

Moderne Methoden der numerischen linearen Algebra

<http://www.math.uni-konstanz.de/~rutka/UEBUNGEN/LinAlg/NumLinAlg.html>

Aufgabenblatt 6

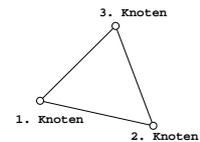
Aufgabe 1: Minimalflächen

Sei $u : \Omega \rightarrow \mathbb{R}$, $\Omega \subset \mathbb{R}^2$ eine Funktion, deren Graph $((x, y), u(x, y))$ die Minimalfläche über Ω mit den Randwerten $u_D : \partial\Omega \rightarrow \mathbb{R}$ beschreibt. Eine Approximation dieser Funktion kann wie folgt konstruiert werden:

1. Als erstes muss der Rand $\partial\Omega$ diskretisiert werden. Seien $(\xi_i, \eta_i) \in \partial\Omega$ für $i = 1, 2, \dots, n$, $n \in \mathbb{N}$, die Eckpunkte des Randpolygons, welches $\partial\Omega$ approximiert.
2. In dem zweiten Schritt wird das Innere des Gebietes Ω diskretisiert.

Innerhalb des Gebietes werden zufällig m Punkte (x_i, y_i) , $i = 1, 2, \dots, m$ generiert. Zusammen mit den Randpunkten bilden sie die Knotenmenge $K := (\cup_{i=1}^m (x_i, y_i)) \cup (\cup_{j=1}^n (\xi_j, \eta_j))$. Jedem Punkt aus K wird ein Index zugewiesen. In der Regel, wird K als eine $\mathbb{R}^{(m+n) \times 2}$ Matrix dargestellt, deren i -te Zeile die Koordinaten des i -ten Knotens enthält.

Nun wird das Gebiet Ω *trianguliert*, das heißt, es wird in sich nicht überlappende dreieckige Untergebiete zerteilt, wobei die Ecken der Dreiecke genau durch die Knoten aus K gegeben sind. Sei M die Anzahl der Dreiecke.

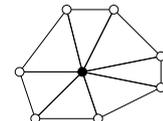


Eine Triangulierung ist durch die Knotenmenge K und eine Triangulierungsmatrix T eindeutig bestimmt. Die Triangulierungsmatrix $T \in \mathbb{N}^{M \times 3}$ hat folgende Bedeutung: die Zahlen in der k -ten Reihe von T sind die Knotenindizes der Ecken des k -tes Dreieckes.

3. Zuletzt benötigen wir die Gleichungen für die approximierte Funktion u . Minimalflächen sind dadurch grob charakterisiert, daß *die Funktionswerte benachbarter Punkte sich nur wenig voneinander unterscheiden*. Wir benutzen eine einfache Approximation dieser Eigenschaft:

$$\begin{cases} u(x_i, y_i) - \text{Mittelwert}(\text{alle Nachbarnknoten}) = 0 & \text{für alle innere Knoten} \\ u(\xi_j, \eta_j) = u_D(\xi_j, \eta_j) & \text{auf dem Rand} \end{cases} \quad (1)$$

Zwei Knoten heißen *benachbart*, wenn sie durch eine Kante eines Dreiecks verbunden sind. In der Abbildung sind alle weißen Punkte Nachbarn des schwarzen Punkts.



Die Lösung des Systems (1) liefert eine Approximation an die Minimalfläche.

Ihre Aufgabe ist es, die Minimalfläche über dem Einheitskreis mit den Randwerten $u_D = 0.2 \sin(5\phi)$ zu finden (ϕ : Azimutalwinkel). Lösen Sie das resultierende System (1) mit dem Matlab Operator “\”.

Einige Bemerkungen zur praktischen Implementierung:

- Die Knoten *innerhalb* des Kreises können wie folgt generiert werden:

1. Zuerst werden $N \in \mathbb{N}$ zufällige Punkte in $(-1, 1) \times (-1, 1)$ generiert. Es empfiehlt sich $N \approx n^2/4$ zu nehmen, wobei n der Anzahl der Eckpunkte des Randpolygons entspricht.
 2. Danach werden von diesen Punkten nur diejenigen ausgewählt, die innerhalb des Einheitskreises liegen (Hinweis: `find` Kommando). Anzahl der Knoten m innerhalb von Ω wird abgezählt.
- Eine Triangulierung liefert das Matlab-Kommando `delaunay`. Zur Visualisierung kann die Funktionen `trimesh` und `trisurf` benutzt werden.
 - Zum Operieren mit Knoten können auch die Kommandos `setdiff` und `unique` sehr hilfreich sein.

Aufgabe 2: Permutation

Visualisieren Sie das Besetzungsmuster der Matrix von Aufgabe 1 (Kommando `spy`). Die Struktur der Matrix soll durch eine Permutation der Indizes verbessert werden. Benutzen Sie den Matlab Befehl `symrcm` und lösen Sie das unnummeriertes Problem.

Vergleichen Sie die Lösung mit dem Ergebnis von Aufgabe 1. Visualisieren Sie die Lösung. Dafür müssen Sie herausfinden, wie man die Knoten passend zur Umbenennung der Matrix-Indizes neu numeriert und die Triangulierungsmatrix T entsprechend anpaßt.

Aufgabe 3: Richardson Iteration

“Packen” Sie das Programm der Aufgabe 2 in eine Funktion, die zu gegebenen $n \in \mathbb{N}$ die Knoten, die Triangulierungsmatrix, und die Matrix sowie die rechte Seite des entsprechenden linearen System liefert.

Lösen Sie das System $Au = b$, $A \in \mathbb{R}^{n \times n}$ mit der vorkonditionierten Richardson Iteration

$$x^{k+1} = (I - PA)x^k + Pb,$$

wobei P die Vorkonditionierungsmatrix ist (siehe Vorlesung) und I die Einheitsmatrix. Nehmen sie die Folgenden Vorkonditionierer:

(A) $P = I$

(B) $P = (\text{diag}(a_{11}, a_{22}, \dots, a_{nn}))^{-1}$ (Jacobi Iteration)

(C) $P = B^{-1}$, wobei

$$B_{ij} = \begin{cases} a_{ij} & \text{falls } i \geq j \\ 0 & \text{falls } i < j \end{cases} \quad (\text{Gauss-Seidel Iteration}).$$

(D) $P = (\frac{1}{\omega} \text{diag}(a_{11}, a_{22}, \dots, a_{nn}))^{-1}$, wobei $\omega \in \mathbb{R}$ der Relaxationsparameter des JOR Verfahren ist.

(E) $P = B^{-1}$, wobei

$$B_{ij} = \begin{cases} \frac{1}{\omega} a_{ii} & \text{falls } i = j \\ a_{ij} & \text{falls } i > j \\ 0 & \text{falls } i < j \end{cases} \quad (\text{SOR Verfahren}).$$

Wieviele Iterationen brauchen Sie, um eine bestimmte Toleranz des Residuums zu erreichen? Welche Werte des Relaxationsparameters ω führen im JOR und SOR Verfahren zu einer besseren Konvergenz?

Visualisieren Sie die Spektren der Iterationsmatrix $M := I - PA$. Nehmen Sie

$$\omega = \{0.1, 0.5, 0.9, 1, 1.2, 1.5, 1.9, 2.1\}.$$

In welchen Fällen ist die Konvergenz nicht zu erwarten?