



Numerische Lineare Algebra - Kurzeinführung in Matlab

Hinweise:

- Die Aufgaben auf diesem Blatt sind zum großen Teil dem WiMa-Praktikum aus dem SS 2012 entnommen. Auf der entsprechenden Webseite des Instituts für Numerische Mathematik finden Sie auch noch mehr Aufgaben mit Lösungen zum Üben.
- Matlab bietet eine sehr umfangreiche Dokumentation. Für eine gegebene Funktion, z. Bsp. liefern `help round` und `doc round` nützliche Informationen zu der Funktion `round`.
- Matlab speichert Ergebnisse zwischen. Das kann u. U. zu Fehlern führen. Daher sollten Sie vor jedem neuen Aufruf eines Skripts `clear all` aufrufen.

Aufgaben:

1. Einfache Zuweisungen und Schleifen

Im Folgenden beschäftigen wir uns mit `while` und `for` Schleifen.

- Schreiben Sie eine *Funktion*, die zu einer gegebenen positiven reellen Zahl x die kleinste ganze Zahl $n \in \mathbb{N}$ mit $n \geq x$ berechnet. Verifizieren Sie Ihr Ergebnis mithilfe der Funktion `ceil`.
- Schreiben Sie ein *Skript*, das eine ganze Zahl m von der Eingabe einliest (Stichwort: `input`) und für alle Zahlen $1, \dots, m$ die Anzahl der ganzzahligen Teiler der jeweiligen Zahl berechnet und in einem Vektor b der Länge m speichert.
- Schreiben Sie ein Skript, das folgendes Spiel simuliert: Sie gehen ins Casino und spielen Roulette (Zahlen(0 - 36)). Dazu haben Sie sich das folgende System überlegt: Sie setzen 2€ am Anfang auf Gerade (2,4,6,...). Wenn Sie richtig getippt haben, erhalten Sie das Doppelte Ihres Einsatzes und verlassen damit das Casino. Wenn Sie falsch getippt haben, ist Ihr Einsatz verloren; dann verdoppeln Sie im folgenden Spiel Ihren Einsatz und setzen wieder auf Gerade. Sie wiederholen diese Prozedur solange, bis Sie das erste Mal gewonnen haben und Sie das Doppelte Ihres Einsatzes erhalten (und das Casino verlassen). Wie lange mussten Sie spielen und wie viel Geld haben Sie insgesamt eingesetzt bzw. gewonnen? (Hinweis: Verwenden Sie `randi(37)-1` zur Simulation des Ergebnis eines Roulettespiels).

2. Elementare Matrixoperationen

Wir betrachten die Matrix und den Vektor

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 1 \end{pmatrix}, \quad b := \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}.$$

Legen Sie die Matrix A und den Spaltenvektor b an. Was ist das Ergebnis der folgenden Ausdrücke:

- `b * b`, `b .* b`, `b .^3` und `b ./ b`
- `A * b`
- `A \ b`, `b / A` und `b' / A` (funktioniert analog auch für Matrizen).
- `A'` und `b'`
- `A ./ A'`

An dieser Stelle sei auch auf die Matlab-Dokumentation verwiesen:

3. Zugriff auf (Teil-)Matrizen

Die Befehle

```
X=magic(6);  
b=linspace(1,6,6);
```

erzeugen ein 6×6 magisches Quadrat X und einen Vektor b . Was ist das Ergebnis der folgenden Ausdrücke?

- (a) $X(3,2)$
- (b) $b(b)$
- (c) $X(b', \text{end})$
- (d) $X(:,4)$
- (e) $X(1,b)$
- (f) $X(b(1),b(3))$
- (g) $X([4,5])$
- (h) $X(3:2:9)$
- (i) $X(1,1:2:5)$
- (j) $X(6:-2:2,1:3:4)$

4. Verändern und Zusammensetzen von Matrizen

Erzeugen Sie eine 5×5 -Zufallsmatrix A mit Werten zwischen 0 und 10 mithilfe der Matlabfunktionen `round` und `rand`. Führen Sie die folgenden Operationen durch:

- (a) Ersetzen Sie die 2te Spalte durch die Zahlen 1 bis 5.
- (b) Tauschen Sie die 3te Spalte mit der 4ten Spalte.
- (c) Setzen Sie alle Elemente auf der Hauptdiagonalen auf 0.
- (d) Löschen Sie die 4te Zeile.
- (e) Ersetzen Sie das 1te Element der 5ten Spalte durch 42.
- (f) Fügen Sie vor die erste Spalte der Matrix eine weitere Spalte hinzu, deren Einträge alle 0 sind.

5. Plotten von Funktionen

$$f : x \rightarrow \cos(x) \quad \text{und}$$
$$h : x \rightarrow \frac{x^2}{6} - 3.$$

- (a) Stellen Sie f und g auf dem Intervall $[-2\pi, 2\pi]$ graphisch dar.
- (b) Plotten Sie zusätzlich noch den Einheitskreis mit dem Radius 2.

Hinweis: Mit dem Befehl `hold on` wird das Aktuelle Graphikfenster offen gehalten. Es ist damit möglich mehrere Plots in einen Graph zu zeichnen. Mit `figure` kann ein neues Fenster für einen neuen Plot erzeugt werden.