

Tutorial: Lineare Gleichungssysteme

(basierend auf dem Skript Numerik für Ingenieure von R. Axthelm)

Wir möchten das lineare Gleichungssystem der Form $Ax = b$ lösen. Wir gehen davon aus, dass die Matrix A eine $n \times n$ Matrix mit vollem Rang ist, d.h. das Gleichungssystem ist eindeutig lösbar. Von der linearen Algebra wissen sie, dass sie das System auf $x = A^{-1}b$ umformen können. Sie werden in den folgenden Aufgaben sehen, dass das Berechnen der Inversen rechnerisch aufwändig ist und daher zum Lösen eines linearen Gleichungssystem nicht verwendet wird.

Schritt A: Zeitmessung für Inverse und LGS

1. Starten sie ihren Computer von der Command Line mit dem folgenden Befehl: `matlab -singleCompThread`. Alternativ starten sie Matlab wie gewohnt und geben `maxNumCompThreads(1)` ein. Matlab wird eine Warnung herausgeben, dass die Funktion in zukünftigen Versionen nicht mehr vorhanden sein wird. Das soll sie im Moment nicht weiter stören.
Neuere Computer haben mehrere Prozessoren. Standardmässig verwendet Matlab alle zur Verfügung stehenden Prozessoren. Das wird ihre Zeitmessung durcheinanderbringen, deshalb beschränken wir uns in dieser Aufgabe auf einen Prozessor.
2. Mit dem Befehl `inv(A)` können sie die Inverse von A berechnen. Mit `tic` und `toc` können sie die Berechnungszeit messen. Der Befehl `tic` startet die Messung, während `toc` die Messung stoppt und den Wert retourniert. Schreiben sie eine Funktion `MeasureCalcTime(n)`, die ihnen die Zeit berechnet, die Matlab benötigt, um eine Matrix A der Dimension $n \times n$ zu invertieren. Sie erstellen A mit dem Befehl `A=magic(n)`; . Für b verwenden sie einen Spaltenvektor mit Einseinträgen.
3. Schreiben sie eine Funktion `Timing()`, die ihre Funktion `MeasureCalcTime(n)` für verschiedene Werte von n aufruft und die entsprechenden Zeiten für jedes n abspeichert. Füllen sie anschliessend die Werte in Tabelle 1 ein.
4. Überprüfen sie, ob die Berechnungszeiten proportional zu n^3 sind.
5. Ändern sie ihre Funktion so ab, dass sie nun die Zeit für die Lösung des linearen Gleichungssystem messen (Matlab-Befehl: `A\b`).
6. Lassen sie ihre Funktion `Timing` laufen und befüllen sie Tabelle 2 .
7. Überprüfen sie, ob die Werte proportional zu n^3 sind.

Tabelle 1: Zeit für die Berechnung der invertierten Matrix

n	Zeit	Verhältnis zum vorigen n
161		-
321		
641		
1281		
2561		
5121		
10241		

Tabelle 2: Zeit für LU-Zerlegung und Lösen des linearen Gleichungssystems

n	Zeit	Verhältnis zum vorigen n
161		-
321		
641		
1281		
2561		
5121		
10241		

8. Vergleichen sie die beiden Tabellen 1 und 2 miteinander und schreiben sie ihre Resultate in Tabelle 3.

Tabelle 3: Vergleich Zeitmessung

n	Zeit für Inverse / Zeit für lin. Gl. Sys.
161	
321	
641	
1281	
2561	
5121	
10241	

9. Die Theorie besagt, dass das Berechnen der Inversen etwa 3 bis 4 (je nach Optimierung) Mal so lange geht wie das Lösen des Gleichungssystems. Stimmen ihre Resultate mit der Theorie überein?

Fazit:

Berechnen sie NIE die Inverse, wenn sie ein lineares Gleichungssystem lösen wollen.

Schritt B: Dünnbesetzte Matrizen

10. Wir interessieren uns nun dafür, ob das Lösen eines linearen Gleichungssystem immer proportional zu n^3 ist. Dazu rufen sie den Befehl `gallery('tridiag',n)` auf und betrachten die entstandene Matrix. Wodurch unterscheidet sie sich von ihren bisher erstellten Matrizen?
11. Befüllen sie die Tabelle 4 für die Matrix aus der Gallery.

Tabelle 4: Zeit für LU-Zerlegung und Lösen des linearen Gleichungssystems

n	Zeit	Verhältnis zum vorigen n
161		-
321		
641		
1281		
2561		
5121		
10241		

12. Welchen Wert erhalten sie für p in n^p

1-3,5-6,8, 10,11

Siehe m-Files.

4, 7, 9,12

Die Zeiten sind proportional zu n^3 . Wenn sie in der Tabelle das Verhältnis berechnen erhalten sie näherungsweise einen Faktor 8, da $(2*n)^3/n^3 = 2^3$ ergibt. Das Berechnen der Inversen dauert, je nach Implementation ca. 3 bis 4 Mal länger als das Lösen des LGS. $N_{LU}(n) = \frac{n^3}{3} + n^2 - \frac{n}{3}$ und $N_{Inv}(n) = \frac{4n^3}{3} + \frac{n}{3}$

11

Mit dem Befehl erzeugen sie eine dünnbesetzte Matrix (sparse). Es werden nur die Element der Tridiagonal abgespeichert. Alle andern Null-Einträge werden nicht gespeichert. Die Algorithmen, die Matlab für diese Matrizen verwendet, sind deutlich effizienter als für eine volle Matrix.

13

Das LGS kann in diesem Fall mit $p = 1$ gelöst werden.