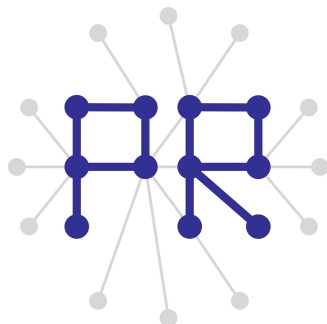
# Einführung in die Informatik I

## Kapitel I.11: Matrizen

Prof. Dr. Marcin Grzegorzek[1]

Research Group for Pattern Recognition
www.pr.informatik.uni-siegen.de

Institute for Vision and Graphics
University of Siegen, Germany



[1] Die im Rahmen dieser Lehrveranstaltung verwendeten Lernmaterialien wurden uns zum Großteil von Herrn Prof. Dr. Wolfgang Wiechert und Herrn Prof. Dr. Roland Reichardt zur Verfügung gestellt.

# Inhaltsverzeichnis

**I. MATLAB-Einführung**

1. Voraussetzungen und Konventionen
2. Variablen und arithmetische Ausdrücke
3. Automatisierungen von Berechnungen
4. Logische Ausdrücke
5. Verzweigungen
6. Schleifen
7. Fehlersuche in Programmen
8. Funktionen
9. Arbeitsweise von Funktionen
10. Vektoren
11. Matrizen

# Inhaltsverzeichnis II

**II. Algorithmen**

1. Suchen
2. Spezielle Suchalgorithmen
3. Sortieren
4. Rekursion und Quicksort

# 2-dimensionale Felder

- Ein 2-dimensionales Feld (auch: Matrix) ist eine tabellarische Anordnung von Zahlenwerten.
- Die Tabelle kann eine beliebige Zahl *n* von Zeilen und *m* von Spalten haben. Sprechweise: „*n* x *m* – Matrix“
- Beispiele:
  - rechteckige 3x4-Matrix

| | | | |
|---|---|---|---|
| 1 | -5 | 8 | -9 |
| -1 | 3 | 5 | 4 |
| -3 | 45 | 4 | -3 |

  - quadratische 3x3-Matrix

| | | |
|---|---|---|
| 1 | -5 | 8 |
| -1 | 3 | 5 |
| -3 | 45 | 4 |

  - einzeilige 1x3-Matrix

| | | |
|---|---|---|
| -1 | 3 | 5 |

  - einspaltige 3x1-Matrix

| |
|---|
| -5 |
| 3 |
| 45 |

- Vektoren sind identisch mit 1xn- bzw. nx1-Matrizen. Einzelwerte (Skalare) sind identisch mit 1x1-Matrizen.
- MATLAB speichert (fast) alle Daten als Matrizen. MATLAB = MATrix LABoratory

# Vorkommen von Matrizen

Wertetabellen

| x | $x^2$ | $x^3$ |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 1 | 1 |
| 2 | 4 | 8 |
| 3 | 9 | 27 |
| 4 | 16 | 64 |

Ohne die Überschriften

Tabellenkalkulation

| | Stud. | Sem | Pkte | Note |
|---|---|---|---|---|
| Meier | MB | 1 | 65 | 2.0 |
| Müller | WI | 3 | 73 | 1.3 |
| Schmitz | MB | 3 | 23 | 4.7 |
| Schulte | WI | 1 | 45 | 3.0 |

Nur Zahlenfelder bilden eine Matrix

# Vorkommen von Matrizen

Entfernungstabellen

| | Frankfurt | Köln | München | Stuttgart |
|---|---|---|---|---|
| Frankfurt | 0 | 200 | 350 | 300 |
| Köln | 200 | 0 | 550 | 450 |
| München | 350 | 550 | 0 | 200 |
| Stuttgart | 300 | 450 | 200 | 0 |

Diagonale mit Nullen füllen

Spielergebnisse

| | **A** | **B** | **C** | **D** |
|---|---|---|---|---|
| **A** | 0 | 2 | 3 | 4 |
| **B** | 1 | 0 | 2 | 3 |
| **C** | 2 | 0 | 0 | 2 |
| **D** | 0 | 4 | 0 | 0 |

A gegen B spielt 2:1

# Vorkommen von Matrizen

Digitale Bilder (Bitmaps)

4 MegaPixel
= 2000 x 2000 Pixel

Rot-Matrix (Ausschnitt)

| | | | | |
|---|---|---|---|---|
| 0.12 | 0.15 | 0.16 | 0.56 | 0.64 |
| 0.13 | 0.56 | 0.57 | 0.62 | 0.62 |
| 0.14 | 0.56 | 0.58 | 0.63 | 0.64 |

Grün-Matrix (Ausschnitt)

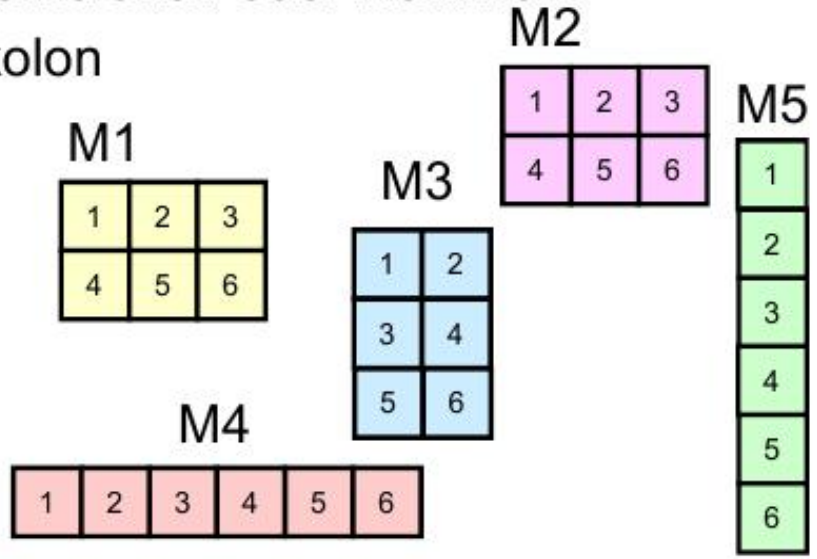
| | | | | |
|---|---|---|---|---|
| 0.12 | 0.15 | 0.16 | 0.56 | 0.64 |
| 0.13 | 0.56 | 0.57 | 0.62 | 0.62 |
| 0.14 | 0.56 | 0.58 | 0.63 | 0.64 |

Blau-Matrix (Ausschnitt)

| | | | | |
|---|---|---|---|---|
| 0.12 | 0.15 | 0.16 | 0.56 | 0.64 |
| 0.13 | 0.56 | 0.57 | 0.62 | 0.62 |
| 0.14 | 0.56 | 0.58 | 0.63 | 0.64 |

# Erzeugung von Matrizen in MATLAB

- Direkte Matrizendefinition:
  - Matrix in eckigen Klammern
  - Werte getrennt durch Leerzeichen oder Komma
  - Zeilenwechsel mit Semikolon
- Beispiele:
  - **M1=[1 2 3; 4 5 6]**
  - **M2=[1,2,3; 4,5,6]**
  - **M3=[1 2; 3 4; 5 6]**
  - **M4=[1 2 3 4 5 6]**
  - **M5=[1;2;3;4;5;6]**




- Initialisieren und Füllen mit Nullen:
  - **M=zeros(*Zeilenzahl*,*Spaltenzahl*)**

# Zeilenfortsetzung

- Matrizen können sehr groß werden
  (viele Zeilen und Spalten)

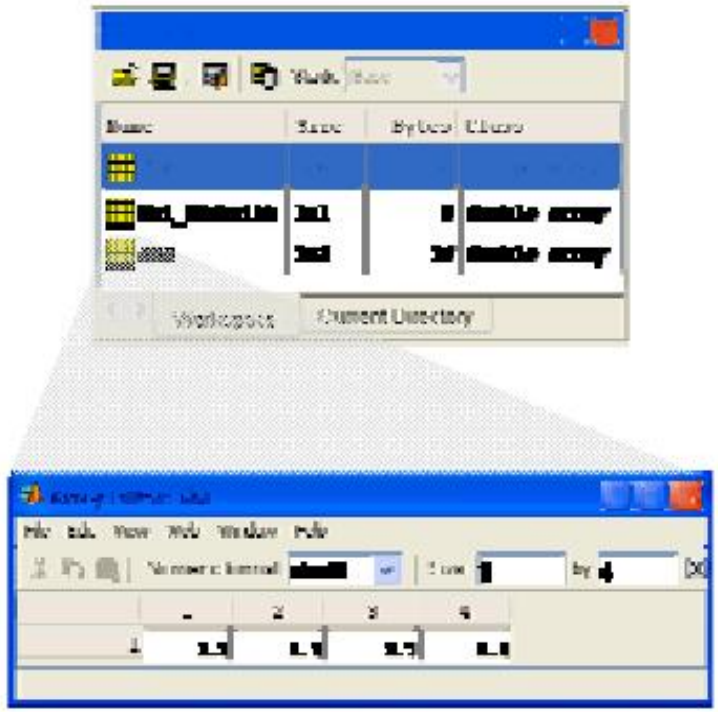
- Übersichtliche Matrizen-Notation in MATLAB-Skripten
  mit dem Zeilenfortsetzungssymbol . . .

```
M=[1 2 3;...
   4 5 6;...
   7 8 9]
```

```
% Drehstreckmatrix
P=[r*sin(phi), -r*cos(phi);...
   r*cos(phi),  r*sin(phi)]
```

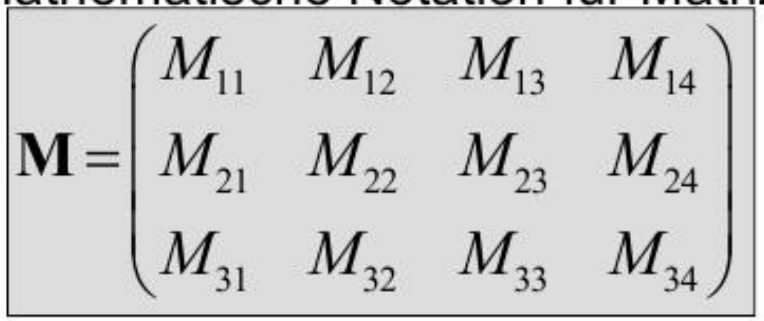
# Editieren von Matrizen

- Im *Workspace* werden die Variablen mit der Matrixdimension (***Size***) und dem Speicherbedarf (***Bytes***) angezeigt.
- Durch einen Doppelklick auf einen Variablennamen öffnet sich ein neues Fenster mit dem *Array Editor*.



# Zugriff auf einzelne Matrizeneinträge

- Mathematische Notation für Matrizen und Matrizeinträge:



$$\mathbf{M} = \begin{pmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & M_{22} & M_{23} & M_{24} \\ M_{31} & M_{32} & M_{33} & M_{34} \end{pmatrix}$$


- In MATLAB erfolgt der Zugriff auf einen Matrixeintrag mit dem Klammeroperator:
  - Notation: `M(1,1),M(1,2),...,M(2,1),M(2,2),...`
  - Die Indices müssen im Bereich 1,…,n bzw. 1,…,m liegen.
- Die Abmessungen einer Matrix können abgefragt werden mit:
  - `size(M,1)` Zeilenzahl
  - `size(M,2)` Spaltenzahl
  - `size(M )` Zeilenzahl und Spaltenzahl als 1x2-Vektor

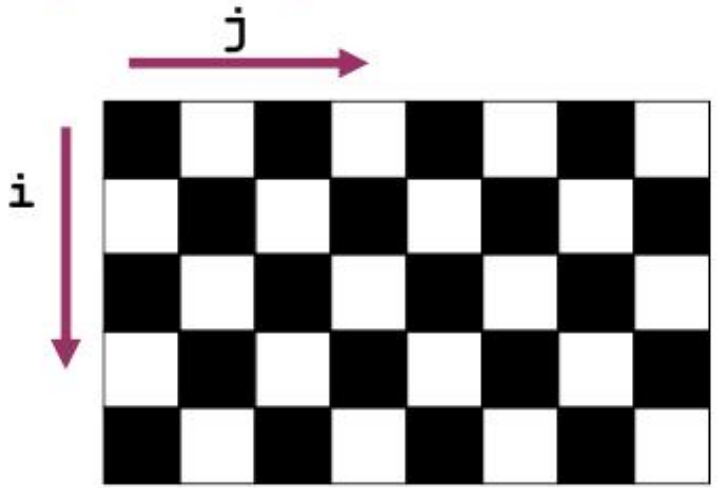
# Beispiele für den Zugriff

- Matrix initialisieren:
  - `M=zeros(2,3)` → 0 0 0 / 0 0 0
- Schreibender Zugriff:
  - `M(2,3)=1` → 0 2 0 / 0 0 1
  - `M(1,2)=2`
  - `M(0,3)=5` → Fehlermeldung
  - `M(4,5)=2` → Undefiniertes Ergebnis
- Lesender Zugriff:
  - `x=M(2,2)` → 0
  - `y=M(1,2)` → 2
  - `z=M(3,3)` → Fehlermeldung
- Längenabfrage:
  - `s=size(M)` → 2 3
  - `s(2)` → 3
  - `size(M,2)` → 3

# Schachbrettmuster

- Quadratisches 100x100-Schachbrett:

```
M=zeros(100,100); % Initialisieren
for i=1:2:100     % ungerade Zeilen
   for j=1:2:100  % ungerade Spalten
      M(i,j)=1;
   end;
end;
for i=2:2:100     % gerade Zeilen
   for j=2:2:100  % gerade Spalten
      M(i,j)=1;
   end;
end;
% besetzte
% Zellen
% zeigen
spy(M);
```



# Beispiel: Bearbeitung eines S/W-Bildes

- Es wird ein S/W-Bild als Matrix mit 128 Graustufen von der Platte eingelesen und dargestellt:

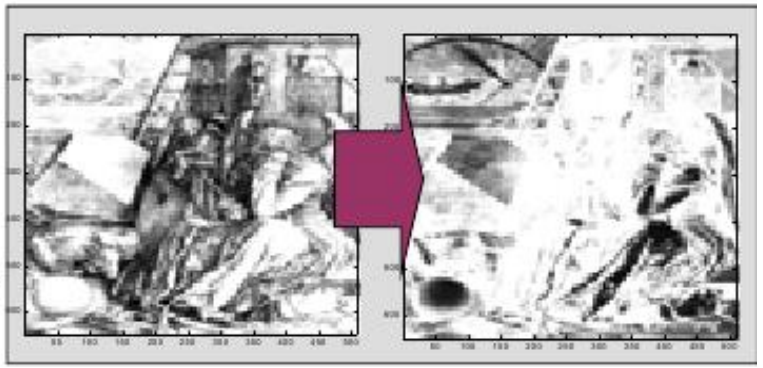
```
load('durer','X');          % Laden der Matrix X
colormap('gray');           % Farbzuordnungstabelle
image(X);                   % Bild darstellen
```

- Das Bild wird 20% dunkler gemacht:

```
for i=1:size(X,1)           % Geschachtelte
    for j=1:size(X,2)       % for-Schleifen
        X(i,j)=round(0.8*X(i,j));
    end;
end;
```

- Das Bild wird invertiert:

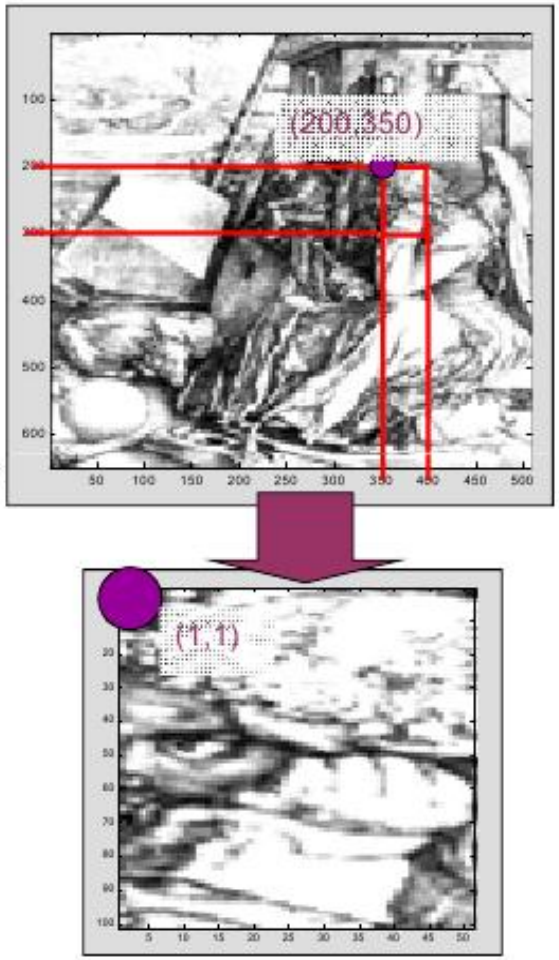
```
for i=1:size(X,1)
    for j=1:size(X,2)
        X(i,j)=128-X(i,j);
    end;
end;
```



# Beispiel: Bildausschnitt

- Der Ausschnitt
  Zeile 200/300
  Spalte 350/400
  soll auskopiert werden:

```
Y=zeros(101,51);
for i=200:300
    for j=350:400
        Y(i-199,j-349)=X(i,j);
    end;
end;
image(Y);
```



# Plot-Befehl mit einer Matrix

- Wertetabelle gegeben (stehend oder liegend):

| x | $y_1$ | $y_2$ | $y_n$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| 2 | 4 | 8 | … |
| 3 | 9 | 27 | … |
| … | … | … | … |

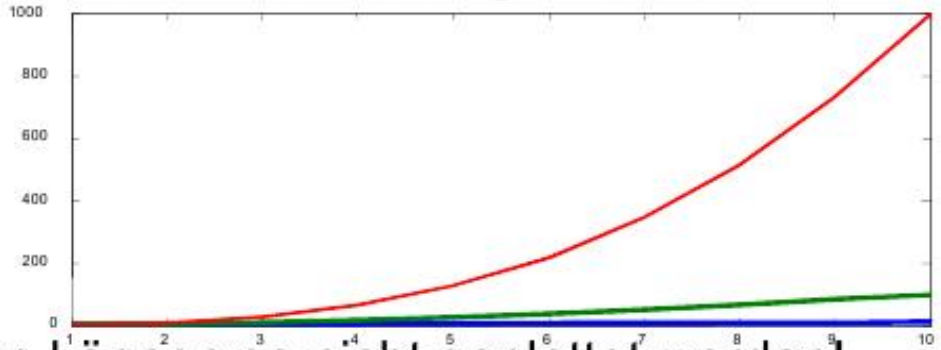
**x** **Y**

**x**

| x | 0 | 1 | 2 | … |
|---|---|---|---|---|
| $y_1$ | 0 | 1 | 4 | … |
| $y_2$ | 0 | 1 | 8 | … |
| … | | | | |
| $y_n$ | 0 | 1 | … | … |

**Y**

- Plot jeder Spalte von Y : `plot(Y);`




Zeilen können so nicht geplottet werden!

# Beispiel: Übersetzungen einer Gangschaltung

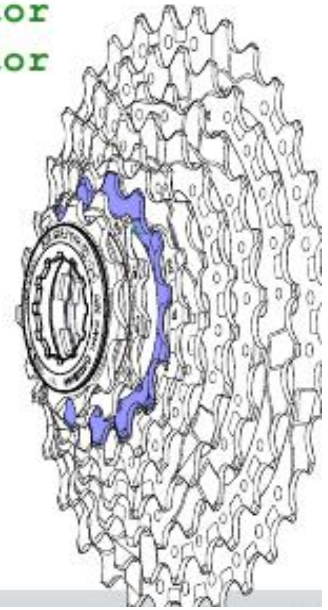
- Über-setzungs-tabelle:

| | Hinten | | | | | |
|---|---|---|---|---|---|---|
| **Vorne** **V/H** | 10 | 12 | 14 | 16 | 18 | 21 |
| 44 | 4.40 | 3.67 | 3.14 | 2.75 | 2.44 | 2.10 |
| 32 | 3.20 | 2.67 | 2.29 | 2.00 | 1.78 | 1.52 |
| 22 | 2.20 | 1.83 | 1.57 | 1.38 | 1.22 | 1.05 |

- 
```
function UT = UebersetzungsTabelle(Vorne,Hinten);
% Vorne : Anzahl Zähne am vorderen Ritzel als Vektor
% Hinten: Anzahl Zähne am hunteren Ritzel als Vektor

% Initialisierung der Tabelle als Matrix
% Achtung: Zeilen und Spalten vertauscht !
UT = zeros(length(Hinten),length(Vorne));

% Zwei Schleifen für alle Ritzel vorne und hinten
for v=1:length(Vorne)
    for h=1:length(Hinten)
        UT(h,v)=Vorne(v)/Hinten(h);
    end
end
```



# Plot der Übersetzungen

- Aufrufbeispiel:

```
Hinten = [10,12,14,16,18,21,24,28,32];
Vorne  = [44,32,22];
UT = UebersetzungsTabelle(Vorne,Hinten);
```

Matrix wird spaltenweise dargestellt

- 

```
% Diagramm der Übersetzungen
plot(UT,'-x');
xlabel('Nummer des Gangs');
ylabel('Übersetzung');
```