Institut für Numerische und Angewandte Mathematik
FB Mathematik und Informatik der Universität Münster
Prof. Dr. Mario Ohlberger, Dr. Stephan Rave und Tobias Wedemeier

16. Mai 2017

Übung zum Programmierpraktikum

Numerische Mehrskalenmethoden und Modellreduktion

Sommersemester 2017 — Blatt 2

Aufgabe 1

(a) Wählen Sie eine beliebige Diskretisierung `d` von Blatt 1, Aufgabe 2. Fügen Sie einen `CubicParameterSpace` als Parameterraum mittels

```
d = d.with_(parameter_space=CubicParameterSpace(...))
```

hinzu.

(b) Lösen Sie die Diskretisierung für 30 verschiedene Parameter, die Sie mittels der `sample_uniformly`- und `sample_randomly`-Methoden von `CubicParameterSpace` erhalten haben. Was ist bei der wiederholten Anwendung von `sample_randomly` zu beachten?

Speichern Sie die erhaltenen Lösungen in einem `VectorArray U`. Visualisieren Sie anschließend das Array `U`. Geben Sie außerdem die H^1- und die H^1_0-Normen der Vektoren in U aus.

Aufgabe 2

(a) Sei `U` ein beliebiges `VectorArray` und `V` ein `VectorArray` gleicher Dimension der Länge 1. Leiten Sie ein lineares Gleichungssystem für die Koeffizienten der orthogonalen Projektion (in euklidischer Norm) des in `V` enthaltenen Vektors auf den von den Vektoren in `U` aufgespannten Raum bezüglich dieser Basisvektoren her.

(b) Assemblieren und lösen Sie dieses Gleichungssystem für das `VectorArray U` aus Aufgabe 1 (b) und ein `VectorArray V`, welches eine weitere Lösung von `d` enthält. Formen Sie die Linearkombination der Koeffizienten mit den Basisvektoren in `U`. Visualisieren Sie das Ergebnis zusammen mit V. Visualisieren Sie auch den Projektionsfehler.

(c) Wiederholen sie (a) und (b) für die H^1_0-orthogonale Projektion auf `U`. Berechnen Sie Euklidische, H^1_0-, sowie L^2-Norm der Projektionsfehler für beide Projektionen.

Hinweise: Verwenden Sie die Operatoren in `d.products`.

(d) Berechnen Sie die maximalen Projektionsfehler auf die ersten `N` Vektoren in `U` für die jeweils gleichen 100 zufällig gewählten Lösungen von `d`. Plotten Sie die Projektionsfehler in Abhängigkeit von `N`.

(e) Berechnen Sie die Kondition der Projektionsmatrizen in Abhängigkeit von `N` und plotten Sie das Ergebnis.

Definition 1 (Greedy-Algorithmus)

Input: Menge von Vektoren $\mathcal{M} \subset V$, Ziel-Basisgröße N
Output: Basis $b_1, \ldots, b_N$

```
for n := 1 to N
        b_n := argmax_{m∈M} ‖m − P_span(b_1,...,b_{n−1})(m)‖
endfor
```

$$b_n := \operatorname{argmax}_{m \in \mathcal{M}} \left\| m - P_{\operatorname{span}(b_1, \ldots, b_{n-1})}(m) \right\|$$

Aufgabe 3 (Greedy-Algorithmus)

(a) Implementieren Sie den oben definierten Greedy-Algorithmus für ein die Vektoren in $\mathcal{M}$ enthaltenes `VectorArray`, einen das Skalarprodukt auf V repräsentierenden `Operator product` und eine Ziel Basisgröße `N`.

(b) Wenden Sie Ihren Algorithmus auf das `VectorArray U` mit `N` = 30 an. Berechnen Sie Projektionsfehler und Konditionen wie in Aufgabe 2 (d) und (e) für das erhaltene Ergebnis-`VectorArray`. Vergleichen Sie die Ergebnisse mit den Ergebnissen aus Aufgabe 2.