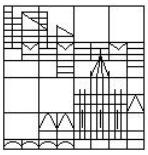

Universität Konstanz
FB Mathematik & Statistik
Prof. Dr. M. Junk
Sebastian Sahli

Ausgabe: 01.06.2016
Abgabe: 08.06.2016

Übungen zu Numerik PDGL II

Blatt 8

Dieses Blatt knüpft direkt an Blatt 7 an. Die dort geschriebenen Programme sollen hier weiterverwendet und ergänzt werden.

Aufgabe 1 (Programmieraufgabe):

- Neben den Funktionen auf $\Omega := (0,1)$ spielt auch der Dualraum $\mathcal{F}(\Omega,\mathbb{R})'$ eine große Rolle. Implementieren Sie – analog zur Klasse `Function` – zur Beschreibung eines allgemeinen Funktionals auf $\mathcal{F}(\Omega,\mathbb{R})$ eine Klasse `Functional`, die eine abstrakte `of`-Methode zur Auswertung des Funktionals zur Verfügung stellt.
- Konkrete Funktionale auf $\mathcal{F}(\Omega,\mathbb{R})$ sind beispielsweise *Auswertefunktionale*. Ein Auswertefunktional Λ ist charakterisiert durch die Eigenschaft $\Lambda(f) = f(\bar{x})$ für einen Punkt $\bar{x} \in \Omega$. Erstellen Sie eine von `Functional` abgeleitete Klasse `EvaluationFunctional`, die dies umsetzt. Die `of`-Methode soll dabei ein `Function`-Objekt erwarten und dessen Auswerte-Methode verwenden.
- Eine weitere wichtige Art von Funktionalen sind Linearkombinationen von Auswertefunktionalen. Erstellen Sie eine weitere von `Functional` abgeleitete Klasse `MultiEvaluationFunctional`, die eine Liste von Auswertestellen und Koeffizienten übergeben bekommt und implementieren Sie deren `of`-Methode.
- Später wichtig werdende spezielle Linearkombinationen von Auswertefunktionalen sind die *Quadraturformeln*, die durch geeignet gewählte Auswertestellen und Gewichte gekennzeichnet sind. Erstellen Sie als Beispiel eine von `MultiEvaluationFunctional` abgeleitete Klasse `TrapezoidalRule`, die am Konstruktor einen Parameter $N \in \mathbb{N}$ übergeben bekommt und die Auswertestellen und Gewichte so setzt, dass das resultierende Funktional der zusammengesetzten Trapezregel mit N Teilintervallen entspricht.
- Sind die Funktionale als eigene Klasse gegeben, lässt sich die `isIndependent`-Methode der `FunctionSystem`-Klasse zur Überprüfung der linearen Unabhängigkeit eleganter gestalten. Schreiben Sie diese Methode neu, dieses mal direkt in der `FunctionSystem`-Klasse (d.h. nicht mehr als abstrakte Methode). Verwenden Sie dabei nur die Methoden der `Function`- bzw. `Functional`-Klasse.
- Zu diesem Zweck wird es nötig sein, dass ein Funktionensystem in der Lage ist, seine erzeugenden Funktionen als `Function`-Objekte zurückzuliefern. Erweitern Sie die `FunctionSystem`-Klasse um eine (abstrakte) Methode `get(i)`, die die i-te Basisfunktion als `Function`-Objekt „verpackt“ zurückgeben soll und implementieren Sie diese für die `HandleFunctionSystem`-Klasse.
- Wichtige Funktionensysteme sind die von Monomen bis zu einem gewissen Grad erzeugten. Erstellen Sie eine von `FunctionSystem` abgeleitete Klasse `MonomSystem`, die einen maximalen Grad t übergeben bekommt und implementieren Sie alle von `FunctionSystem` geerbten abstrakten Methoden.