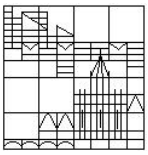

Universität Konstanz
FB Mathematik & Statistik
Prof. Dr. M. Junk
Sebastian Sahli

Ausgabe: 25.05.2016
Abgabe: 01.06.2016

Übungen zu Numerik PDGL II

Blatt 7

Aufgabe 1 (Programmieraufgabe):

Im eindimensionalen Fall ist sowohl bei P_t- als auch bei Q_t-Elementen die Referenzzelle gegeben durch das Intervall $(0, 1)$. Aus diesem Grund lohnt es sich, zunächst den Raum $\mathcal{F}((0, 1), \mathbb{R})$ genauer zu betrachten - auch im Programm.

- Wir wollen zunächst *Funktionensysteme* auf $(0, 1)$ betrachten. Allgemein lässt sich ein Funktionensystem charakterisieren über eine endliche Menge von Funktionen, von denen es erzeugt wird. Wie diese Funktionen beschrieben werden, ist zunächst nicht von Belang. Wichtig wird es jedoch sein, alle erzeugenden Funktionen auf einen Schlag an einer Stelle x auswerten zu können. Erstellen Sie eine Klasse `FunctionSystem` mit einer abstrakten Methode `evaulate`, die ein solches Funktionensystem beschreibt.

- Die einfachste Form Funktionensysteme darszustellen, ist über eine Liste von function handles. Erstellen Sie eine von `FunctionSystem` abgeleitete Klasse, die eine Liste von function handles verwaltet und implementieren Sie deren `evaluate`-Funktion.

- Von besonderem Interesse sind Systeme linear unabhängiger Funktionen, die eine Basis des von ihnen aufgespannten Raumes bilden. Erweitern Sie die `FunctionSystem`-Klasse um eine Methode `isIndependent`, die entscheidet, ob das Funktionensystem linear unabhängig ist. Als Kriterium kann das von Aufgabe 1 von Blatt 5 verwendet werden. Dazu ist es nötig, dass einem Funktionensystem (optional) ein Funktionalsystem $(\Lambda_i)_{i\in I}$ übergeben wird, das die Bedingungen der genannten Aufgabe erfüllt.

- Von weiterer Wichtigkeit sind – wenig überraschend – *Funktionen* auf $(0, 1)$. Eine Funktion wird charakterisiert durch die Fähigkeit, an einer Stelle x *ausgewertet* werden zu können.

  Funktionen lassen sich beispielsweise darstellen über function handles oder mithilfe eines Funktionensystems. Im letzteren Fall müssen das Funktionensystem und die Koordinaten der Funktion bezüglich dieses Funktionensystems angegeben werden.

  Modellieren Sie diese Situation ausgehend von einer Klasse `Function`, die eine abstrakte Methode `of` (zur Auswertung) bereitstellt.

- Funktionen sollten die Möglichkeit besitzen, zu einer anderen Funktion hinzuaddiert und mit ihr multipliziert zu werden. Erstellen Sie daher zwei von `Function` abgeleitete Klassen mit entsprechende implementierter `of`-Funktion, die die Summe bzw. das Produkt zweier Funktion repräsentieren. Erweitern Sie die ursprüngliche `Function`-Klasse um zwei Methoden, die jeweils ein weiteres `Function`-Objekt übergeben bekommen und ein Funktionsobjekt zurückliefern, das die Summe bzw. das Produkt mit der übergebenen Funktion darstellt.