

# Einführung in die Informatik I

Kapitel I.3: Automatisierungen von Berechnungen

Prof. Dr.-Ing. Marcin Grzegorzek

Juniorprofessur für Mustererkennung im Institut für Bildinformatik

Department Elektrotechnik und Informatik

Fakultät IV der Universität Siegen

31.10.2012

# Inhaltsverzeichnis

- I. MATLAB-Einführung
  - 1. Voraussetzungen und Konventionen
  - 2. Variablen und arithmetische Ausdrücke
  - 3. Automatisierung von Berechnungen**
  - 4. Logische Ausdrücke
  - 5. Verzweigungen
  - 6. Schleifen
  - 7. Fehlersuche in Programmen
  - 8. Funktionen
  - 9. Arbeitsweise von Funktionen
  - 10. Vektoren
  - 11. Matrizen
- II. Algorithmen
- III. MATLAB-Fortsetzung
- IV. Wissenschaftliche Werkzeuge

# Wiederholen von Befehlen

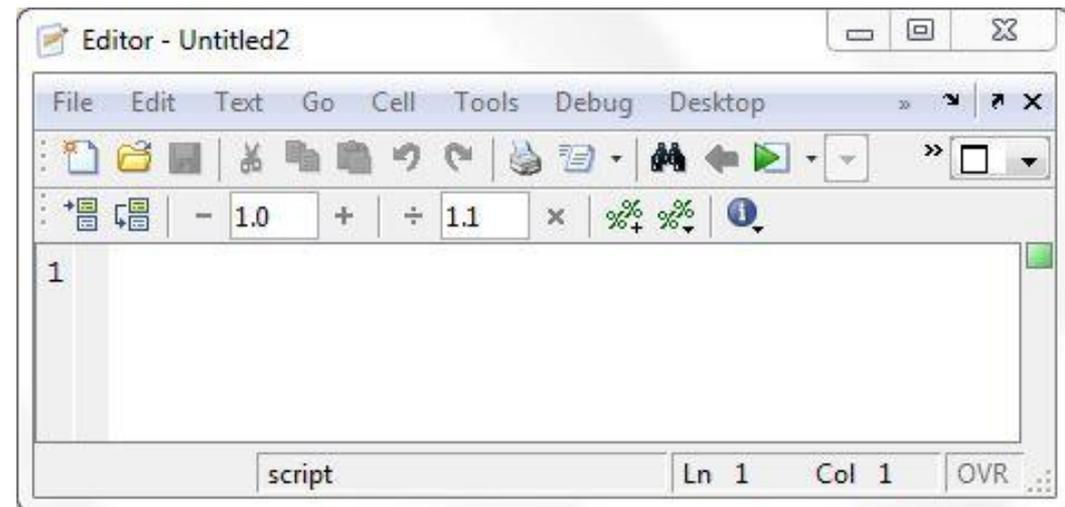
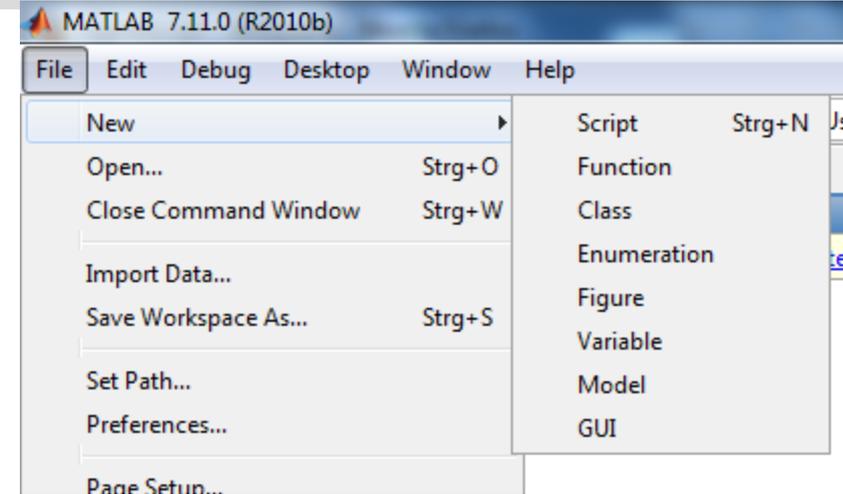
- MATLAB stellt drei verschiedene Verfahren für das wiederholte Ausführen von Befehlen zur Verfügung
  - Im Command Window mit den Pfeiltasten die zuletzt verwendeten Befehle aufrufen.
  - In der Command History die zuletzt verwendeten Befehle auswählen.
  - Automatische Abarbeitung mehrerer Befehle durch Skripte.

# Skripte in MATLAB (1)

- MATLAB wird primär nicht über das Command Window bedient, weil das auf Dauer zu umständlich ist.
- Die einzelnen Befehle können in einer Datei gespeichert und anschließend auf einmal ausgeführt werden.
- Diese Dateien müssen die Dateierweiterung „.m“ haben (genannt M-File).
- Durch den Dateinamen wird der Befehlssatz von MATLAB erweitert, d.h. Dateiname = Befehl, das Ausführen eines Skriptes kann also mit Eingabe des Dateinamens dieses Skriptes (ohne „.m“) im Command Window veranlasst werden.

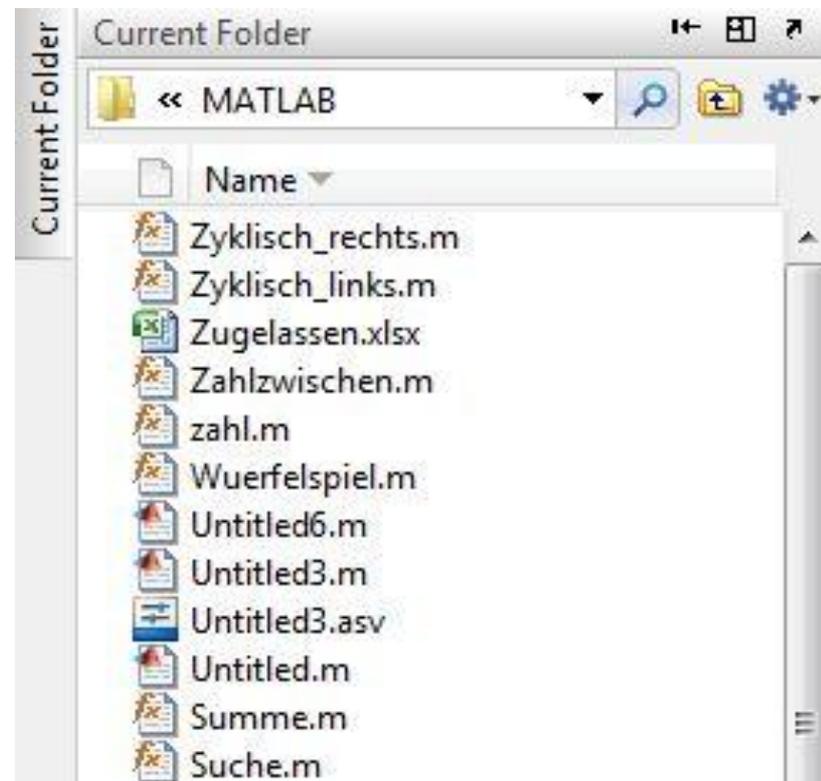
# Skripte in MATLAB (2)

- Neue Skripte kann man erstellen mit
  - Dem Menü-Befehl File/New/Script
  - <Strg>-N innerhalb des Editors
- Ein neues Editor-Fenster wird geöffnet.
- Sinnvoller Weise speichert man das neue Skript schon vor der Fertigstellung unter dem zukünftigen Namen ab:
  - File/Save As...
- Auch hier empfehlen sich „sprechende“ Dateinamen.



# MATLAB: Current Folder

- Im Fenster Current Folder werden die Dateinamen des aktuellen Verzeichnisses angezeigt.
- Dateien mit der Erweiterung „.m“ sind MATLAB-Programme oder -Funktionen.
- Skripte werden immer erst im aktuellen Verzeichnis gesucht.

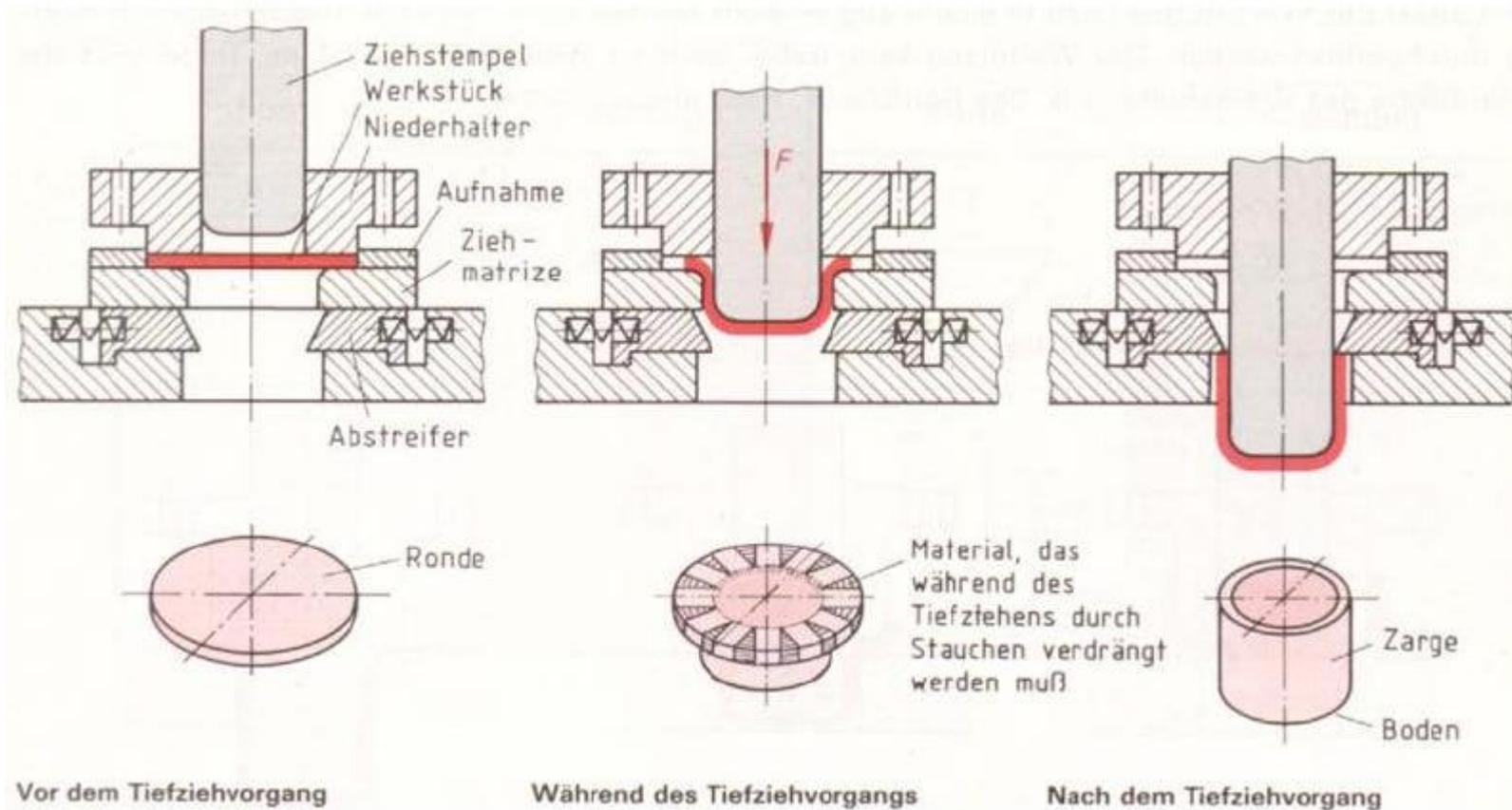


# Interaktive Eingaben

- Soll ein Skript mehrmals mit veränderlichen Parameterwerten ausgeführt werden, so kann man die Werte vom Nutzer interaktiv eingeben lassen.
- Ausgabe einer Meldung im Command Window und warten auf eine Benutzereingabe:  
    `D=input('Bitte geben Sie den Ausgangsdurchmesser in dm an:')`  
    Zwischen den Hochkommata kann ein beliebiger Text stehen.
- Ablauf der Befehlsausführung:
  - Text erscheint auf dem Bildschirm.
  - Skript wartet auf Eingabe des Nutzers.
  - Eingegebener Wert wird der Variable zugewiesen (im Bsp. oben der Variablen D).
- Analog erfolgt eine Ausgabe von Text und Variablenwerten im Command Window:  
`disp('Maximales Volumen:');`  
`disp(Vmax);`
- Zur Ausgabe ebenfalls wichtig: `fprintf` und `sprintf`.

# Beispiel: Extremwertaufgabe

- In einer Stanzerei werden kreisrunde Reststücke aus Stahlblech tiefgezogen. Die Reststücke haben variierende Durchmesser  $D$ . Ziel ist es, aus jedem Reststück einen Napf mit möglichst großem Volumen  $V$  herzustellen. → Extremwertaufgabe



# Beispiel: Extremwertaufgabe

- Mathematische Lösung: Für jeden Kreisdurchmesser  $D$  ist eine Extremwertaufgabe zu lösen ( $r$ : gewählter Napfradius,  $h$ : gewählte Napfhöhe):

$$\text{Extremalbedingung} : V = \pi * r^2 * h$$

$$\text{Nebenbedingung} : 2 * r + 2 * h = D \Leftrightarrow h = \frac{D}{2} - r$$

Nebenbedingung eingesetzt in Extremalbedingung :

$$V = \pi * r^2 * \left(\frac{D}{2} - r\right) = \pi * r^2 * \frac{D}{2} - \pi * r^3$$

Extrema von  $V$  mit erster Ableitung bestimmen :

$$V' = \pi * r * D - 3 * \pi * r^2, \quad V'' = \pi * D - 6 * \pi * r$$

$$1. \quad V' = 0$$

$$\Leftrightarrow \pi * r * D - 3 * \pi * r^2 = 0$$

$$\Leftrightarrow r * (D - 3 * r) = 0$$

$$\Leftrightarrow r = 0, \quad r = \frac{D}{3}$$

$r = 0$  entfällt als Maximum von  $V$ , Kontrolle mit zweiter Ableitung :

$$2. \quad V'' \neq 0$$

$$V''\left(\frac{D}{3}\right) = \pi * D - 2 * \pi * D = -\pi * D$$

$$\Rightarrow \text{Maximum von } V \text{ bei } r = \frac{D}{3}, \text{ dann ist } h = \frac{D}{3}.$$

$$\text{Das Volumen des Napfes ist dann } V = \pi * \frac{D^3}{54}.$$

# Beispiel: Extremwertaufgabe

- Schwieriger würde die Aufgabe, wenn zusätzliche Nebenbedingungen (zum Beispiel  $h$  muss mindestens 1 cm groß sein) gelten würden.
- In MATLAB kann eine derartige Extremwertaufgabe näherungsweise sehr einfach in Form von automatisierten Berechnungen gelöst werden.
- Auch derartige Nebenbedingungen, können einfach in MATLAB realisiert werden.

# Beispiel: Extremwertaufgabe

```
D=input('Bitte geben Sie den Kreisdurchmesser in dm an:')
Vmax=0;
for r=0:0.01:D
    h=D/2-r; %Nebenbedingung
    V=pi*r^2*h; %Extremalbedingung
    if V>=Vmax
        Vmax=V;
        rgewaehlt=r;
        hgewaehlt=h;
    end
end
str=sprintf('Der zu wählende Radius des Napfes liegt bei %s dm', rgewaehlt);
str1=sprintf('Die resultierende Hoehe liegt bei %s dm', hgewaehlt);
str2=sprintf('Daraus ergibt sich das maximale Volumen des Napfes von %s
    dm^3', Vmax);
disp(str)
disp(str1)
disp(str2)
```

# Beispiel: Extremwertaufgabe

```
D=input('Bitte geben Sie den Kreisdurchmesser in dm an:')
Vmax=0;
for r=0:0.01:D
    h=D/2-r; %Nebenbedingung
    V=pi*r^2*h; %Extremalbedingung
    if (V>=Vmax) & (h>=0.1) %zusätzliche Nebenbedingung
        Vmax=V;
        rgewaehlt=r;
        hgewaehlt=h;
    end
end
str=sprintf('Der zu wählende Radius des Napfes liegt bei %s dm', rgewaehlt);
str1=sprintf('Die resultierende Hoehe liegt bei %s dm', hgewaehlt);
str2=sprintf('Daraus ergibt sich das maximale Volumen des Napfes von %s
    dm^3', Vmax);
disp(str)
disp(str1)
disp(str2)
```