

---

Übung zum Programmierpraktikum  
**Numerische Mehrskalenmethoden und Modellreduktion**  
Sommersemester 2017 — Blatt 3

---

**Aufgabe 1** (Code cleanup)

- (a) Erstellen Sie ein Modul `problems`, in dem Sie Methoden `problem_1_1_a()`, ..., `problem_1_2_c()` implementieren. Diese sollen die in den Aufgaben von Blatt 1 zu definierenden analytischen Probleme zurückgeben.
- (b) Erstellen Sie ein Modul `algorithms` mit einer Methode

```
orthogonal_projection(U, basis, product=None,  
                      basis_is_orthogonal=False,  
                      return_projection_matrix=False)
```

und einer Methode

```
greedy_approximation(U, basis_size, rtol=0, product=None),
```

die neben der generierten Basis auch die Folge der im Algorithmus berechneten Projektionsfehler zurückliefert und zusätzlich abbricht, sobald der relative Approximationsfehler unter die durch `rtol` gegebene Schranke fällt.

**Aufgabe 2**

- (a) Wiederholen Sie die Berechnungen von Blatt 2 Aufgabe 2, wobei Sie zusätzlich die Basisvektoren in `U` mittels `pymor.algorithms.gram_schmidt` mit dem Gram-Schmidt-Verfahren orthogonalisieren. Vergleichen Sie die Ergebnisse.
- (b) Erweitern Sie auch `greedy_approximation`, sodass in jeder Iteration der neu gewählte Basisvektor zur bestehenden Basis orthogonalisiert wird. Nutzen Sie hierzu den `offset`-Parameter der `gram_schmidt`-Methode. Wiederholen Sie die Experimente von Blatt 2 Aufgabe 3.

**Zusatzaufgabe 1**

Schreiben Sie eine Methode `word_problem`, die für einen beliebigen String ein analytisches Problem im Sinne von Blatt 1 Aufgabe 2 (c) erzeugt. Jedem Buchstaben soll dabei ein Parameter zugeordnet sein.

*Hinweise: Nutzen Sie hierzu die Module `PIL.ImageDraw` und `PIL.ImageFont`.*