

---

# Grundlagen der Bildverarbeitung

## *Übung 0 - Einführung in MATLAB*

---

Gurbandurdy Dovletov, M.Sc.

Raum: BC 414

Tel.: 0203-379-3583

Email: [gurbandurdy.dovletov@uni-due.de](mailto:gurbandurdy.dovletov@uni-due.de)

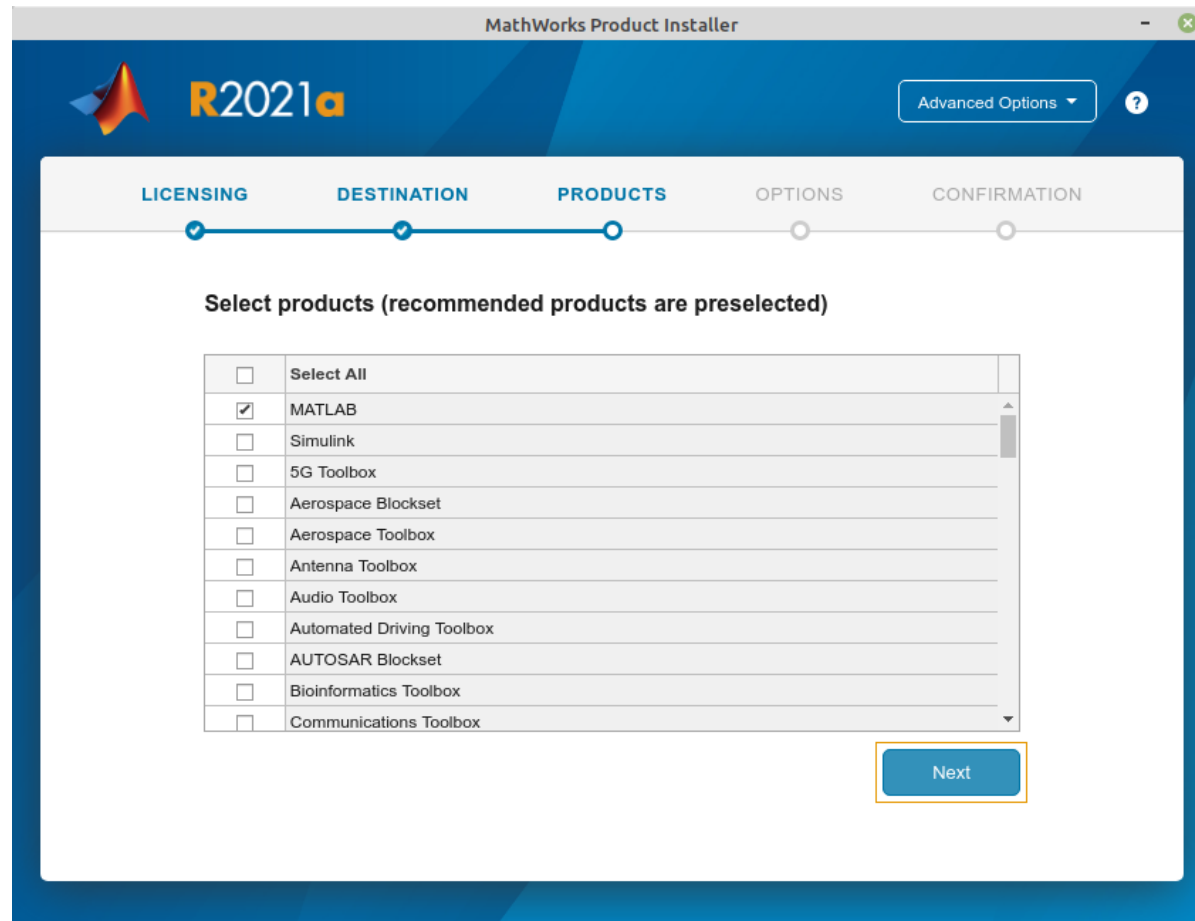
7. April 2022

# Quiz

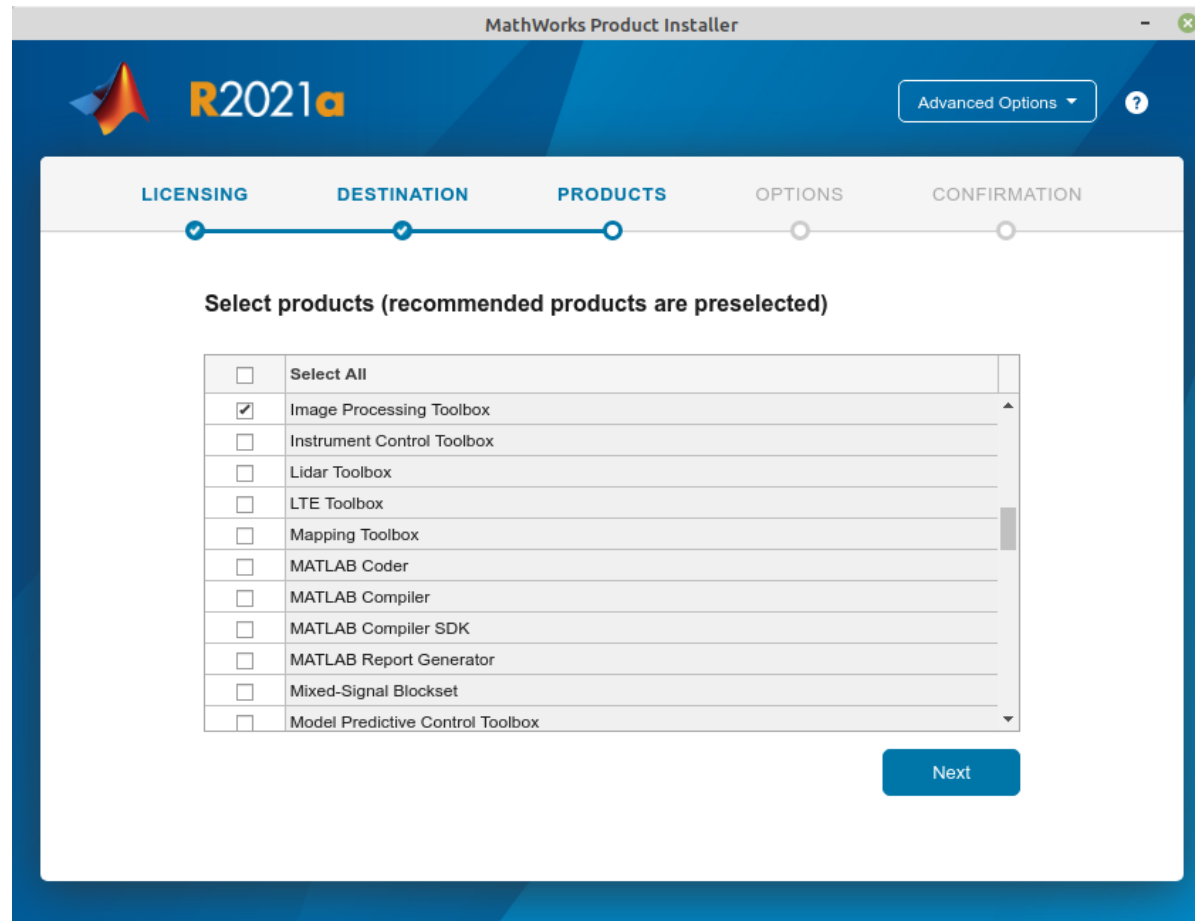
---

- Welche Verarbeitungsebenen haben Sie in der Vorlesung kennen gelernt?

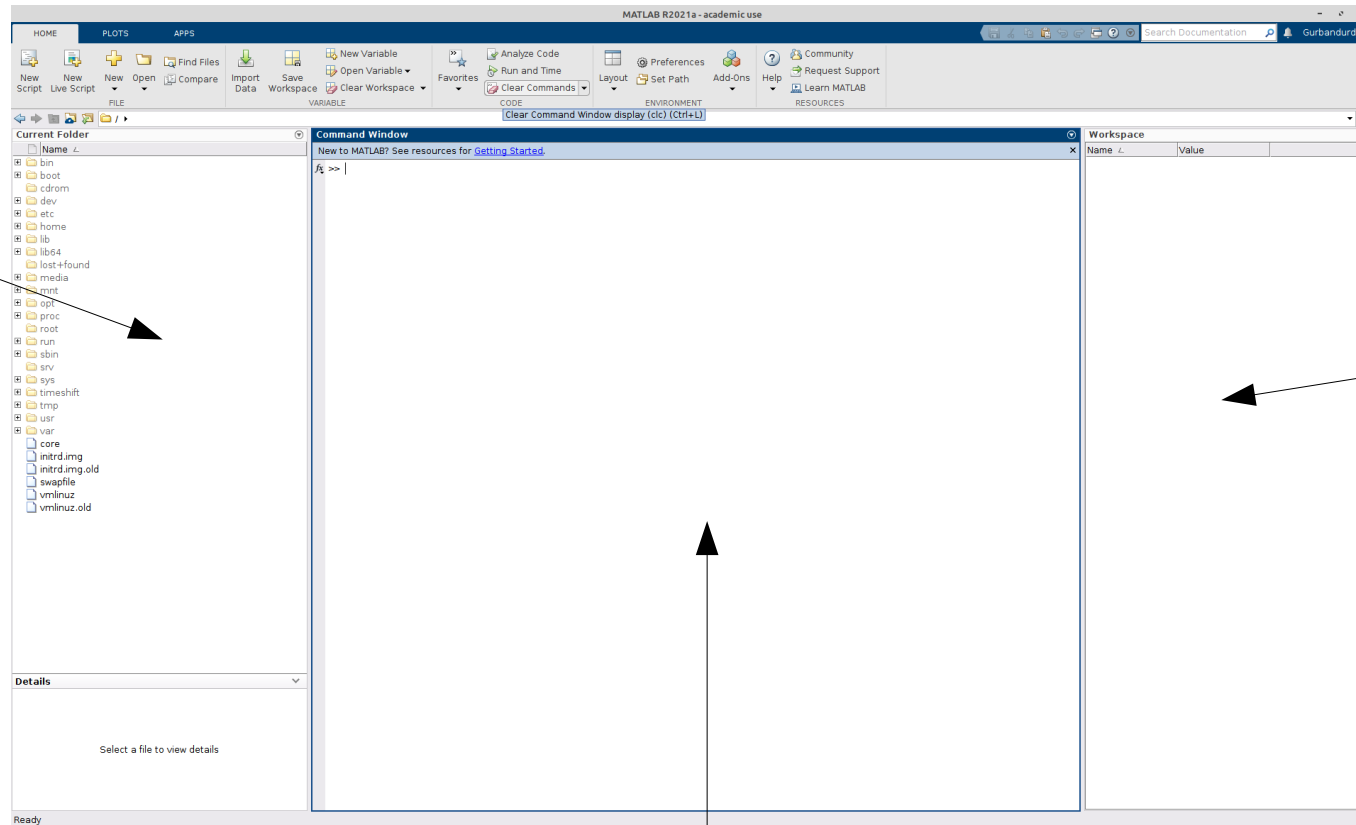
# Matlab Installation (Produkte)



# Matlab Installation (Produkte)



# Die Entwicklungsumgebung



Zugriff auf  
Dateien

Workspace:  
Anzeige von  
bekannten  
Variablen/Daten

Konsole:  
Eingabe von Befehlen

# Was ist Matlab?

---

- **Matrix** laboratory
- matrixbasierte High-Level Sprache mit Entwicklungsumgebung
- Einfachste Datenstruktur: Array

# Matrizen und Arrays

---

- Grundlegende Datenstruktur: Array
- Skalar: 0-dim Array
- Vektor: 1-dim Array
- Matrix: 2-dim Array

# Erzeugung von Arrays

---

- Verwendung von eckigen Klammern
- Array wird zeilenweise definiert
- Spalten werden durch Leerzeichen oder Komma getrennt
- Zeilen werden durch Semikolon getrennt
- Beispiele:
  - $A = [1 \ 2 \ 3; \ 4 \ 5 \ 6; \ 6 \ 7 \ 8];$
  - $sv = [1;2;3];$
  - $zv = [1 \ 2 \ 3];$ 
    - %oder  $zv = [1,2,3];$



# Erzeugung einer Null-Matrix

---

- `N = zeros(5,1);`
- Erster Parameter: Anzahl der Zeilen
- Zweiter Parameter: Anzahl der Spalten
- Funktioniert auch bei n-dim Arrays

# Erzeugung einer zufälligen Matrix

---

- `R = rand(5,5);`
- Erster Parameter: Anzahl der Zeilen
- Zweiter Parameter: Anzahl der Spalten
- Werte liegen zwischen 0 und 1
  
- Funktioniert auch bei n-dim Arrays

# Erzeugung eines „Laufvektors“

---

- $v1 = 1:10;$ 
  - Zeilenvektor, der die Zahlen 1,2,3,...,10 beinhaltet
- $v2 = 1:2:20;$ 
  - Zeilenvektor, der die Zahlen 1,3,5,...,19 beinhaltet
  - Die erste Zahl definiert den Startwert
  - Die mittlere Zahl definiert die Schrittweite
  - Die letzte Zahl definiert die Schranke
- $v3 = 1:-2:-20;$

# Indizierung

- Indizierung beginnt mit 1 (nicht mit 0 wie in C++)
- Matrizen können mit Vektoren indiziert werden

```
%Zufallsmatrix 100x100 (Werte zwischen 0 und 1)
```

```
A = rand(100,100);
```

```
%Wert in 10. Zeile und 20. Spalte
```

```
x = A(10,20);
```

```
%Teilmatrix von Zeile 10-20 und Spalte 30-40
```

```
T1 = A(10:20, 30:40);
```

```
%Teilmatrix von Zeile 40-100 und Spalte 1-100
```

```
T2 = A(40:end, : );
```

```
%Teilmatrix von Zeile 33-67 und Spalte 1-100
```

```
T2 = A(round(end/3):round(end/3*2), : );
```

# Einige praktische Operatoren

---

- $A+10$  ist komponentenweise definiert
- $A*B$  ist die übliche Matrixmultiplikation
- $A.*B$  ist eine komponentenweise Multiplikation (Hadamard-Produkt)
- $A.^3$  potenziert ebenfalls komponentenweise
- $A'$  berechnet die Transponierte von  $A$
- $\text{inv}(A)$  berechnet das Inverse von  $A$
- $\text{size}(A)$  gibt die Größe der Matrix wieder

# Änderung der Matrixform

---

- $C = [A, B]$       hängt B rechts von A dran
- $C = [A; B]$       hängt B unter A dran
- $B = \text{repmat}(A, [4, 5]);$       wiederholt A in B  
4 x 5 mal
- $A = [1 \ 2 \ 3; 4 \ 5 \ 6];$   
 $B = \text{reshape}(A, [3, 2]);$       ordnet die 2x3 Matrix  
als 3x2 Matrix an

# Funktionen und Skripte

---

- Matlab verwendet .m-Dateien
- Sofern nicht anders vorgegeben, sind .m-Dateien zunächst Skripte
- `function [output] = fun_name(input)`  
definiert eine Funktion
- Wird eine .m-Datei gestartet hat sie nur Zugriff auf Funktionen, die im „Current Folder“ abgelegt sind
- Auf Funktionen in anderen Ordnern kann durch `addpath('Pfad')` sowohl relativ als auch absolut verwiesen werden
- Sollen auch Unterordner berücksichtigt werden:  
`addpath(genpath('Pfad'))`

# Schleifen, Bedingungen, etc...

---

- ```
for i = 1:10:100
    disp(['Iteration: ' num2str(i)]);
end
```
- ```
i = 1;
while (i <= 100)
    disp(['Iteration: ' num2str(i)]);
    i = i + 1;
    %i++ gibt es leider nicht
end
```
- ```
if ((a >= c) && (c <=b)) || d
    e = 10;
else
    e = 11;
end
```
- ```
A = rand(100,100);
ind = A < 0.5;
A(ind) = 0;
```



# Aufgabe 0.1

---

- Erstellen Sie eine Nullmatrix A mit 8x10 Elementen.
- Erstellen Sie eine Matrix B der Größe 4x4 mit zufällig gewählten Einträgen.
- Setzen Sie die Elemente von B in die Mitte von A ein und speichern Sie diese neue Matrix als C.
- Transponieren Sie C.
- Geben Sie die 3. Zeile von C aus.
- Ändern Sie die Struktur von C auf die Größe 20x4 und speichern Sie die geänderte Matrix als C\_reshape ab.
- Vergleichen Sie beide Matrizen. Wie funktioniert die Strukturänderung?

# Aufgabe 0.2

---

- Erstellen Sie einen Vektor  $V$  mit 100 beliebigen Einträgen.
- Setzen Sie jeden 2. Wert von  $V$  auf 0.
- Löschen Sie jeden 2. Wert von  $V$ .

# Aufgabe 0.3

---

- Erstellen Sie ein Array M mit 1000 x 4 beliebigen Einträgen.
  - Jede Zeile des Arrays soll eine 2x2 Matrix der Form  $\begin{bmatrix} a_{11} & a_{12} & a_{21} & a_{22} \end{bmatrix}$  repräsentieren.
  - Damit ist M ein Vektor der Länge 1000, der als Einträge 2x2 Matrizen hat.
- Berechnen Sie für jede Matrix die Determinante ohne dabei Schleifen zu verwenden.