Fachbereich Mathematik & Informatik
Freie Universität Berlin
Prof. Dr. Ralf Kornhuber, Max Kahnt

4. Übung zur Vorlesung

Numerik I

SoSe 2019

`http://numerik.mi.fu-berlin.de/wiki/SS_2019/NumerikI.php`

Abgabe: Fr., 17. Mai 2019, 16:00 Uhr

1. Aufgabe (2 TP)
Es seien die Punkte $(u_1, v_1) = (1, \frac{\sqrt{7}}{2})$, $(u_2, v_2) = (0, \frac{\sqrt{15}}{4})$ und $(u_3, v_3) = (\frac{2}{\sqrt{3}}, \frac{1}{2})$ gegeben. Von den Punkten weiß man, dass sie einer Kreisgleichung der Form

$$u^2 + v^2 = r^2$$

genügen sollten. Bestimmen Sie den Parameter $r > 0$ optimal im Sinne der kleinsten Fehlerquadrate.

2. Aufgabe (2 TP)
Sei $x = (9, 2, 6)$ und $y = (-11, 0, 0)$.

a) Finden Sie die Householder-Reflexion H mit $Hx = y$.

b) Finden Sie von Null verschiedene Vektoren u und v mit

$$Hu = -u \qquad \text{und} \qquad Hv = v.$$

3. Aufgabe (2 TP)
Zeigen Sie: Sei $A \in \mathbb{R}^{m \times n}$, $m \geq n$, Q die orthogonale Matrix einer QR-Zerlegung von A, wobei A vollen Rang habe. Dann bilden die ersten n Spalten von Q eine Orthonormalbasis des Bildes $R(A)$ von A.

4. Aufgabe (6 PP)

a) Implementieren Sie in `matlab` Funktionen mit den Signaturen

```
[Q,R] = qr_givens(A),
[Q,R] = qr_householder(A),
```

die zu einer beliebigen Matrix $A \in \mathbb{R}^{m \times n}$ QR–Zerlegungen mittels Givens–Rotationen bzw. Householder–Reflexionen berechnen.

Hinweis: Versuchen Sie Ihre Routinen effizient implementieren. Das bedeutet, dass Sie beispielsweise keine Operationen auf Matrix–Einträge vornehmen, die dadurch generell nicht verändert werden, und dass Sie keine Matrix–Matrix–Multiplikationen durchführen, wenn die zugehörige Operation auch effizienter durch Matrix–Vektor–Multiplikation realisiert werden kann.

b) Testen Sie Ihre Implementationen aus Aufgabenteil a) und b), indem Sie QR–Zerlegungen für folgende Matrizen berechnen und das Produkt $Q \cdot R$ mit der jeweiligen Eingabematrix vergleichen:

   i) `A = magic(4)`,

   ii) `B = eye(4)`,

   iii) `C = A(:,2:3)`,

   iv) `D = A(2:3,:)`.

   Verifizieren Sie jeweils, dass Q eine Orthogonalmatrix und R eine obere Dreiecksmatrix ist.

   Hinweis: In der Vorlesung wird vor allem $m \geq n$ betrachtet. Hier wird aber auch $m < n$ getestet.

c) Wir betrachten die Hilbert– und Wilkinson–Matrizen $A = H, W \in \mathbb{R}^{n \times n}$, $n \in \{10, 20, \ldots, 100\}$. Setzen Sie jeweils $x_0 = (1, \ldots, n)^T$, sowie $b = Ax_0$. Verwenden Sie jeweils die beiden QR–Zerlegungen mittels Ihrer Implementationen aus Aufgabenteil a), sowie die LR–Zerlegung mittels der eingebauten Matlab-Implementation `lu` um das resultierende Gleichungssystem $Ax = b$ zu lösen. Geben Sie jeweils die Kondition der Matrizen A und ihrer Zerlegungen an sowie den jeweiligen relativen Fehler bezüglich der euklidischen Norm. Beschreiben und interpretieren Sie die Ergebnisse.

   Hinweis: Die Hilbert–matrix ist in `matlab` aufrufbar mittels `hilb(n)`. `matlab` implementiert auch eine Funktion `wilkinson(n)` mit teilweise ähnlichen Eigenschaften, aber anderer Darstellung als in der Vorlesung für die Wilkinson–Matrix gegeben. Implementieren Sie für diese Aufgabe selbst die Matrix analog zum Skript. Außerdem dürfen Sie zur Berechnung der Lösung mittels einer Zerlegung den `matlab`-Backslash-Operator verwenden.

Allgemeine Hinweise

Die Punkte unterteilen sich in Theoriepunkte (TP) und Programmierpunkte (PP). Bitte beachten Sie die auf der Vorlesungshomepage angegebenen Hinweise zur Bearbeitung und Abgabe der Übungszettel, insbesondere der Programmieraufgaben.