Prof. Dr. Frank Noé
Dr. Christoph Wehmeyer
Tutoren:
Florian Litzinger (Di. 12-14; Di. 16-18)
Maikel Nadolski (Mi. 12-14; Mi. 14-16)

2. Übung zur Vorlesung **Numerik I**

Abgabe: Freitag, 2. Mai 2014, 10:00 Uhr, Tutorenfächer

Aufgabe 1 *(Spaltenpivotsuche, 3 + 2 P)*:

a) Implementieren Sie die LR-Zerlegung mit Spaltenpivotsuche. Schreiben Sie also eine Funktion $[LR, perm, cond] = \textbf{LRDecompPivot}(A)$, welche die LR-Zerlegung mit Spaltenpivotsuche einer Matrix A durchführt. Wie zuvor ist LR eine Matrix, deren oberer Teil die Matrix R, und deren unterer Teil die Matrix L repräsentiert. Außerdem ist $perm$ ein Vektor der Länge n, der die vollständige Permutation der Zeilenindizes wiedergibt, d.h. $perm(k)$ gibt an, in welche Zeile die k-te Zeile der Matrix durch die Anwendung des Algorithmus verschoben wurde. Weiterhin ist $cond$ ein Vektor der Länge $n-1$, der die Konditionen der Restmatrizen $A^{(k)}(k+1 : end, k+1 : end)$ nach jedem Eliminationsschritt enthält. Schreiben Sie außerdem eine Funktion $x = \textbf{SolveLRPivot}(b, LR, perm)$, welche dann das entsprechende LGS unter Nutzung der Permutation $perm$ löst.

b) Lösen Sie mit Hilfe der obigen Methoden das lineare Gleichungssystem für die Vandermonde Matrix vom letzten Übungszettel, mit $n = 60$. Vergleichen Sie die Ergebnisse mit der Lösung ohne Pivotsuche. Wie verhalten sich die Koeffizientenvektoren? Modifizieren Sie Ihre Funktion **LRDecomp** vom letzten Mal so, dass auch hier die Konditionen der Restmatrizen zurückgegeben werden. Plotten sie die Konditionen aus beiden Methoden. Werten Sie auch wieder das Interpolationspolynom für beide Methoden auf einem feinen Gitter aus und plotten Sie die Lösungen. Was können Sie feststellen?

Aufgabe 2 *(spd-Matrizen, 3P)*:

Sei A eine symmetrisch, positiv definite Matrix. Zeigen Sie:

a) Jede Untermatrix der Form

$$A^{(ij)} := \begin{pmatrix} a_{ii} & a_{ij} \\ a_{ji} & a_{jj} \end{pmatrix}$$

ist ebenfalls positiv definit.

b) Es gilt:

$$
\begin{aligned}
|a_{ij}| &\leq \sqrt{a_{ii}a_{jj}} \\
&\leq \frac{1}{2}(a_{ii} + a_{jj})
\end{aligned}
$$

c) Folgern Sie daraus:

$$
\max_{i,j} |a_{ij}| = \max_i a_{ii}
$$

Aufgabe 3 *(Cholesky-Zerlegung, 2P):*

Implementieren Sie den Algorithmus zur Cholesky-Zerlegung, d.h. schreiben Sie eine Funktion $L = \textbf{Cholesky}(A)$, die eine Matrix A in ihre Cholesky-Zerlegung L umwandelt. Sie können die neu berechneten Elemente in der alten Matrix speichern. Wenden Sie diese Funktion dann auf die Matrix

$$
A = \begin{pmatrix}
2 & -1 & 0 & \ldots & 0 \\
-1 & 2 & -1 & \ldots & 0 \\
\vdots & & & & \vdots \\
0 & \ldots & -1 & 2 & -1 \\
0 & \ldots & 0 & -1 & 2
\end{pmatrix}
$$

an. Diese Matrix hat also stets den Wert 2 auf der Diagonalen und -1 auf den ersten beiden Nebendiagonalen. Berechnen Sie die Cholesky-Zerlegung von A für $n = 100, 500, 1000, 1500, 2000, 2500, 3000$ und stoppen Sie die dafür benötigte Zeit. Dazu können Sie die Matlab Funktionen **tic** und **toc** benutzen. Plotten Sie die Größe der Matrix gegen die benötigte Zeit.

Bemerkung: Die Matrix A ist ein Standard-Beispiel, das bei der Lösung partieller Differentialgleichungen auftritt. Die Aufgabe soll Ihnen ein Gefühl dafür geben, dass man bei wirklich großen, aber dünn besetzten Matrizen auf iterative Verfahren zurückgreifen muss, da herkömmliche Methoden schlichtweg zu aufwändig sind.