EBERHARD KARLS UNIVERSITÄT TÜBINGEN

MATHEMATISCH-NATURWISSENSCHAFTLICHE FAKULTÄT

Fachbereich Geowissenschaften

Einführung in Matlab

3. Matrizen

Prof. Dr. Christiane Zarfl, Dipl.-Inf. Willi Kappler

Sie wissen bereits...

- wie Sie in Matlab mit Vektoren rechnen.
- wie Sie Ergebnisse/Daten in 2D-Grafiken darstellen.
- wie Sie durch Scripte/Programme Befehlsfolgen wiederverwertbar machen.

Wie kann ich mit Matrizen rechnen?
Und wie kann ich 3D-Grafiken erstellen?

Nach diesem dritten Block...

- rechnen Sie sicher mit Matrizen.
- können Sie lineare Gleichungssysteme durch in Matlab implementierte Matrizenfunktionen einfach berechnen.
- können Sie zweidimensionale Daten darstellen:
    - in einer 3D-Abbildung
    - in einem Oberflächenplot ("Höhenlinien")

Matrizen

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

- Matrizen sind zweidimensionale Zahlenfelder, die man wie Tabellen schreiben kann.
- Matrizen sind nützlich für:
  - die Beschreibung tensorieller Eigenschaften (z.B. optischer Brechungsindex, magnetische Suszeptibilität, Spannungstensor).
  - die Formulierung linearer Gleichungssysteme.
  - die Diskretisierung zweidimensionaler Daten

Beispiel: lineares Gleichungssystem

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

$$\begin{aligned} 3x_1 + 4x_2 - 2x_3 &= 4 \\ -1x_1 + 2x_2 + 8x_3 &= -1 \\ 2x_1 - 5x_3 &= 3 \end{aligned}$$

- In Matrix-Vektor Schreibweise: $Ax = b$, Formales Lösen: $x = A^{-1}b$
- Definition von Matrix A in Matlab:
- `A = [3,4,-2; -1,2,8; 2,0,-5];` , Semikolon in eckiger Klammer: neue Zeile
- Definition von Spaltenvektor b in Matlab:
- `b = [4; -1; 3];` , Semikolon in eckiger Klammer: neue Zeile

LGS Fortsetzung

- Invertieren der Matrix A mit `inv(A)`: `x = inv(A)*b`
- Das Invertieren einer Matrix ist “numerisch teurer” als das Lösen eines einzelnen Gleichungssystems
- Hierfür hat Matlab den Backslash: `x = A \ b`: bedeutet “löse $Ax = b$ nach x”

Transponieren, Zeilen- und Spaltenvektoren

- Transponieren = Vertauschen von Zeilen und Spalten
- Wird in Matlab mit einem Apostroph hinter der Variable durchgeführt
- Bsp.: Definiere b als Zeilenvektor und wandle es in Spaltenvektor um:
- `b = [4, -1, 3]; b = b'`, (Ein Gleichheitszeichen ist eine Zuweisung)
- Achtung! Das Gleichungssystem Ax = b kann nur gelöst werden, wenn
  - A eine quadratische, invertierbare Matrix ist (die Determinante ist nicht null oder A hat vollen Rang)
  - b, ein Spaltenvektor, hat die selbe Zeilenanzahl wie Matrix A
  - 90% der Fehler in Anwendungen der linearen Algebra (Vektor- und Matrixrechnungen) sind falsche Dimensionierungen, meistens wegen fehlender Transponierung

Zugriff auf Elemente einer Matrix

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

- Der Zugriff auf einzelne Elemente wird durch runde Klammern angegeben: `A(3,2)` gibt mir den Inhalt des Elements in der **dritten Zeile** und **zweiten Spalte** von Matrix *A*.
- Ein Doppelpunkt als Index heißt "alle" (s. `doc colon`):
  - `A(2,:)` gibt alle Werte der **2. Zeile** aus (das ist ein Zeilenvektor)
  - `A(:,1)` gibt alle Werte der **1. Spalte** aus (das ist ein Spaltenvektor)
- Man kann auch Werte eines bestimmten Elementes einer Matrix zuweisen: `A(3,2) = 7` weist dem Element in der dritten Zeile und zweiten Spalte den Wert 7 zu.

Fortsetzung

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- Eine absolute leere Matrix (keine Zeilen, keine Spalten) kann durch leere eckige Klammern erzeugt werden:
  - `B = [];` erzeugt eine leere Matrix
  - `A(:,1) = [ ]` bewirkt, dass die erste Spalte von Matrix A gelöscht wird
- `size(A)` gibt die Dimension der Matrix wieder. Das Ergebnis ist ein Vektor `[Zeilenzahl, Spaltenzahl]`

Erzeugung spezieller Matrizen

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

- `zeros(2)`: ergibt eine 2×2 Nullmatrix
- `zeros(2,4)`: ergibt eine 2×4 Nullmatrix
- `eye(3)`: ergibt eine 3×3 Einheitsmatrix (Einsen auf der Hauptdiagonalen)
- `ones(3)`: ergibt eine 3×3 Matrix aus lauter Einsen
- `rand(2,3)`: ergibt eine 2x3 Matrix mit Zufallszahlen (gleichverteilt zwischen 0 und 1)
- ...

Übung 3-1

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Geben Sie folgende Matrix ein:

$$A = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 0 & 1 \\ 2 & 4 & 6 \end{pmatrix}$$

Was ist das Ergebnis der folgenden Anweisungen? (Überlegen Sie, bevor Sie mit Matlab rechnen.)

```
B = A' ↵
A(1,:) ↵
A(:,3) ↵
A(2,2) ↵
B(2,2) ↵
B(3,2) ↵
size(A), size(B) ↵
```

Übung 3-2

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Ersetzen Sie in einer 3×4 Matrix B mit Zufallszahlen das erste Element in der ersten Zeile durch 0 und streichen Sie die gesamte zweite Spalte von B.

Übung 3-3

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Erzeugen Sie eine 2×5 Matrix A mit Zufallszahlen. Bilden Sie daraus:

1. eine Matrix B, die aus den ersten drei Spalten von A besteht
2. eine Matrix C, die aus der zweiten und vierten Spalte von A besteht

Finden Sie die dazu nötigen Anweisungen mit `help colon` und `help paren` selbst heraus.

Matrixoperationen

- `A+B, A-B, A*B` bezeichnen die Matrizenaddition, -subtraktion und -multiplikation.
- Achtung! `A*B` ist wirklich eine Matrixmultiplikation, die elementweise Multiplikation ist `A.*B`
- Vergleiche:
  - `[1 2; -2 5] .* [3 6; 0 -1]`
  - `[1 2; -2 5] * [3 6; 0 -1]`
- Für quadratische Matrizen:

Fortsetzung

- Für **quadratische** Matrizen:
- `det(A)` ist die Determinante der Matrix A
- `eig(A)` berechnet die Eigenwerte
- `inv(A)` ist die Inverse der Matrix
- Ein kleiner Test der Operationen:
  - `A = rand(3)`
  - `det(A)`
  - `e = eig(A), prod(e)`
  - `iA = inv(A)`
  - `1/det(iA)`

Übung 3-4

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Sei `A = eye(3,3)`, `B = [1 2 3; 4 5 6; 7 8 9]`. Was ist das Ergebnis von:

`A+B` ↵	`B.^2` ↵
`A*2` ↵	`A.*B` ↵
`A/2` ↵	`A*B` ↵

Überlegen Sie, bevor Sie mit Matlab rechnen!

Übung 3-5

Lösen Sie das Gleichungssystem:

$$
\begin{aligned}
x + 4y + 6z &= 5 \\
3x - z &= 7 \\
2x + 7y - 3z &= 1
\end{aligned}
$$

- Aufgabe: Visualisierung einer Funktion `z = f(x,y)`, die von zwei Koordinaten abhängt
- Beispiel: `z = x*y`
- Herangehensweise:
    1. Erzeuge ein regelmäßiges Gitter von x- und y-Werten (Vektoren `X` und `Y`)
    2. Führe elementweise Berechnung von `z` durch (erzeuge eine Matrix `Z` mit den z-Werten zu jedem x,y-Paar)
    3. Plotte die Daten mit speziellen Befehlen

Fortsetzung

In Matlab:

1. Regelmäßiges Gitter:
    - Erzeuge Vektoren der regelmäßigen x- und y-Werte:
      `xvec = [-2:1:2]; yvec = [-1:1:1];`
    - Erzeuge hieraus ein regelmäßiges Gitter mit `meshgrid`:
      `[X,Y] = meshgrid(xvec,yvec);`
2. Jetzt *elementweise* Berechnung: `Z = X.*Y;`

Visualisierungsmöglichkeiten

- 3D-Visualisierung als Gitternetz: `mesh`

```
xvec = [-2:0.1:2]; yvec = [-1:0.1:1];
[X,Y] = meshgrid(xvec,yvec);
Z = X.*Y;
mesh(X,Y,Z); % s. auch "doc mesh"
xlabel('x'); ylabel('y'); zlabel('z');
```

- 3D-Visualisierung als Oberfläche: `surf`

```
xvec = [-2:0.1:2]; yvec = [-1:0.1:1];
[X,Y] = meshgrid(xvec,yvec);
Z = X.*Y;
surf(X,Y,Z); % s. auch "doc surf"
xlabel('x'); ylabel('y'); zlabel('z');
```

- Testen Sie, was `shading interp` bewirkt

Fortsetzung

2D-Visualisierung mit Isolinien und -flächen: `contour` und `pcolor`

- Erzeugen einer Karte mit 50 Höhenlinien: z.B.:
  `contour(X,Y,Z,50);`
- Erzeugen einer Karte mit gefüllten Flächen, die Höhenwerte darstellen:
  `pcolor(X,Y,Z); shading interp`
- Eine Farblegende gibt es mit `colorbar`

Übung 3-6

Stellen Sie die lineare Funktion $z = -0.4x - 0.8y + 3$ für $0 \leq x \leq 5$ und $0 \leq y \leq 5$ graphisch dar.

Um welche Art von Fläche handelt es sich?

Übung 3-7

Stellen Sie die Funktion $z = x^2 + y^2$ für $-3 \leq x \leq 3$ und $-3 \leq y \leq 3$ graphisch dar.

Verwenden Sie `subplot` um mehrere Darstellungsmöglichkeiten in einer Grafik zu vergleichen

Ausblick

- Wie kann ich häufig vorkommende Berechnungen/Abläufe automatisieren?