfloor n/2 \rfloor} \frac{(-n/2)_k(-n/2+1/2)_k}{k!2^{n-2k}} H_{n-2k}(x) = \frac{n!}{2^n} \sum_{k=0}^{\lfloor n/2 \rfloor} \frac{1}{k!(n-2k)!} H_{n-2k}(x),$$

$$\begin{aligned} x^n &= \frac{(-2)^n}{(\alpha+2)_n} \sum_{m=0}^{n} \frac{(-n)_m(\alpha+1)_m(\alpha/2+3/2)_m}{(n+2+\alpha)_m(\alpha/2+1/2)_m m!} B_m^{(\alpha)}(x) \\ &= (-2)^n \sum_{m=0}^{n} (2m+\alpha+1) \frac{(-n)_m \Gamma(\alpha+m+1)}{m!\Gamma(n+m+\alpha+2)} B_m^{(\alpha)}(x). \end{aligned}$$

Beweis:
Im vorherigen Abschnitt haben wir bereits drei Kreuzregeln für die Verbindungskoeffizienten von $P_n(x)$ und $Q_m(x)$ hergeleitet. Diese wenden wir nun auf den Spezialfall $P_n(X) = x^n$ an.
Da $Q_m(x)$ ebenfalls ein klassisches OPS ist, erfüllt es die Eigenwertdifferentialgleichung

$$\overline{\sigma}(x)Q_m''(x) + \overline{\tau}(x)Q_m'(x) + \overline{\lambda}Q_m(x) = 0$$

mit $\overline{\sigma}(x) = \overline{a}x^2 + \overline{b}x + \overline{c}$ sowie die Ableitungsregel

$$\overline{\sigma}(x)Q_m'(x) = \overline{\alpha}_m Q_{m+1}(x) + \overline{\beta}_m Q_m(x) + \overline{\gamma}_m Q_{m-1}(x).$$

Da wir das spezielle Verbindungsproblem $P_n(x) = x^n$ betrachten, erhalten wir wegen

$$\begin{aligned} \overline{\sigma}(x)P_n'(x) &= (\overline{a}x^2 + \overline{b}x + \overline{c})nx^{n-1} \\ &= \overline{a}x^2nx^{n-1} + \overline{b}nxx^{n-1} + \overline{c}nx^{n-1} \\ &= \overline{a}nx^{n+1} + \overline{b}nx^n + \overline{c}nx^{n-1} \\ &= \overline{a}nP_{n+1}(x) + \overline{b}nP_n(x) + \overline{c}nP_{n-1}(x) \end{aligned}$$

die Ableitungsregel

$$\overline{\sigma}(x)P_n'(x) = \overline{a}nP_{n+1}(x) + \overline{b}nP_n(x) + \overline{c}nP_{n-1}(x). \tag{41}$$

Wir erhalten also die neue Situation $a_n = 1$ und $b_n = c_n = 0$. Angewandt auf die erste Kreuzregel erhalten wir

$$C_m(n+1) = \overline{a}_{m-1}C_{m-1}(n) + \overline{b}_m C_m(n) + \overline{c}_{m+1}C_{m+1}(n). \tag{42}$$

Im Falle von (36) gilt $\widehat{a}_n = \dfrac{1}{n+1}$ sowie $\widehat{b}_n = \widehat{c}_n = 0$ und wir erhalten

$$\frac{1}{n+1}C_m(n+1) = \overline{\overline{\widehat{a}}}_{m-1}C_{m-1}(n) + \overline{\overline{\widehat{b}}}_m C_m(n) + \overline{\overline{\widehat{c}}}_{m+1}C_{m+1}(n). \tag{43}$$

Wenden wir unsere Erkenntnisse der neuen Ableitungsregel auf die dritte Kreuzregel an, so erhalten wir

$$\overline{a}nC_m(n+1) + \overline{b}nC_m(n) + \overline{c}nC_m(n-1) = \overline{\alpha}_{m-1}C_{m-1}(n) + \overline{\beta}_m C_m(n) + \overline{\gamma}_{m+1}C_{m+1}(n). \tag{44}$$

Zur Herleitung der Rekursionsformeln verwenden wir *Mathematica*. Dazu geben wir zuerst die hergeleiteten Kreuzregeln ein.

In[86]:= $\mathbf{cr1} = \mathrm{oa}_{m-1}\,\mathrm{C}_{m-1}[n] + \mathrm{ob}_m\,\mathrm{C}_m[n] + \mathrm{oc}_{m+1}\mathrm{C}_{m+1}[n] == \mathrm{C}_m[n+1]\,;$

$\mathbf{cr2} = \hat{\mathrm{oa}}_{m-1}\,\mathrm{C}_{m-1}[n] + \hat{\mathrm{ob}}_m\,\mathrm{C}_m[n] + \hat{\mathrm{oc}}_{m+1}\mathrm{C}_{m+1}[n] == \frac{1}{n+1}\mathrm{C}_m[n+1]\,;$

$\mathbf{cr3} = \mathrm{oa}\,n\,\mathrm{C}_m[n+1] + \mathrm{ob}\,n\,\mathrm{C}_m[n] + \mathrm{oc}\,n\,\mathrm{C}_m[n-1] ==$
$\quad \mathrm{o}\alpha_{m-1}\,\mathrm{C}_{m-1}[n] + \mathrm{o}\beta_m\,\mathrm{C}_m[n] + \mathrm{o}\gamma_{m+1}\mathrm{C}_{m+1}[n]\,;$

$\mathbf{cr4} = \mathrm{oa}\,n\,\mathrm{C}_m[n+1] + \mathrm{ob}\,n\,\mathrm{C}_m[n] ==$
$\quad \mathrm{o}\alpha_{m-1}\,\mathrm{C}_{m-1}[n] + \mathrm{o}\beta_m\,\mathrm{C}_m[n] + \mathrm{o}\gamma_{m+1}\mathrm{C}_{m+1}[n]\,;$

Elimination von $C_m(n-1)$ und $C_m(n+1)$ sowie $C_m(n)$ und $C_m(n+1)$ im Falle $\overline{c} = 0$ liefert uns die Rekursionen

In[87]:= $\mathbf{rek1} = \mathbf{Eliminate}[\{\mathbf{cr1}, \mathbf{cr2}, \mathbf{cr3}\}, \{\mathrm{C}_m[n+1], \mathrm{C}_m[n-1]\}][[1]]$

Out[87]= $\mathrm{oa}_{m-1}\,\mathrm{C}_{m-1}(n) == n\,\hat{\mathrm{oa}}_{m-1}\,\mathrm{C}_{m-1}(n) + \hat{\mathrm{oa}}_{m-1}\,\mathrm{C}_{m-1}(n) - \mathrm{ob}_m\,\mathrm{C}_m(n) +$
$\quad n\,\hat{\mathrm{ob}}_m\,\mathrm{C}_m(n) + \hat{\mathrm{ob}}_m\,\mathrm{C}_m(n) - \mathrm{oc}_{m+1}\,\mathrm{C}_{m+1}(n) + n\,\hat{\mathrm{oc}}_{m+1}\,\mathrm{C}_{m+1}(n) + \hat{\mathrm{oc}}_{m+1}\,\mathrm{C}_{m+1}(n)$

und

In[88]:= $\mathbf{rek2} = \mathbf{Eliminate}[\{\mathbf{cr1}, \mathbf{cr2}, \mathbf{cr4}\}, \{\mathrm{C}_m[n+1], \mathrm{C}_{m-1}[n]\}][[1]]$

Out[88]= $\mathrm{ob}_m\left(-\mathrm{oa}\,\hat{\mathrm{oa}}_{m-1}\,n^2 - \mathrm{oa}\,\hat{\mathrm{oa}}_{m-1}\,n + \mathrm{o}\alpha_{m-1}\right)\mathrm{C}_m(n) ==$
$\mathrm{ob}\,\hat{\mathrm{oa}}_{m-1}\,\mathrm{C}_m(n)\,n^2 - \mathrm{oa}\,\mathrm{oa}_{m-1}\,\hat{\mathrm{ob}}_m\,\mathrm{C}_m(n)\,n^2 + \mathrm{oa}\,\mathrm{oc}_{m+1}\,\hat{\mathrm{oa}}_{m-1}\,\mathrm{C}_{m+1}(n)\,n^2 -$
$\quad \mathrm{oa}\,\mathrm{oa}_{m-1}\,\hat{\mathrm{oc}}_{m+1}\,\mathrm{C}_{m+1}(n)\,n^2 - \mathrm{ob}\,\mathrm{oa}_{m-1}\,\mathrm{C}_m(n)\,n + \mathrm{ob}\,\hat{\mathrm{oa}}_{m-1}\,\mathrm{C}_m(n)\,n - \mathrm{o}\beta_m\,\hat{\mathrm{oa}}_{m-1}\,\mathrm{C}_m(n)\,n -$
$\quad \mathrm{oa}\,\mathrm{oa}_{m-1}\,\hat{\mathrm{ob}}_m\,\mathrm{C}_m(n)\,n + \mathrm{o}\alpha_{m-1}\,\hat{\mathrm{ob}}_m\,\mathrm{C}_m(n)\,n + \mathrm{oa}\,\mathrm{oc}_{m+1}\,\hat{\mathrm{oa}}_{m-1}\,\mathrm{C}_{m+1}(n)\,n -$
$\quad \mathrm{o}\gamma_{m+1}\,\hat{\mathrm{oa}}_{m-1}\,\mathrm{C}_{m+1}(n)\,n - \mathrm{oa}\,\mathrm{oa}_{m-1}\,\hat{\mathrm{oc}}_{m+1}\,\mathrm{C}_{m+1}(n)\,n + \mathrm{o}\alpha_{m-1}\,\hat{\mathrm{oc}}_{m+1}\,\mathrm{C}_{m+1}(n)\,n +$
$\quad \mathrm{oa}_{m-1}\,\mathrm{o}\beta_m\,\mathrm{C}_m(n) - \mathrm{o}\beta_m\,\hat{\mathrm{oa}}_{m-1}\,\mathrm{C}_m(n) + \mathrm{o}\alpha_{m-1}\,\hat{\mathrm{ob}}_m\,\mathrm{C}_m(n) - \mathrm{oc}_{m+1}\,\mathrm{o}\alpha_{m-1}\,\mathrm{C}_{m+1}(n) +$
$\quad \mathrm{oa}_{m-1}\,\mathrm{o}\gamma_{m+1}\,\mathrm{C}_{m+1}(n) - \mathrm{o}\gamma_{m+1}\,\hat{\mathrm{oa}}_{m-1}\,\mathrm{C}_{m+1}(n) + \mathrm{o}\alpha_{m-1}\,\hat{\mathrm{oc}}_{m+1}\,\mathrm{C}_{m+1}(n)$

Nach Einsetzen der Koeffizienten und einem Shift von m bei der ersten Rekursion erhalten wir (38)

In[89]:= $\mathbf{rek3} = \mathbf{rek1} /.\, m \to m+1;$

In[90]:= $\mathbf{rek4} =$
$\quad \mathbf{Map}[\mathbf{Factor},$
$\quad\quad \mathbf{Collect}[$
$\quad\quad\quad \mathbf{Simplify}[\mathbf{Numerator}[\mathbf{Together}[\mathbf{rek3}[[1]] - \mathbf{rek3}[[2]]]]],$
$\quad\quad\quad \{\mathrm{C}_m[n], \mathrm{C}_{m+1}[n], \mathrm{C}_{m+2}[n]\}]]$

Out[90]= $(m-n)\,(2\,m\,\mathrm{oa} + \mathrm{od})\,(2\,m\,\mathrm{oa} + \mathrm{oa} + \mathrm{od})\,(2\,m\,\mathrm{oa} + 3\,\mathrm{oa} + \mathrm{od})\,\mathrm{C}_m(n)$
$\quad (2\,m\,\mathrm{oa} + 2\,\mathrm{oa} + \mathrm{od})^2 - (m+1)\,(2\,m\,\mathrm{oa} + \mathrm{oa} + \mathrm{od})\,(2\,m\,\mathrm{oa} + 3\,\mathrm{oa} + \mathrm{od})$
$\quad \left(2\,\mathrm{oa}\,\mathrm{ob}\,m^2 + 2\,\mathrm{oa}\,\mathrm{ob}\,m + 2\,\mathrm{ob}\,\mathrm{od}\,m - n\,\mathrm{ob}\,\mathrm{od} + \mathrm{ob}\,\mathrm{od} + 2\,n\,\mathrm{oa}\,\mathrm{oe} + \mathrm{od}\,\mathrm{oe}\right)\mathrm{C}_{m+1}(n)$
$\quad (2\,m\,\mathrm{oa} + 2\,\mathrm{oa} + \mathrm{od}) - (m+1)\,(m+2)\,(2\,m\,\mathrm{oa} + \mathrm{od})\,(m\,\mathrm{oa} + n\,\mathrm{oa} + \mathrm{oa} + \mathrm{od})$
$\quad \left(4\,m^2\,\mathrm{oc}\,\mathrm{oa}^2 + 8\,m\,\mathrm{oc}\,\mathrm{oa}^2 + 4\,\mathrm{oc}\,\mathrm{oa}^2 - m^2\,\mathrm{ob}^2\,\mathrm{oa} - 2\,m\,\mathrm{ob}^2\,\mathrm{oa} - \mathrm{ob}^2\,\mathrm{oa} + \mathrm{oe}^2\,\mathrm{oa} +\right.$
$\quad\quad \left.4\,m\,\mathrm{oc}\,\mathrm{od}\,\mathrm{oa} + 4\,\mathrm{oc}\,\mathrm{od}\,\mathrm{oa} + \mathrm{oc}\,\mathrm{od}^2 - m\,\mathrm{ob}^2\,\mathrm{od} - \mathrm{ob}^2\,\mathrm{od} - \mathrm{ob}\,\mathrm{od}\,\mathrm{oe}\right)\mathrm{C}_{m+2}(n)$

```
In[91]:= rek5 =
           Map[Factor,
             Collect[
               Simplify[Numerator[Together[rek2[[1]] - rek2[[2]]]]]/.
                oc → 0, {C_m[n], C_{m+1}[n]}]]
```

Out[91]= $(m+1)(m-n-1)(m\,\mathrm{oa}+n\,\mathrm{oa}+\mathrm{od})$
$(m\,\mathrm{ob}+\mathrm{oe})(m\,\mathrm{oa}\,\mathrm{ob}+\mathrm{od}\,\mathrm{ob}-\mathrm{oa}\,\mathrm{oe})\,\mathrm{C}_{m+1}(n)-$
$(m-n-1)(m-n)(2\,m\,\mathrm{oa}+\mathrm{od})(2\,m\,\mathrm{oa}+\mathrm{oa}+\mathrm{od})(m\,\mathrm{oa}\,\mathrm{ob}+\mathrm{od}\,\mathrm{ob}-\mathrm{oa}\,\mathrm{oe})\,\mathrm{C}_m(n)$

und (39) nach Elimination von Vielfachen

```
In[92]:= Map[Factor,
           Collect[
             1/PolynomialGCD[Coefficient[rek5, {C_m[n], C_{m+1}[n]}][[1]],
                 Coefficient[rek5, {C_m[n], C_{m+1}[n]}][[2]]]rek5,
             {C_m[n], C_{m+1}[n]}]]
```

Out[92]= $(m-n)(2\,m\,\mathrm{oa}+\mathrm{od})(2\,m\,\mathrm{oa}+\mathrm{oa}+\mathrm{od})\,\mathrm{C}_m(n)-$
$(m+1)(m\,\mathrm{oa}+n\,\mathrm{oa}+\mathrm{od})(m\,\mathrm{ob}+\mathrm{oe})\,\mathrm{C}_{m+1}(n)$

Um die Darstellung (40) zu verifizieren zeigen wir die Gleichheit der Termverhältnisse, welche (39) und (40) liefern. Dazu formen wir (39) um und erhalten

$$\frac{C_{m+1}(n)}{C_m(n)} = \frac{(m-n)(\overline{d}+2\overline{a}m)(\overline{d}+\overline{a}+2\overline{a}m)}{(m+1)(\overline{b}m+\overline{e})(\overline{a}m+n+\overline{a}+\overline{d})}.$$

In Mathematica setzen wir

In[93]:= $\mathtt{Cm} = \dfrac{\mathtt{Pochhammer}\left[\frac{e}{b}, n\right]}{\mathtt{Pochhammer}\left[\frac{d}{a}, n\right]}\left(\dfrac{-b}{a}\right)\hat{}\,n$
$\dfrac{\mathtt{Pochhammer}[-n, m]\,\mathtt{Pochhammer}\left[\frac{d}{2a}, m\right]\mathtt{Pochhammer}\left[\frac{a+d}{2a}, m\right]}{\mathtt{Pochhammer}\left[\frac{e}{b}, m\right]\mathtt{Pochhammer}\left[\frac{a*n+d}{a}, m\right]m!}$
$\left(\dfrac{4a}{b}\right)\hat{}\,m$

Out[93]= $\dfrac{4^m\left(\frac{a}{b}\right)^m\left(-\frac{b}{a}\right)^n\left(\frac{d}{2\,a}\right)_m\left(\frac{a+d}{2\,a}\right)_m\left(\frac{e}{b}\right)_n(-n)_m}{m!\left(\frac{d}{a}\right)_n\left(\frac{e}{b}\right)_m\left(\frac{d+a\,n}{a}\right)_m}$

und betrachten

In[94]:= $\dfrac{\mathtt{Cm/.m \to m+1}}{\mathtt{Cm}}$ `//FunctionExpand`

Out[94]= $\dfrac{(d+2\,a\,m)(2\,m\,a+a+d)(m-n)}{(m+1)(e+b\,m)(d+a\,m+a\,n)}$

Die Termverhältnisse sind also gleich. Somit ist die Gültigkeit von (40) gezeigt.
Die Potenzdarstellungen der klassischen OPS erhalten wir dann aus den zuvor hergeleiteten Darstellungen. □

Mit den Resultaten aus Satz 4.17 können wir nun auch explizite Formeln für die Verbindungskoeffizienten der klassischen orthogonalen Polynome finden.

Satz 4.18
Die klassischen OPS haben folgende Verbindungskoeffizienten

$$P_n^{(\alpha,\beta)}(x) = \sum_{m=0}^{n}(2m+\gamma+\beta+1)\frac{\Gamma(n+\beta+1)\Gamma(n+m+\alpha+\beta+1)}{\Gamma(m+\beta+1)\Gamma(n+\alpha+\beta+1)}\cdot$$

$$\frac{\Gamma(m+\gamma+\beta+1)}{\Gamma(n+m+\gamma+\beta+2)}\frac{(\alpha-\gamma)_{n-m}}{(n-m)!}P_m^{(\gamma,\beta)},$$

$$P_n^{(\alpha,\beta)}(x) = \sum_{m=0}^{n}(-1)^{n-m}(2m+\alpha+\delta+1)\frac{\Gamma(n+\alpha+1)\Gamma(n+m+\alpha+\beta+1)}{\Gamma(m+\alpha+1)\Gamma(n+\alpha+\beta+1)}\cdot$$

$$\frac{\Gamma(m+\alpha+\delta+1)}{\Gamma(n+m+\alpha+\delta+2)}\frac{(\beta-\delta)_{n-m}}{(n-m)!}P_m^{(\alpha,\delta)}(x),$$

$$C_n^{(\alpha)}(x) = \frac{\Gamma(\beta)}{\Gamma(\alpha)\Gamma(\alpha-\beta)}\sum_{k=0}^{\lfloor n/2\rfloor}\frac{(n-2k+\beta)\Gamma(k+\alpha-\beta)\Gamma(n-k+\alpha)}{k!\Gamma(n-k+\beta+1)}C_{n-2k}^{(\beta)}(x),$$

$$L_n^{(\alpha)}(x) = \sum_{m=0}^{n}\frac{(\alpha-\beta)_{n-m}}{(n-m)!}L_m^{(\beta)}(x),$$

$$B_n^{(\alpha)}(x) = \frac{(-1)^n(\alpha-\beta)_n}{(\beta+2)_n}\sum_{m=0}^{n}\frac{(-n)_m(\beta+1)_m(\beta/2+3/2)_m(n+\alpha+1)_m}{(n+2+\beta)_m(\beta/2+1/2)_m(\beta-\alpha+1-n)_m m!}(-1)^m B_m^{(\beta)}(x)$$

$$= \sum_{m=0}^{n}(-1)^m(2m+\beta+1)\frac{(-n)_m\Gamma(\beta+m+1)(n+\alpha+1)_m\Gamma(\beta-\alpha+1)}{m!\Gamma(n+m+\beta+2)\Gamma(m-n+\beta-\alpha+1)}B_m^{(\beta)}(x).$$

Beweis:
Wir suchen die Koeffizienten $C_m(n)$ der Verbindungsformel

$$P_n(x) = \sum_{m=0}^{n} C_m(n)Q_m(x).$$

Zusammen mit

$$P_n(x) = \sum_{j\in\mathbb{Z}} A_j(n)x^j$$

sowie

$$x^j = \sum_{m\in\mathbb{Z}} B_m(j)Q_m(x)$$

erhalten wir

$$P_n(x) = \sum_{j\in\mathbb{Z}}\sum_{m\in\mathbb{Z}} A_j(n)B_m(j)Q_m(x).$$

Vertauschen der Summationsreihenfolge liefert

$$C_m(n) = \sum_{j\in\mathbb{Z}} A_j(n)B_m(j).$$

Falls, wie z.B. im Falle der Gegenbauerpolynome, für $P_n(x)$ und x^j Darstellungen der Form

$$P_n(x) = \sum_{j\in\mathbb{Z}} A_j(n)x^{n-2j}$$

und

$$x^j = \sum_{m\in\mathbb{Z}} B_m(j)Q_{j-2m}(x)$$

erfüllt sind, so erhalten wir

$$D_m(n) = \sum_{j\in\mathbb{Z}} A_j(n)B_{m-2j}(n-2j)$$

mit

$$P_n(x) = \sum_{m=0}^{n} D_m(n)Q_{n-2m}(x).$$

Weil der Summand $F(j,m,n) := A_j(n)B_m(j)$ ein hypergeometrischer Term bzgl. j,m,n ist, können wir mit dem Fasenmyer- oder Zeilbergeralgorithmus die gesuchten Rekursionen für $C_m(n)$ bzgl m und n herleiten. In allen Fällen erhalten wir Rekursionen erster Ordnung, welche hypergeometrische Terme und somit die angegebenen Darstellungen liefern. Wir wollen nun die Gültigkeit der Darstellungen am Beispiel der Laguerre- und Besselpolynome zeigen. Dazu überprüfen wir die Termverhältnisse $\frac{A_{m+1}}{A_m}$. Zur Bestimmung der Rekursionen aus $F(j,m,n)$ verwenden wir die Implementierung des Zeilbergeralgorithmus *SumRekursion* aus dem *SpecialFunctions* Package.
Zuerst deklarieren wir

$$A_j(n) = \binom{n+\alpha}{n}\frac{(-n)_j}{j!(\alpha+1)_j}$$

gemäß Satz 4.17 und

$$B_m(j) = j!\binom{j+\beta}{j-m}(-1)^m$$

gemäß (19). Dies liefert uns nach Anwendung des Zeilbergeralgorithmus

```
In[95]:= lhyp = Binomial[n + α, n] Pochhammer[-n, j] / (Pochhammer[α + 1, j] j!);
In[96]:= lpot = j! Binomial[j + β, j - m] (-1)^m;
In[97]:= SumRekursion[lpot * lhyp, j, A[m]]
Out[97]= (m - n) A(m) - (m - n - α + β + 1) A(m + 1) == 0
```

Umformung liefert das Termverhältnis

$$\frac{A_{m+1}}{A_m} = \frac{n-m}{-m+n+\alpha-\beta-1}.$$

Betrachten wir nun

```
In[98]:= Lag = Pochhammer[α - β, n - m] / (n - m)!;
```

```
In[99]:= Lag/.m → m + 1 / Lag //FunctionExpand
```
$$\texttt{Out[99]=}\ \frac{n-m}{-m+n+\alpha-\beta-1}$$

so sehen wir, dass die Termverhältnisse übereinstimmen. Daraus folgt die Gültigkeit der Darstellung.
Analog erhalten wir für die Besselpolynome die Rekursion

```
In[100]:= bhyp = Pochhammer[-n, j]Pochhammer[n + α + 1, j](-1/2)^j / j!;
In[101]:= bpot = (-2)^j (2m + β + 1)Pochhammer[-j, m]Gamma[β + m + 1] / (m! Gamma[j + m + β + 2]);
In[102]:= SumRekursion[bhyp * bpot, j, A[m]]
```
$$\texttt{Out[102]=}\ (m-n)\,(m+n+\alpha+1)\,(m+\beta+1)\,(2\,m+\beta+3)\,A(m)+ (m+1)\,(2\,m+\beta+1)\,(m+n+\beta+2)\,(m-n-\alpha+\beta+1)\,A(m+1)==0$$

mit dem Termverhältnis

$$\frac{A_{m+1}}{A_m}=-\frac{(m-n)(m+n+\alpha+1)(m+\beta+1)(2(m+1)+\beta+1)}{(m+1)(2m+\beta+1)(mn+\beta+2)(m-n-\alpha+\beta+1)}.$$

Mit

```
In[103]:= Bes =
    ((-1)^m (2m + β + 1)Pochhammer[-n, m]Pochhammer[n + α + 1, m]
        Gamma[β + m + 1]Gamma[β - α + 1])/
     (m! Gamma[n + m + β + 2]Gamma[m - n + β - α + 1]);
In[104]:= Bes/. m → m + 1 / Bes //FunctionExpand
```
$$\texttt{Out[104]=}\ -\frac{(m-n)\,(m+n+\alpha+1)\,(m+\beta+1)\,(2\,(m+1)+\beta+1)}{(m+1)\,(2\,m+\beta+1)\,(m+n+\beta+2)\,(m-n-\alpha+\beta+1)}$$

folgt dann die Gleichheit der Termverhältnisse und somit die Gültigkeit der Darstellung.
□

In einigen Anwendungen ist es von Interesse, wie sich die Änderung eines OPS in Richtung eines Parameters durch das gegebene System selbst ausdrückt. Also z.B.: Wie ändert sich das Jacobi-System, wenn sich der Parameter α ändert?
Dazu betrachten wir folgenden Satz:

Satz 4.19
Es gelten die folgenden Parameterdarstellungen für die Parameterableitungen der klassischen OPS:

$$\frac{\partial}{\partial\alpha}P_n^{(\alpha,\beta)}(x)=\sum_{m=0}^{n-1}\frac{1}{\alpha+\beta+1+m+n}(P_n^{(\alpha,\beta)}(x)+$$
$$\frac{\alpha+\beta+1+2m}{n-m}\frac{(\beta+m+1)_{n-m}}{(\alpha+\beta+m+1)_{n-m}}P_m^{(\alpha,\beta)}(x)),$$
$$\frac{\partial}{\partial\beta}P_n^{(\alpha,\beta)}(x)=\sum_{m=0}^{n-1}\frac{1}{\alpha+\beta+1+m+n}(P_n^{(\alpha,\beta)}(x)+$$

$$(-1)^{n-m}\frac{\alpha+\beta+1+2m}{n-m}\frac{(\alpha+m+1)_{n-m}}{(\alpha+\beta+m+1)_{n-m}}P_m^{(\alpha,\beta)}(x)),$$

$$\frac{\partial}{\partial\alpha}C_n^{(\alpha)}(x) = \sum_{m=0}^{n-1}\left(\frac{2(1+m)}{(2\alpha+m)(2\alpha+1+2m)}+\frac{2}{2\alpha+m+n}\right)C_n^{(\alpha)}(x)$$

$$+\sum_{m=0}^{n-1}\frac{2(1+(-1)^{n-m})(\alpha+m)}{(2\alpha+m+n)(n-m)}C_m^{(\alpha)}(x),$$

$$\frac{\partial}{\partial\alpha}L_n^{(\alpha)}(x) = \sum_{m=0}^{n-1}\frac{1}{n-m}L_m^{(\alpha)}(x),$$

$$\frac{\partial}{\partial\alpha}B_n^{(\alpha)}(x) = \sum_{m=0}^{n-1}\frac{1}{\alpha+n+m+1}(B_n^{(\alpha)}(x)+$$

$$(-1)^{n-m}\frac{2m+\alpha+1}{(n-m)}\frac{n!}{(\alpha+m+1)_{n-m}m!}B_m^{(\alpha)}(x)).$$

Beweis:
Wir betrachten die Verbindungsbeziehung

$$P_n^{(\alpha)}(x) = \sum_{m=0}^{n} C_m(n;\alpha,\beta)P_m^{(\beta)}(x)$$

und den Differenzenquotienten

$$\frac{P_n^{(\alpha)}(x)-P_n^{(\beta)}(x)}{\alpha-\beta},$$

so erhalten wir

$$\begin{aligned}
\frac{P_n^{(\alpha)}(x)-P_n^{(\beta)}(x)}{\alpha-\beta} &= \sum_{m=0}^{n}\frac{C_m(n;\alpha,\beta)}{\alpha-\beta}P_m^{(\beta)}(x)-\frac{P_n^{(\beta)}(x)}{\alpha-\beta}\\
&= \frac{C_n(n;\alpha,\beta)}{\alpha-\beta}P_n^{(\beta)}(x)+\sum_{m=0}^{n-1}\frac{C_m(n;\alpha,\beta)}{\alpha-\beta}P_m^{(\beta)}(x)-\frac{P_n^{(\beta)}(x)}{\alpha-\beta}\\
&= \frac{C_n(n;\alpha,\beta)-1}{\alpha-\beta}P_n^{(\beta)}(x)+\sum_{m=0}^{n-1}\frac{C_m(n;\alpha,\beta)}{\alpha-\beta}P_m^{(\beta)}(x).
\end{aligned}$$

Aufgrund der Stetigkeit der Polynomsysteme $P_n^{(\alpha)}(x)$ bzgl. α gilt dann die Gleichung

$$\frac{\partial}{\partial\alpha}P_n^{(\alpha)}(x) = \lim_{\beta\to\alpha}\frac{C_n(n;\alpha,\beta)-1}{\alpha-\beta}P_n^{(\alpha)}(x)+\sum_{m=0}^{n-1}\lim_{\beta\to\alpha}\frac{C_m(n;\alpha,\beta)}{\alpha-\beta}P_m^{(\alpha)}(x).$$

Wir wollen die Herleitung nun am Beispiel der Laguerrepolynome zeigen. Gemäß Satz 4.18 erhalten wir für die Laguerrepolynome die Darstellung

$$C_m(n;\alpha,\beta) = \frac{(\alpha-\beta)_{n-m}}{(n-m)!}.$$

Mit *Mathematica* erhalten wir dann durch Grenzwertbildung

```
In[105]:= verbkoeff = Pochhammer[α - β, n - m]/(n - m)!;

In[106]:= Limit[(verbkoeff - 1 /. m → n)/(α - β), α → β]//FunctionExpand
Out[106]= 0

In[107]:= Limit[verbkoeff/(α - β), α → β]//FunctionExpand
```

Out[107]= $\frac{1}{n-m}$

Einsetzen in die zuvor hergeleitete Gleichung liefert uns dann die Darstellung für die Parameterableitung der Laguerrepolynome. □

4.7 Erzeugende Funktionen

Im letzten Abschnitt wollen wir erzeugende Funktionen mittels Computeralgebra finden.
Als erstes Beispiel wollen wir die erzeugende Funktion der Laguerrepolynome herleiten. Dazu definieren wir die erzeugende Funktion als Potenzreihe

$$F(z) := \sum_{k=0}^{\infty} L_n^{(\alpha)}(x) z^k$$

mit den Laguerrepolynomen als Koeffizienten. Zur Herleitung der erzeugenden Funktionen verwenden wir *Mathematica* und das *SpecialFunctions* Package.
Zuerst transformieren wir die explizite Darstellung der Laguerrepolynome (19) mittels des Zeilbergeralgorithmus in eine Rekursion

```
In[108]:= RE1 = SumRekursion[Binomial[n + α, n - k] (-1)^k/k! x^k, k, A[n]]
```

Out[108]= $(n+\alpha+1)\,A(n) - (2\,n - x + \alpha + 3)\,A(n+1) + (n+2)\,A(n+2) == 0$

Diese Rekursion transformieren wir mittels *RETODE* in eine Differentialgleichung

```
In[109]:= DE1 = RETODE[RE1, A, n, F, z]
```

Out[109]= $F'(z)\,(z-1)^2 + (x + z + z\,\alpha - \alpha - 1)\,F(z) == 0$

welche wir mit *DSolve* lösen können.

```
In[110]:= Dgl1 = DSolve[DE1, F[z], z]//Simplify
```

Out[110]= $\left\{\left\{F(z) \to e^{\frac{x}{z-1}}\,(z-1)^{-\alpha-1}\,c_1\right\}\right\}$

Nun müssen wir noch c_1 bestimmen. Dazu ermitteln wir den Startwert

```
In[111]:= startwL = Sum[Binomial[n + α, n - k] (-1)^k/k! x^k, {k, 0, 0}] /. n → 0
```

Out[111]= 1

Zusammen mit dem Startwert erhalten wir jetzt c_1

```
In[112]:= subst1 = Solve[Dgl1[[1]][[1]][[2]] == startwL, C[1]]/.z → 0
```

Out[112]= $\left\{\left\{c_1 \to (-1)^{\alpha+1}\,e^x\right\}\right\}$

Einsetzen von c_1 liefert dann die erzeugende Funktion der Laguerrepolynome

$$F(z) = e^{-\frac{xz}{1-z}}(1-z)^{-\alpha-1}.$$

Als Nächstes wollen wir die exponentielle erzeugende Funktion der Hermitepolynome bestimmen. Wir betrachten also den Ansatz

$$\sum_{n=0}^{\infty} H_k(x)\frac{z^n}{n!}.$$

Mit der Darstellung (20) der Hermitepolynome und dem Zeilbergeralgorithmus erhalten wir dann die Rekursion

```
In[113]:= RE2 = SumRekursion[ (Pochhammer[-n/2, k] Pochhammer[-(n-1)/2, k] (-1/x^2)^k (2x)^n / k!) / n!, k, A[n]]
Out[113]= 2 A(n) - 2 x A(n + 1) + (n + 2) A(n + 2) == 0
```

An dieser Stelle wollen wir die Funktionsweise der Funktion *RETORE* genauer betrachten. Um aus einer Rekursionsgleichung eine Differentialgleichung zu erhalten müssen wir die Substitution

$$k^l A(k+m) \mapsto \sum_{r=0}^{l} \binom{l}{r} (-m)^{l-r} \frac{\theta^r F(x)}{x^m}$$

durchführen, wobei $\theta F(x)$ den Differentialoperator mit

$$\theta F(x) = xF'(x)$$

darstellt.
Auf analoge Weise können wir mittels der Substitution

$$x^j f^{(k)}(x) \mapsto (n+1-j)_k \cdot A_{n+k-j}$$

eine Differentialgleichung in eine Rekursion transformieren. Siehe dazu [CompAlg].

Aus der obigen Substitutionsregel folgen die Spezialfälle

$$\begin{aligned} kA(k+m) &\mapsto -m\frac{F(x)}{x^m} + \frac{\theta F(x)}{x^m} \\ kA(k) &\mapsto \theta F(x) \\ A(k) &\mapsto \frac{F(x)}{x^m}. \end{aligned}$$

Multiplizieren wir nun die Rekursion RE2 aus, so erhalten wir

$$2A(n) - 2xA(n+1) + nA(n+2) + 2A(n+2) = 0.$$

Substituieren wir nun gemäß den definierten Regeln so erhalten wir

$$\begin{aligned} 2F(z) - \frac{2xF(z)}{z} + \left((-2)\frac{F(z)}{z^2} + \frac{zF'(z)}{z^2}\right) + \frac{2F(z)}{z^2} &= 0 \\ \Rightarrow 2F(z) - \frac{2xF(z)}{z} + \frac{F'(z)}{z} &= 0 \\ \Rightarrow 2zF(z) - 2xF(z) + F'(z) &= 0 \\ \Rightarrow F'(z) - 2(x-z)F(z) &= 0 \end{aligned}$$

Vergleich mit dem Ergebnis von *DETORE* bestätigt unser Resultat

In[114]:= **DE2 = RETODE[RE2, A, n, F, z]**

Out[114]= $F'(z) - 2\,(x - z)\,F(z) == 0$

Lösen dieser Differentialgleichung liefert

In[115]:= **Dgl2 = DSolve[DE2, F[z], z]**

Out[115]= $\{\{F(z) \to e^{2\,x\,z - z^2}\, c_1\}\}$

Auch hier müssen wir wieder c_1 bestimmen.

In[116]:= **startwH =** $\sum_{k=0}^{0} \frac{\frac{\text{Pochhammer}\left[-\frac{n}{2}, k\right]\text{Pochhammer}\left[-\frac{n-1}{2}, k\right]\left(-\frac{1}{x^{\wedge}2}\right)^{\wedge}k\,(2x)^{\wedge}n}{k!}}{n!}$ **/. n → 0**

Out[116]= 1

In[117]:= **subst2 = Solve[Dgl2[[1]][[1]][[2]] == startwH, C[1]]/.z → 0**

Out[117]= $\{\{c_1 \to 1\}\}$

Einsetzen von c_1 liefert dann die exponentielle erzeugende Funktion der Hermitepolynome

$$F(z) = e^{2xz - z^2}.$$

Die erzeugenden Funktionen der Bessel- und Jacobipolynome werden wir hier nicht herleiten, da wir für diese Polynomsysteme äußerst komplizierte Differentialgleichungen erhalten, welche nicht ohne weiterführende Überlegungen lösbar sind.
Als letztes Beispiel wollen wir abschließend die Legendrepolynome betrachen, welche den Spezialfall $\alpha = \beta = 0$ der Jacobipolynome darstellen. Gemäß (22) erhalten wir dann die Darstellung

$$P_n^{(0,0)}(x) = \left(\begin{array}{c|c} -n, n+1 & \\ & \frac{1-x}{2} \\ 1 & \end{array}\right).$$

Dies liefert die Rekursion

In[118]:= **RE6 = SumRekursion[** $\frac{\text{Pochhammer}[-n, k]\,\text{Pochhammer}[n+1, k]}{\text{Pochhammer}[1, k]\,k!}\left(\frac{1-x}{2}\right)^{\wedge}k$**, k, A[n]]**

Out[118]= $(n+1)\,A(n) - (2\,n+3)\,x\,A(n+1) + (n+2)\,A(n+2) == 0$

und die Differentialgleichung

In[119]:= **DE6 = RETODE[RE6, A, n, F, z]**

Out[119]= $(z - x)\,F(z) + \left(z^2 - 2\,x\,z + 1\right)F'(z) == 0$

Lösen dieser Differentialgleichung liefert

In[120]:= **Dgl3 = DSolve[DE6, F[z], z]//Simplify**

Out[120]= $\left\{\left\{F(z) \to \frac{c_1}{\sqrt{z^2 - 2\,x\,z + 1}}\right\}\right\}$

Als nächstes bestimmen wir c_1

In[121]:= **startwLe =** $\sum_{k=0}^{0} \frac{\text{Pochhammer}[-n, k]\,\text{Pochhammer}[n+1, k]}{\text{Pochhammer}[1, k]\,k!}\left(\frac{1-x}{2}\right)^{\wedge}k$ **/. n → 0**

```
Out[121]= 1
In[122]:= subst3 = Solve[Dgl3[[1]][[1]][[2]] == startwLe, C[1]]/.z → 0
Out[122]= {{c1 → 1}}
```

Einsetzen von c_1 liefert dann die erzeugende Funktion der Laguerrepolynome

$$F(z) = \frac{1}{\sqrt{1 - 2xz + z^2}}.$$

5 Fazit

Ziel dieser Arbeit war es, die klassischen orthogonalen Polynome als Lösungen der Eigenwertdifferentialgleichung herzuleiten sowie ihre Struktureigenschaften und Verbindungsbeziehungen zu betrachten.
Wir haben gesehen, dass man sehr weit reichende Struktureigenschaften wie z.B. Ableitung und Stammfunktion mittels Rekursionen darstellen kann. Für die Koeffizienten der Rekursionen war es sogar möglich, geschlossene Formeln mittels Koeffizientenvergleich zu finden. Ein unverzichtbares Hilfsmittel für das Aufstellen der geschlossenen Formeln war *Mathematica*, da es aufgrund der hohen Komplexität der nötigen Rechnungen nur sehr schwer möglich wäre, dies per Hand durchzuführen.
Im weiteren hat sich herausgestellt, dass es sogar möglich ist, explizite Darstellungen für die klassischen OPS, die Momente der klassischen OPS und die Verbindungsbeziehungen zu finden. Auch hier waren Rekursionsgleichungen die Basis. In diesen Fällen wiesen diese sogar hypergeometrische Termverhältnisse auf und es bot sich die Möglichkeit, mittels der hypergeometrischen Koeffizientenformel explizite Darstellungen zu bestimmen. In einigen Fällen haben wir sogar gesehen, dass es noch einfachere Darstellungen gibt, welche aber schwer herzuleiten sind. Aus diesem Grund haben wir uns darauf beschränkt diese zu verifizieren, indem wir gezeigt haben, dass die Termverhältnisse übereinstimmen.
Ingesamt haben wir gesehen, dass man mittels Methoden der linearen Algebra und Computeralgebra die klassischen OPS vollständig klassifizieren und ihre interen und externen Beziehungen bestimmen kann.

Literatur

[CAM] Koepf, W.: *Computer Algebra Methods for Orthogonal Polynomials*. Plenary talk at the International Conference on Difference Equations, Special Functions and Applications, Munich, Germany, 25-30 July 2005, Jim Cushing, Saber Elyadi, Ruppert Lasser, Vassilis Papageorgiou, Andreas Ruffing, Walter van Assche (Eds), World Scientific, 2007, 325-343.

[Chi] Chihara,T. S.: *An Introduction to Orthogonal Polynomials*. Gordon and Breach, New York, 1978.

[ClassOrth] Nikiforov, A. F., Suslov, Uvarov, S. K.,V. B.: *Classical Orthogonal Polynomials of a Discrete Variable*. Springer, Berlin-Heidelberg-New York, 1991.

[CompAlg] Koepf, W.: *Computeralgebra. Eine algorithmisch orientierte Einführung*. Springer, Berlin-Heidelberg-New York, 2006.

[COP] Koepf, W.: *Computeralgebra und orthogonale Polynome.* Universität Kassel. Wintersemester 2005/2006.

[HypSum] Koepf, W.: *Hypergeometric Summation.*Vieweg. Braunschweig / Wiesbaden, 1998.

[Les] Lesky, P.: *Über Polynomlösungen von Differentialgleichungen und Differenzengleichungen zweiter Ordnung*. Anzeiger der Österreichischen Akademie der Wissenschaften, math.-naturwiss. Klasse 121, 1985, 29-33.

[Rai] Earl D. Rainville, Ph D.: *Special Functions*. Chelsea Publishing Company. Bronx, New York, 1971.

[Rep] Koepf, W., Schmersau, D.: *Representations of orthogonal polynomials*. Journal of Computational and Applied Mathematics 90, 1998, 57-94.

[SpecFunc] Nikiforov, A. F., Uvarov, V. B.: *Special Functions of Mathematical Physics*. Birkhäuser, Basel-Boston, 1988.

[Tab] Bronstein, Semendjajew, Musiol, Mühling: *Taschenbuch der Mathematik*. Verlag Harri Deutsch, Frankfurt am Main, 2005, 2006.

[Tri] Tricomi,F. G.: *Vorlesungen über Orthogonalreihen.* Springer, Berlin/ Göttingen/Heidelberg, 1955.

Ich versichere hiermit, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Martin Meisrimel