Karlsruher Institut für Technologie
Institut für Algebra und Geometrie
PD Dr. T. Arens
Dipl.-Math. S. Heumann

Karlsruhe, den 21.10.2010

Inverse Probleme – Matlab Übungen

Die Hilbert-Matrix

Die $n \times n$-Hilbert-Matrix ist definiert als

$$H_n = \left(\frac{1}{j+k-1} \right)_{j,k=1,\ldots,n} .$$

H_n ist eine symmetrische, positiv-definite Matrix. Sie ist außerdem ein klassisches Beispiel für eine schlecht konditionierte Matrix. Man kann zeigen, dass die Konditionszahl mit n exponentiell wächst.

In Matlab stellen Sie eine Hilbertmatrix mit `hilb(n)` auf. Die Konditionszahl (bezüglich der Spektralnorm) einer Matrix `A` berechnen Sie mit `cond(A)`.

Aufgabe 1: Stellen Sie Hilbert-Matrizen für verschiedene Werte von n auf und speichern Sie diese in Matlab-Variablen ab. Berechnen Sie die Konditionszahlen dieser Matrizen.

Lineare Gleichungssysteme mit Hilbert-Matrizen

Sind eine $n \times n$-Matrix `A` und ein n-Spaltenvektor `y` in Matlab definiert, so kann das lineare Gleichungssystem $Ax = y$ durch den Befehl

```
x = A \ y
```

gelöst werden (LU-Zerlegung, im Falle symmetrischer Matrizen Cholesky-Zerlegung). Die Notation soll an eine „Division von Links“ erinnern.

Wir wollen uns einen Lösungsvektor vorgeben und die zugehörige rechte Seiten des linearen Gleichungssystems bestimmen. Anschließend soll versucht werden, die Lösung durch Lösen des LGS zurückzugewinnen. Zusätzlich wollen wir die rechte Seite durch einen Fehler stören.

Aufgabe 2: Stellen Sie mit dem Befehl `ones(n,1)` Spaltenvektoren auf, deren Koeffizienten alle 1 sind. Dabei entspricht n gerade den Werten, die Sie in Aufgabe 1 gewählt haben. Mit der Matrizenmultiplikation `y = A * x` berechnen Sie die rechten Seiten der zugehörigen linearen Gleichungssysteme. Zu `y` addieren Sie zufällige Fehler vom Betrag höchstens 10^{-7}:

```
y_delta = y + 1e-7 * (-1 + 2*rand(size(y)))
```

Die Funktion `rand` erzeugt eine Zufallsmatrix der angebenen Größe mit Koeffizienten im Intervall $[0, 1]$.

Lösen Sie nun die linearen Gleichungssysteme mit `y_delta` und vergleichen Sie mit dem exakten Ergebnis. Verwenden Sie auch die „exakten“ Vektoren `y`.

Woher kommt das Aufblähen der Fehler?

Die Hilbert-Matrix H_n ist symmetrisch. Daher besitzt der $\mathbb{R}^n$ eine Orthonormalbasis aus Eigenvektoren von H_n. Ist V_n eine (dann notwendig orthogonale) Matrix, die diese Eigenvektoren als Spalten besitzt und D_n eine Diagonalmatrix, die die Eigenwerte in der entsprechenden Reihenfolge auf der Diagonalen hat, so gilt die Darstellung

$$H_n = V_n \, D_n \, V_n^\top \, .$$

Die Inverse H_n^{-1} ergibt sich wegen der Orthogonalität von V_n als

$$H_n^{-1} = V_n \, D_n^{-1} \, V_n^\top \, .$$

In Matlab erhält man diese Matrizen durch die Zuweisung `[V,D] = eig(H)`. Die Transponierte von `V` wird durch `V.'`, die Inverse von `D` durch `inv(D)` berechnet. Alternativ erhält man durch `d = diag(D)` einen Vektor `d` mit den Diagonaleinträgen von `D`. Mit `f = 1./d` erzeugt man den Vektor mit den Kehrwerten als Einträgen und mit `diag(f)` erhält man daraus die Inverse von `D`.

Aufgabe 3: Bestimmen Sie V_n und D_n für die bisher gerechneten Fälle und untersuchen Sie diese Matrizen. Diskutieren Sie anhand dieser Matrizen, wodurch es bei der Lösung der linearen Gleichungssyteme zur Fehlerverstärkung kommt.

Aufgabe 4: Überlegen Sie sich Strategien, wie Sie die Fehlerverstärkung abmildern und auch bei den sehr schlecht konditionierten Fällen eine sinnvolle Lösung erreichen können.

Hinweis: Überlegen Sie sich zunächst mit Papier und Stift genau, wie Ihre Strategie aussehen soll. Bei der Umsetzung in Matlab helfen wir Ihnen anschließend gerne.