Mathematik am Computer

MATLAB, Teil III

M. Grote, H. Harbrecht und M. Schmidlin

Universität Basel

26.10 –29.10.2026

Übersicht

1 Matrizen

2 Matrizen in MATLAB

Definition von Matrizen

Eine **Matrix** ist eine rechteckige Anordnung von Zahlen, die in Zeilen und Spalten organisiert sind.
Matrizen können der Darstellung linearer Abbildungen zwischen Vektorräumen dienen.

Eine lineare Abbbildung $\Phi : \mathbb{R}^n \to \mathbb{R}^m$ ist eindeutig bestimmt durch die Bilder

$$\mathbf{c}_1 = \Phi(\mathbf{e}_1), \ldots, \mathbf{c}_n = \Phi(\mathbf{e}_n),$$

wobei $\mathbf{e}_1 = (1, 0, 0, \ldots, 0)^\intercal, \mathbf{e}_2 = (0, 1, 0, \ldots, 0)^\intercal, \ldots, \mathbf{e}_n = (0, 0, 0, \ldots, 1)^\intercal$ die Einheitsvektoren des $\mathbb{R}^n$ sind.

Schreiben wir $\mathbf{c}_1, \ldots, \mathbf{c}_n$ in ein rechteckiges Schema

$$A := (\mathbf{c}_1 | \ldots | \mathbf{c}_n),$$

so erhalten wir eine **Matrix** A mit mn Elementen in m Zeilen und n Spalten.

Definition von Matrizen

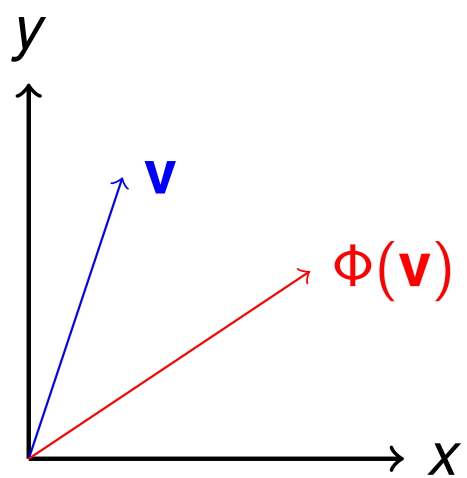
$\mathbb{R}^{m\times n}$ ist die Menge der **reellen Matrizen** mit m Zeilen und n Spalten.

Jede Matrix $\mathbf{A} = (\mathbf{a}_1|\ldots|\mathbf{a}_n) \in \mathbb{R}^{m\times n}$ definiert eindeutig eine lineare Abbildung $\Phi : \mathbb{R}^n \to \mathbb{R}^m$ durch $\mathbf{e}_i \mapsto \mathbf{a}_i$, mit $i = 1, \ldots, n$.
Beachte: $\mathbf{e}_i$ sind die Einheitsvektoren in $\mathbb{R}^n$ und $\mathbf{a}_i \in \mathbb{R}^m$.

Beispiel: Matrix als Werkzeug zur Transformation von Vektoren:

$$\begin{pmatrix} 0 & 1 \\ 2 & 0 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0x + 1y \\ 2x + 0y \end{pmatrix} = \begin{pmatrix} y \\ 2x \end{pmatrix}$$

Diese Matrix tauscht die Koordinaten des Vektors aus und streckt die neue x-Koordinate um den Faktor 2.
In der Grafik transformiert die Matrix $\left(\begin{smallmatrix} 0 & 1 \\ 2 & 0 \end{smallmatrix}\right)$ den Vektor $v{:=}(1,3)^\intercal$ in $\Phi(\mathbf{v}){=}(3,2)^\intercal$.



Grundrechenarten für Matrizen

Die Grundrechenarten für Matrizen sind folgendermassen definiert :

Addition, Subtraktion, Multiplikation mit Skalaren elementweise: $\mathbf{A} = (a_{ij})_{i,j=1}^{m,n}$, $\mathbf{B} = (b_{ij})_{i,j=1}^{m,n} \in \mathbb{R}^{m \times n}$:

$$\mathbf{A} \pm \mathbf{B} = (a_{ij} \pm b_{ij}) \qquad \alpha \mathbf{A} = (\alpha a_{ij})$$

Definiere die Matrix-Vektor-Multiplikation so, dass für $\mathbf{A} \in \mathbb{R}^{m \times n}$ und $\mathbf{x} \in \mathbb{R}^n$ gilt: $\mathbf{y} = \mathbf{A}\mathbf{x} \in \mathbb{R}^m$:

$$y_i = \sum_{j=1}^{n} a_{ij} x_j,$$

wobei gilt $\mathbf{y} = (y_i)_{i=1}^{m}$, $\mathbf{x} = (x_i)_{i=1}^{n}$.

Grundrechenarten für Matrizen

Matrixmultiplikation : Für $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{l \times m}$, $\mathbf{B} = (b_{jk}) \in \mathbb{R}^{m \times n}$ ist

$$\mathbf{C} = \mathbf{AB} = (c_{ik}) \in \mathbb{R}^{l \times n} \text{ mit } c_{ik} = \sum_{j=1}^{m} a_{ij} b_{jk}.$$

Ist $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{m \times n}$ und $\mathbf{b} \in \mathbb{R}^m$ so ist

$$\mathbf{Ax} = \mathbf{b}$$

ein lineares Gleichungssystem für einen unbekannten Vektor $\mathbf{x} \in \mathbb{R}^n$ mit m Gleichungen und n Unbekannte.

Beispiele

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} - \begin{pmatrix} 3 & 2 & 1 \\ 0 & 10 & 0 \end{pmatrix} = \begin{pmatrix} -2 & 0 & 2 \\ 4 & -5 & 6 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 1 \cdot 1 + 2 \cdot 0 + 3 \cdot 2 \\ 4 \cdot 1 + 5 \cdot 0 + 6 \cdot 2 \end{pmatrix} = \begin{pmatrix} 7 \\ 16 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 2 & 0 \end{pmatrix} = \begin{pmatrix} 7 & 3 \\ 16 & 9 \end{pmatrix}$$

Beispiele

Das lineare Gleichungssystem

$$\begin{cases} x - y + z = 0 \\ y - z = -24 \\ x + 2z = 0 \end{cases}$$

ist äquivalent zu

$$\underbrace{\begin{pmatrix} 1 & -1 & 1 \\ 0 & 1 & -1 \\ 1 & 0 & 2 \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} x \\ y \\ z \end{pmatrix}}_{\mathbf{v}} = \underbrace{\begin{pmatrix} 0 \\ -24 \\ 0 \end{pmatrix}}_{\mathbf{b}}$$

Matrizen erstellen

Erzeugen von Matrizen:

```
[1 2 3 ; 4 5 6]      zeros(2,3)      eye(5)
[1,2,3 ; 4,5,6]
```

Erzeugen von speziellen Zeilenvektoren:

`1:3` ergibt den Vektor $\begin{pmatrix} 1 & 2 & 3 \end{pmatrix}$.
`1:0.2:2` ergibt $\begin{pmatrix} 1.0 & 1.2 & 1.4 & 1.6 & 1.8 & 2.0 \end{pmatrix}$.

Zuweisung (um Werte in einer Variablen abzuspeichern):

```
c1 = 6              A = [1 2 3 ; 4 5 6]      v = 1:10
x = [1;2;3]         y = [1 2 3]
```

Command Window

```
>> c1 = 6

c1 =

     6

>> A = [1 2 3 ; 4 5 6]

A =

     1     2     3
     4     5     6

>> v = 1:10

v =

     1     2     3     4     5     6     7     8     9    10

>> x = [1;2;3]

x =

     1
     2
     3

>> y = [1 2 3]

y =

     1     2     3

fx >>
```

Matrizen erstellen

Verkleben von Matrizen:

`[A, zeros(2,2)]` ergibt

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

`B = [A, zeros(2,2); eye(2), A]` speichert die Matrix

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

in der Variablen *B* ab.

Operationen auf Matrizen

Elementauswahl für Zugriff und Zuweisung:

`A(1,2) A(1:2,2) A(1:2,1:3) A(:,[1 3])`

Elementweise *arithmetische* Operationen: `+ - .* ./ .^ .'`

Matrix-Matrix Produkt: $\begin{pmatrix} a & b \\ c & d \end{pmatrix} * \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 1a+3b & 2a+4b \\ 1c+3d & 2c+4d \end{pmatrix}$

Matrix-Matrix Produkt Elementweise: $\begin{pmatrix} a & b \\ c & d \end{pmatrix} .* \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 1a & 2b \\ 3c & 4d \end{pmatrix}$

Elementweise Funktionen:

`abs sin cos exp sqrt min max ...`

Elementweise logische Operationen:

`== ~= < > <= >= & | ~`

Das Ergebnis ist elmentweise 0 (`false`) bzw. 1 (`true`).

Operationen auf Matrizen

Grundrechenoperationen der Linearen Algebra

`+` `-` `*` `^`

Weitere Operationen:

`A\B`	$\mathbf{A}^{-1} \cdot \mathbf{B}$
`A'`	$\mathbf{A}^{\top}$ (falls $\mathbf{A}$ reell)
`det(A)`	$\det(\mathbf{A})$
`inv(A)`	$\mathbf{A}^{-1}$
`rank(A)`	$\operatorname{rang}(\mathbf{A})$

Das Lösen von linearen Gleichungssystemen

Eingabe:

```
A = [ 1 2 ; 3 4 ]
```

$$\mathbf{A} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

```
b = [ 3 ; 5 ]
```

$$\mathbf{b} = \begin{pmatrix} 3 \\ 5 \end{pmatrix}$$

```
A\b
```

$$\mathbf{A}^{-1}\mathbf{b}$$

Ausgabe ist die Lösung von $\mathbf{Ax} = \mathbf{b}$:

```
ans =
 -1
   2
```

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} -1 \\ 2 \end{pmatrix} = \begin{pmatrix} 3 \\ 5 \end{pmatrix}$$

Speicherverwaltung

Wenn man auf einen Matrixeintrag zugreifen will, etwa durch `A(2,3:6)`, so muss die Matrix mindestens die Grösse 2×6 besitzen und entsprechend **vorher** definiert werden.

Wenn man auf `A(2,3:6)` zugreift, ohne die Matrix vorher zu definieren oder falls sie zu klein ist, so wird sie automatisch als Nullmatrix der notwendigen Grösse angelegt oder entsprechend vergrössert. Trotzdem sollte man die Matrix vorher selbst definieren weil:

- die Grössenanpassung sehr viel Laufzeit kostet
- Definition der Matrixgrösse am Anfang ist guter Programmierstil

Der Speicher nicht mehr benötigter Variablen wird durch

`clear` oder `clear`+Variablename

freigegeben.

Matrix-Vektor-Schreibweise

Es gibt (mindestens) zwei Möglichkeiten, den Vektor

$$v = \begin{pmatrix} 1 & -2 & 3 & -4 & \cdots & \cdots & 999 & -1000 \end{pmatrix}$$

zu erzeugen:

```
v = 1:1000;
for i = 1:500
    v(2*i) = -v(2*i);
end
```

```
v = 1:1000;
v(2:2:1000) = -v(2:2:1000);
```

Die zweite Möglichkeit verwendet statt Schleifen die vektorielle Rechnung. Das ist viel schneller!

Man sollte daher alle Matrix-Vektor-Operationen möglichst ohne for-Schleifen umsetzen.