Karlsruher Institut für Technologie

Institut für Angewandte und Numerische Mathematik
Prof. Dr. Tobias Jahnke, Dipl.-Math. techn. Andreas Arnold

Einführung in das Wissenschaftliche Rechnen

Sommersemester 2012

6. Übungsblatt

Abgabe spätestens am 20. 7. 2012

Aufgabe 7:

(a) **Verfeinerung einer Triangulierung** (3 Punkte)
Schreiben Sie eine Funktion `refine_triang`, welche eine grobe Triangulierung dadurch verfeinert, indem die Kantenmittelpunkte zu der Eingabepunktemenge hinzugefügt werden und jedes Dreieck durch vier kleinere Dreiecke ersetzt wird.

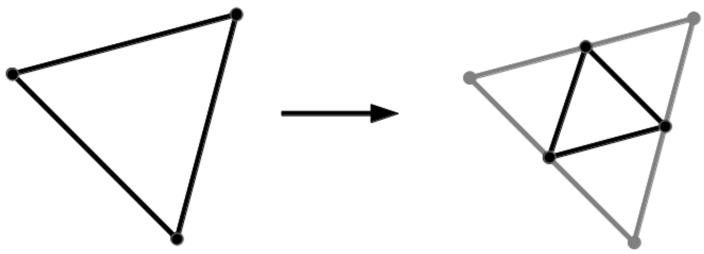


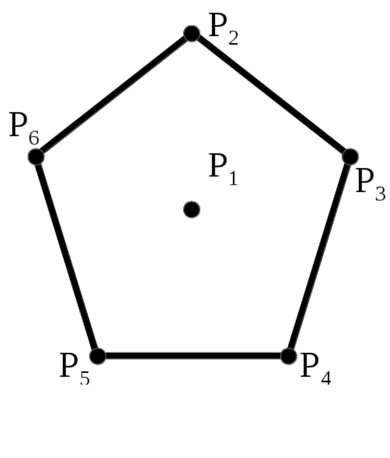
Abbildung 1: Unterteilung eines Dreiecks in vier Dreiecke

Die Funktion soll als Eingabeargumente die Variablen `P, T` und `innerP` haben. Dabei ist `P` die Eingabepunktemenge, `T` die Triangulierung auf dieser Menge und `innerP` ein Binärvektor, welcher angibt, welcher Punkt aus `P` im inneren des Gebiets liegt.
Analog zur Eingabe soll die Funktion als Ausgabe die Variablen `P2, T2` und `innerP2` haben, welche die verfeinerte Triangulierung definieren, sowie die Restriktionsmatrix `R`.
Sie dürfen voraussetzen, dass alle Kanten der groben Triangulierung, deren Punkte auf dem Rand liegen, *Randkanten* sind und das zugrundeliegende Gebiet bereits durch die grobe Triangulierung exakt dargestellt wird.

Nützliche Matlab Befehle: `unique, sort, ismember`.

(b) **Mehrgitterverfahren - V-Zyklus** (3 Punkte)
Wir betrachten als Gebiet Ω das regelmäßige Fünfeck mit den Koordinaten

$$P_1 = (0,0) \text{ als Mittelpunkt und } P_i = (\sin(\alpha_i), \cos(\alpha_i)),\ \alpha_i = \frac{2\pi}{5}(i-2) \text{ für } i = 2, ..., 6.$$



Damit gilt:

$$\Omega = \left\{ \sum_{i=2}^{6} \alpha_i P_i \quad : \quad \sum_{i=2}^{6} \alpha_i < 1 \text{ und } \alpha_i > 0 \text{ für } i = 2, ..., 6 \right\} \text{ ist das Innere des Fünfecks und}$$

$$\partial\Omega = \left\{ \sum_{i=2}^{6} \alpha_i P_i \quad : \quad \sum_{i=2}^{6} \alpha_i = 1 \text{ und } \alpha_i \geq 0 \text{ für } i = 2, ..., 6 \right\} \text{ der Rand.}$$

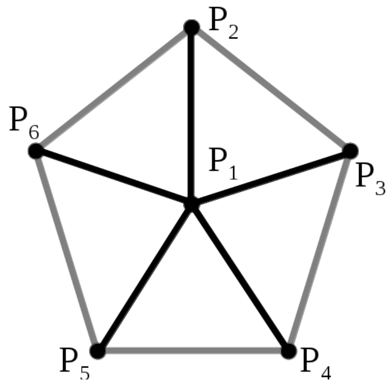
Auf dem Fünfeck betrachten wir das zweidimensionale Poisson-Problem

$$\begin{aligned} -\Delta u(x,y) &= f(x,y) && \text{für alle } (x,y) \in \Omega \\ u(x,y) &= 0 && \text{für alle } (x,y) \in \partial\Omega \end{aligned}$$

mit

$$f(x,y) = 1000\ x\ y\ \left(1 - \frac{1}{4}x^2 - \frac{1}{4}y^2\right).$$

Schreiben Sie eine Funktion `solvepde_fem_mgv`, welche die Triangulierung auf der Punktemenge $P_1, ..., P_6$



durch Aufruf der Funktion `refine_triang` bis zum Level ℓ verfeinert (ℓ-facher Aufruf der Funktion). Stellen Sie auf dem feinsten Gitter das lineare Gleichungssystem zur Lösung der Finite-Elemente-Approximation auf und lösen Sie es mit dem Mehrgitterverfahren (V-Zyklus). Implementieren Sie dazu die rekursive Funktion `MGV` und verwenden Sie als Glättungsoperation die Gauss-Seidl-Iteration mit $\nu_1 = \nu_2 = 3$ Glättungsschritten. Der Startwert $\bar{\mu}^\ell$ für das Mehrgitterverfahren soll dabei, wie in der Vorlesung beschrieben, durch eine „geschachtelte Iteration“ ermittelt werden.
Die Funktion `solvepde_fem_mgv` soll als Eingabeargumente die Variable ℓ haben, welche die Anzahl an Level angibt und die Variable `numV`, welche angibt, wie oft der V-Zyklus hintereinander ausgeführt werden soll.
Als Ausgabe soll die Funktion die Punktemenge des feinsten Gitters, die zugehörige Triangulierung, sowie die Variablen `uFEM_mgv` und `uFEM` haben. Dabei ist `uFEM_mgv` die Lösung, welche das Mehrgitterverfahren zur Approximation der Lösung des Gleichungssystems nutzt und `uFEM` die Lösung, welche den \-Operator von Matlab nutzt.
Plotten Sie die numerische Lösung `uFEM_mgv`, sowie die Differenz im Absolutbetrag zu `uFEM` zu verschiedenen Werten von ℓ und `numV`.

Hinweis: In Matlab können mehrere Datenobjekte wie Matrizen in einem Cell-Array abgespeichert werden. Abspeichern der Matrizen `M1, M2, M3` in einem Cell-Array `C` geschieht beispielsweise durch Eingabe von

```
C = cell(3);
C(1) = {M1}; C(2) = {M2}; C(3) = {M3};
```

Das Laden der Matrizen aus dem Cell-Array geschieht durch

```
M1 = C{1}; M2 = C{2}; M3 = C{3};
```