Prof. Dr. Reinhard Klein
Elena Trunz, Dennis den Brok

Universität Bonn
Institut für Informatik II
2.11.2016

Wintersemester 2016/2017

Übungen zu Angewandte Mathematik: Numerik - Blatt 2

Die Abgabe der Programmieraufgaben (1 und 3) erfolgt per Email bis zum 9.11.2016 um 23:59. Die Abgabe des schriftlichen Teils (Aufgabe 2) erfolgt vor der Vorlesung am Mittwoch, 9.11.2016 um 10:00 (c.t.). Bitte die Gruppennummer mit angeben!

Einleitung In vielen praktischen Situationen stellt sich das Problem, dass Messdaten nur an unregelmäßig verteilten Punkten erhoben wurden und aus diesen Daten nun die entsprechenden Werte an anderen Punkten geschätzt werden sollen. Beispiele dafür sind Wetterdaten, die nur an den jeweiligen Messstationen erfasst wurden. Es gibt eine Reihe von verschiedenen Ansätzen dieses "Scattered Data Interpolation Problem" zu lösen, von denen im Folgenden die so genannten Thin Plate Splines implementiert werden soll. In Aufgabe 1 wird ein Löser für die zugrundeliegenden linearen Gleichungssysteme implementiert. Aufgabe 2 stellt die theoretische Grundlage bereit und in Aufgabe 3 wird dann das eigentliche Verfahren implementiert.

Aufgabe 1 (Lineare Gleichungssysteme mit SVD lösen, *4 Punkte*)

Implementiere eine Funktion `LinearSolve(A,b)`, die zu einem gegebenen linearen Gleichungssystem $A \cdot x = b$ mit $A \in \mathbb{C}^{m \times n}$ und $b \in \mathbb{C}^m$ die Lösung $x \in \mathbb{C}^n$ (wir nehmen an, dass diese existiert) berechnet und zurückgibt. Dazu solltest du die folgenden Teilaufgaben lösen.

a) Berechne die SVD von A. Verwende dazu die Eigenwertzerlegung von $A^T A$.
b) Schreibe eine Funktion `PseudoInverse(A)`, die die Pseudo-Inverse A^+ von A mit Hilfe ihrer SVD berechnet.
c) Nutze die Pseudo-Inverse und schreibe damit die Funktion `LinearSolve(A,b)`. Dazu kann das lineare Gleichungssystem als lineares Ausgleichsproblem $\|A \cdot x - b\|_2 \to \min$ angesehen werden.

Hinweis: Um eine Eigenwertzerlegung durchzuführen kann in Matlab `eig` und in Python `numpy.linalg.eig` verwendet werden.

Aufgabe 2 (Thin Plate Spline fitten, schriftlich, *4 Punkte*)

Thin Plate Splines können genutzt werden um Messpunkte $(x_1, y_1)^T, \ldots, (x_m, y_m)^T \in \mathbb{R}^2$ und Messwerte $z_1, \ldots, z_m \in \mathbb{R}$ exakt zu interpolieren, d.h. es wird eine Funktion $f : \mathbb{R}^2 \to \mathbb{R}$ erzeugt, die

$$\forall i \in \{1, \ldots, m\} : f(x_i, y_i) = z_i$$

erfüllt. Sie simulieren das Biegungsverhalten dünner Metallplatten, die durch die gegebenen Punkte gehen (daher der Name). Zur Konstruktion wird die Funktion $g(r) := r^2 \cdot \log r$ genutzt und f wird definiert wie folgt:

$$f(x, y) := \sum_{j=1}^{m} w_j \cdot g\left(\left\| \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x_j \\ y_j \end{pmatrix} \right\|_2\right)$$

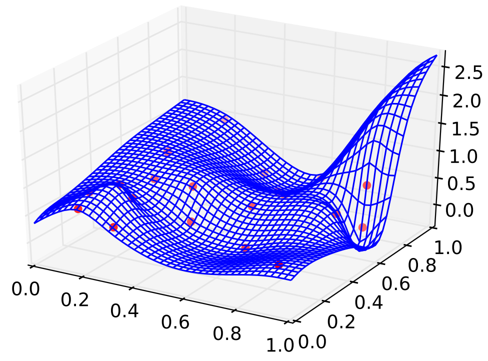

Abbildung 1: Zufällig gewählte Messpunkte (rot) und der resultierende Thin Plate Spline (blau).

Dabei ist $w \in \mathbb{R}^m$ ein zu bestimmender Koeffizientenvektor. Sei $A = (a_{i,j}) \in \mathbb{R}^{m \times m}$ mit

$$a_{i,j} = g\left(\left\| \begin{pmatrix} x_i \\ y_i \end{pmatrix} - \begin{pmatrix} x_j \\ y_j \end{pmatrix} \right\|_2\right).$$

Zeige, dass

$$A \cdot w = (z_1, \dots, z_m)^T \quad \Leftrightarrow \quad \forall i \in \{1, \dots, m\} : f(x_i, y_i) = z_i.$$

Aufgabe 3 (Thin Plate Splines, *4+4=8 Punkte*)

Nun soll an zufällige Daten ein Thin Plate Spline gefittet werden. Das bereitgestellte Framework produziert bei einer richtigen Lösung eine Ausgabe ähnlich der in Abbildung 1 (je nachdem, was für Zufallszahlen generiert werden).

a) Schreibe eine Funktion `ComputeTPSWeights(X,Y,Z)`, die zu gegebenen Messungen den Koeffizientenvektor w für den Thin Plate Spline berechnet.

   Hinweis: Zum Lösen linearer Gleichungssysteme kann entweder Aufgabe 1 verwendet werden oder der Matlab operator \ bzw. `numpy.linalg.solve`.

   Warnung: Es kann Probleme geben, wenn bei der Berechnung der Diagonaleinträge von A $\log(0)$ evaluiert wird. Die Funktion $g(r)$ hat aber einen wohldefinierten Limes in 0. Bei Bedarf kann r durch $\max\{r, 10^{-8}\}$ ersetzt werden.

b) Schreibe eine Funktion `EvaluateTPSSpline(XNew,YNew,X,Y,Weights)`, die unter Verwendung der Ein- und Ausgabe `X`, `Y`, `Weights` von `ComputeTPSWeights(...)` den Thin Plate Spline f an den durch `XNew` und `YNew` gegebenen Stellen evaluiert.