



1.2. Übung. Einführung in die Programmierung (MA 8003)

Hinweis: Alle Aufgaben können ohne Schleifen gelöst werden.

Aufgabe 1.2.1: Der Befehl `[L,R] = lu(A)` erzeugt eine Zerlegung einer invertierbaren Matrix A in eine obere und untere Dreiecksmatrix. Es gilt $A = LR$. Sei $A = 2*\text{eye}(10) + \text{rand}(10)$ eine Zufallsmatrix und $b=2*\text{rand}(10,1)$ ein Zufallsvektor. Erzeugen Sie die LR-Zerlegung mit dem obigen Befehl.

- Lösen Sie das Gleichungssystem $LRx=b$ mit dem Backslash-Operator ohne die Matrix A zu verwenden.
- Lösen Sie dasselbe System nun durch zweimaliges Anwenden des Backslash-Operators und ohne dabei das Produkt LR zu bilden.

Überprüfen Sie Ihre Ergebnisse.

Aufgabe 1.2.2: Mit Hilfe des Befehls `inv` kann die Inverse einer quadratischen Matrix bestimmt werden. Diesen Befehl sollte man allerdings nur dann verwenden, wenn die Inverse explizit benötigt wird und nicht zum Lösen von linearen Gleichungssystemen.

Erzeugen Sie mit `R=rand(1500); r=0.5*(1:1500)'`; eine Matrix und einen Vektor mit Zufallseinträgen. Lösen Sie das LGS $Rx = r$ einmal mit der Inversen und einmal mit dem Backslash-Operator. Messen Sie die Zeit, die jeweils zum Lösen benötigt wird, mit `tic` und `toc` (vgl. Hilfe).

Aufgabe 1.2.3: Gegeben sei eine invertierbare 10×10 -Matrix A . Verwenden Sie den Backslash-Operator, um mit einem Befehl die Inverse von A zu berechnen und überprüfen Sie das Ergebnis.

Aufgabe 1.2.4 (*): Lösen Sie die erste Aufgabe von *Project Euler* (<http://projecteuler.net>): *Add all the natural numbers below one thousand that are multiples of 3 or 5.*

Tipp: Die Lösung ist 233168.

Aufgabe 1.2.5: Lösen Sie die sechste Aufgabe von *Project Euler*:

Find the difference between the sum of the squares of the first one hundred natural numbers and the square of the sum.

Tipp: Die Lösung ist 25164150.

Aufgabe 1.2.6 (*): Gegeben sei ein Vektor v mit allen Zahlen von 1 bis n nicht sortiert, in dem eine zufällige Zahl fehlt.

- a) Wie kann man mit Matlab einen solchen Vektor schnell erzeugen?

Tipp: Suchen Sie in der Hilfe nach „permutation“

- b) Lassen Sie sich mit einer Matlab-Zeile die fehlende Zahl eines solchen Vektors ausgeben.

Aufgabe 1.2.7: Schreiben Sie eine Matlab-Zeile, die für eine vorgegebene ganze Zahl n die Fakultät $n!$ berechnet. Benutzen Sie dabei nicht die Matlab-Funktion `factorial`.

Aufgabe 1.2.8: Erstellen Sie eine beliebige 2×2 Matrix **A** und eine 4×4 Matrix **B**. Addieren Sie **A** zu einer 2×2 Submatrix von **B**.

Aufgabe 1.2.9: Erzeugen Sie einen Vektor $\mathbf{r} = 1:10$. Ändern Sie \mathbf{r} mit einem Befehl so,

- a) dass die Zahlen in umgekehrter Reihenfolge in \mathbf{r} stehen,
- b) dass die ersten fünf Einträge mit den letzten fünf Einträgen vertauscht werden.

Funktioniert Ihre Methode auch für Vektoren beliebiger gerader Länge? Schreiben Sie Ihre Methode ggf. längenunabhängig um.

Aufgabe 1.2.10 (*): Sei \mathbf{v} ein Zeilenvektor. Geben Sie eine andere Möglichkeit an, die Ausgabe des Befehls `repmat(v,15,1)` zu erzeugen ohne den Befehl `repmat` zu verwenden.

Aufgabe 1.2.11 (*): Gegeben sei eine 10×10 -Matrix **R** mit Zufallseinträgen. Setzen Sie die Diagonalelemente von **R** auf den Wert 1.

Aufgabe 1.2.12: Erzeugen Sie mit Hilfe des Doppelpunktoperators und geeigneter Indizierung einen Vektor \mathbf{a} mit den Zahlen zwischen 0 und 20, die folgenden Sequenzen entsprechen:

- a) 0,0,2,2,4,4,6,6,8,8,...
- b) 0,1,3,4,6,7,9,10,12,...

Aufgabe 1.2.13 (*): Erzeugen Sie möglichst einfach die 10×10 -Matrix **A** mit Einträgen

- a) $a_{ij} = 10 * j + i$ **Tipp:** lineares Indizieren!
- b) $a_{ij} = i * j$ **Tipp:** Geeignetes Produkt
- c) $a_{ij} = i + j$ **Tipp:** `repmat`

Aufgabe 1.2.14 (*): Folgender Code stammt aus einer Matlab-Funktion:

```
m = max([1; n(:)]);  
N = [1 1 cumprod(2:m)];  
n(:) = N(n+1);
```

Finden Sie heraus, was dieser Code macht. Aus welchem Raum sollten die Werte in \mathbf{n} stammen, damit dieser Code seine Aufgabe sinnvoll erledigt?

Aufgabe 1.2.15: Erweitern Sie möglichst schnell eine nicht quadratische Matrix um so viele 0-Zeilen bzw. -Spalten, dass sie quadratisch wird.

Aufgabe 1.2.16: Erstellen Sie eine 4×4 Matrix **A** mit dem Befehl `magic` und berechnen Sie die Summe der Spalten, die Summe der Zeilen und die Summe der Diagonale.

Aufgabe 1.2.17: Gegeben sei der Vektor $x=1:12$. Machen Sie sich anhand der Beispiele `reshape(x,3,4)` und `reshape(x,6,2)` klar, wie der `reshape`-Befehl die Zahlen anordnet. Was macht `reshape(x,3,[])`? Wie kann man mit Hilfe des `reshape`-Befehls aus x die folgende Matrix erzeugen:

| | | | |
|---|----|----|----|
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 |