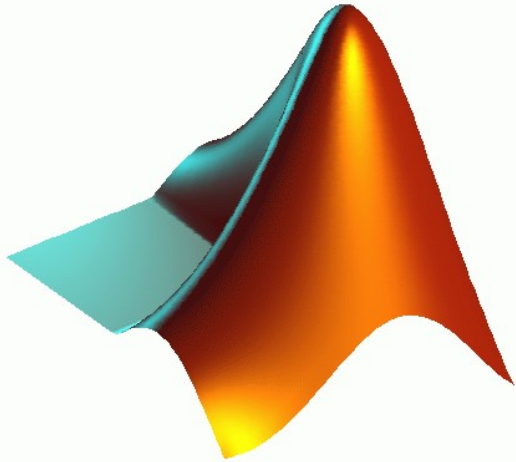
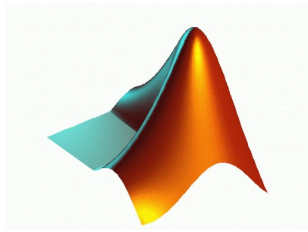


Rechnerpraktikum – Grundlagen der Messtechnik



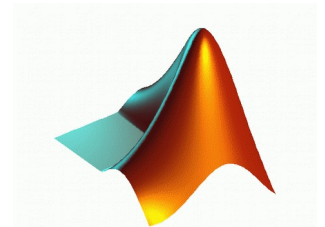
Einführung in
MATLAB

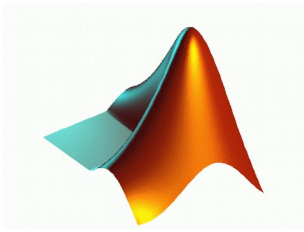
Lehrstuhl für Mechatronik
Universität Rostock
Wintersemester 2018/2019



Gliederung

- MATLAB-Fenster
- Numerische Rechenoperationen
(Rechnen mit Zahlenwerten)
- Symbolische Rechenoperationen
(Rechnen mit Symbolen und Variablen)
- Kontrollstrukturen
- M-Files





MATLAB-Fenster

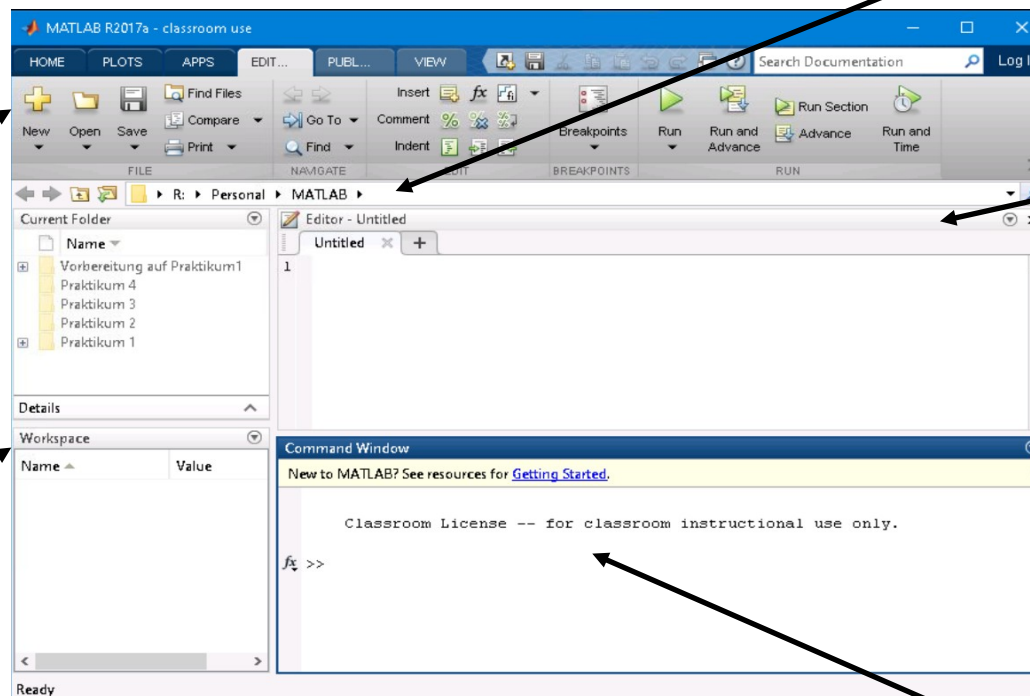
Öffnen einer
neuen MATLAB -
Datei (M-File)

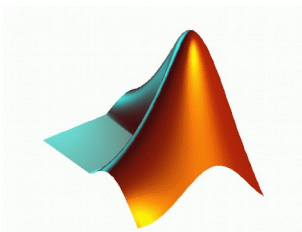
Current Directory /
Workspace

Aktuelles Arbeitsverzeichnis

Editor

Command Window





MATLAB-Fenster

Command Window

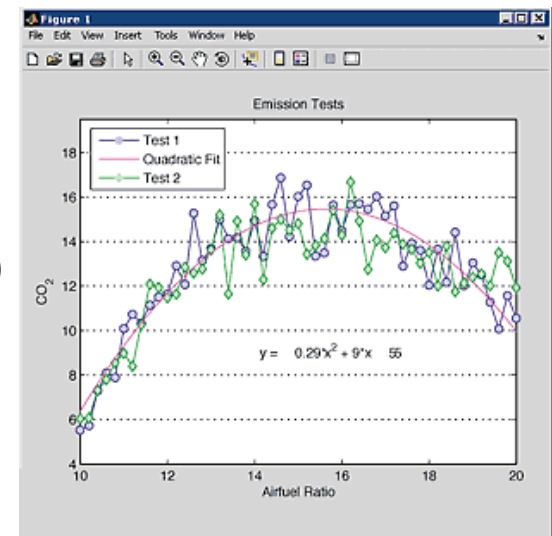
- dient der Eingabe von Befehlen (Funktionsaufrufe, Rechenoperationen)
- Ausgabebereich für Ergebnisse

Current Directory

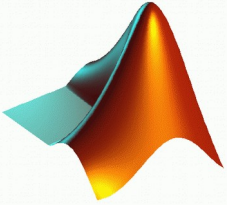
- Inhalt des gegenwärtiges Arbeitsverzeichnisses
- Aufruf sowie Speichern von Dateien (*.fig, *.mat)

Workspace

- Zwischenablage der verwendeten Variablen
- Grafische Ausgabe von Vektoren und Matrizen (Plot)
- Aufrufen und Anzeigen ausgewählter Variablen



Plots in MATLAB



MATLAB-Fenster

Command History

- Verlauf der letzten Befehle über einen bestimmten Zeitraum
- Anwählen von in der Vergangenheit ausgeführten Befehlen

Hilfe!

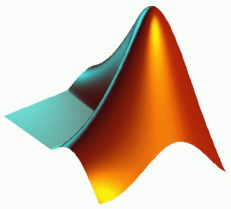
- Zugang zum MATLAB-Handbuch
- Erklärung des verfügbaren Leistungsumfanges (Befehle, Funktionen)
- Anschauliche Beispiele und Demos

Aufrufen der Hilfe im Command Window

`>> help Funktion`



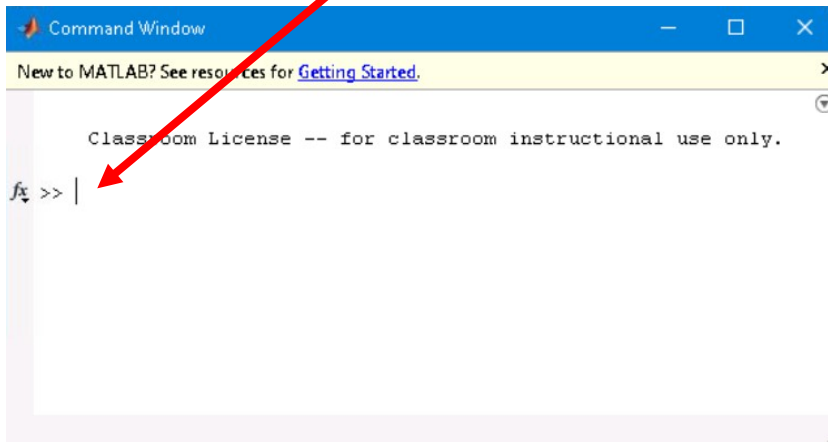
Führt zur Kurzerklärung mit einem Link zur entsprechenden Stelle im Handbuch



Numerische Berechnungen

Rechnen mit Matlab

- Eingabe des Berechnungsbefehls nach dem Doppelpfeil
- anschließend ‚Enter‘ drücken



Mathematische Funktionen

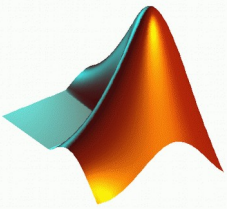
- **cos, sin, sqrt, log, log10, exp** usw.

Operatoren

- **+, -, *, /, =, ~=, >=, <=**

- es gilt:

Punktrechnung vor Strichrechnung



Numerische Berechnungen

Semikolon:

Abtrennung in der Spalte

Alternativ: Zeilenumbruch

Matrix: `>> A = [1, 2, 3; 4, 5, 6; 1, 2, 3];`

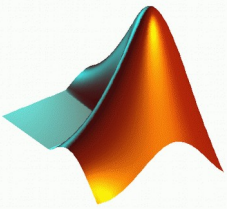
Komma:

Abtrennung in der Zeile

*Hinweis: Komma kann
weggelassen werden;
ist aber zur Lesbarkeit sehr
sinnvoll*

Semikolon:

Ausgabeunterdrückung im
Command Window



Numerische Berechnungen

Rechnen mit Matrizen und Vektoren

- Eingabe eines Vektors

```
>> a = [1; 4; 1]
```

```
a =
```

```
1
```

```
4
```

```
1
```

- Eingabe einer Matrix

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

```
A =
```

```
1 2 3
```

```
4 5 6
```

```
1 2 3
```

- Durchführen einer Transposition mit `\'` bei Vektoren und Matrizen
bei komplexwertigen Ausdrücken `.'` verwenden (andernfalls transponiert
und konjugiert komplex)

```
>> a'
```

```
a =
```

```
1 4 1
```

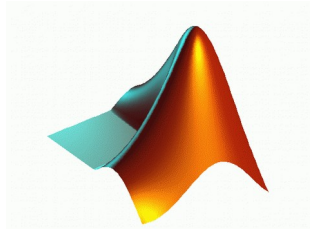
```
>> A'
```

```
A =
```

```
1 4 1
```

```
2 5 2
```

```
3 6 3
```



Numerische Berechnungen

- Ausgabe von gewünschten Elementen/ Zeilen/ Spalten

Vektor

```
>> a(2)
```

```
ans =
```

```
4
```

Matrix

```
>> A(2,:)
```

```
ans =
```

```
4 5 6
```

```
>>A(:,2)
```

```
ans =
```

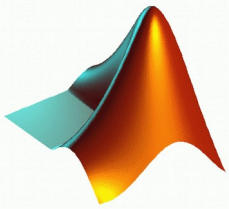
```
2  
5  
2
```

- Erzeugen eines Vektors mit [Startwert: Schrittweite: Endwert]

```
>> a = [0: 0.1: 0.5]
```

```
a =
```

```
0 0.1 0.2 0.3 0.4 0.5
```



Numerische Berechnungen

Rechnen mit Polynomen

- Polynom: $p_1 = 3x^2 + 3.4x - 5$

- in MATLAB als Vektor: `>> p1 = [3 3.4 -5];`

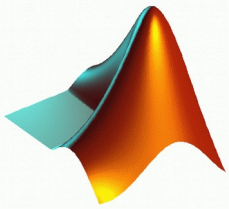
- Nullstellen/Wurzeln: `>>nst = roots(p1)`

```
nst =  
    -1.9766  
     0.8432
```

- Rekonstruktion des Polynoms:

```
>>p1 = poly(nst)  
p1 =
```

```
1.0000 1.1333 -1.6667
```



Numerische Berechnungen

Rechnen mit Polynomen

- Polynommultiplikation:

```
>> p1 = [3 3.4 -5];  
      p2 = [1 2 3];
```

```
>> p3 = conv(p1,p2);
```

- Polynomdivision:

```
>> p4 = deconv(p1,p2);
```

Reihenfolge beachten:

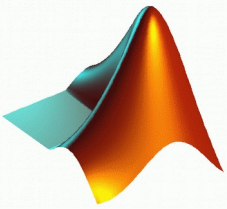
p1 – Dividend p2 - Divisor

- Approximieren eines Polynoms aus gegebenen Vektoren:

```
>> x = [0: 0.5: 10];  
      y = [2.9 2.7 4.8 5.3 7.1 7.6 7.7 ...  
          7.6 9.4 9.0 9.6 10.0 10.2 9.7 ...  
          8.3 8.4 9.0 8.3 6.6 6.7 4.1];
```

```
>> p1 = polyfit(x,y,n);
```

← Ordnung des Polynoms



Numerische Berechnungen

Rechnen mit Polynomen

- Approximationsbefehl (Ausgleichspolynom)

```
>> p1 = polyfit(x,y,n);
```

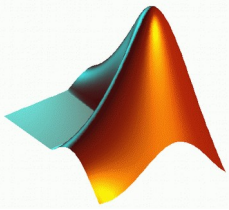
- Erstellen einer grafischen Ausgabe (Plots)

generiert eine monoton steigende Zahlenfolge aus Minimal und Maximalwert des Vektors **x**

```
>> x_i = linspace(min(x), max(x));  
y_i = polyval(p1,x_i);  
plot(x_i, y_i); grid on;
```

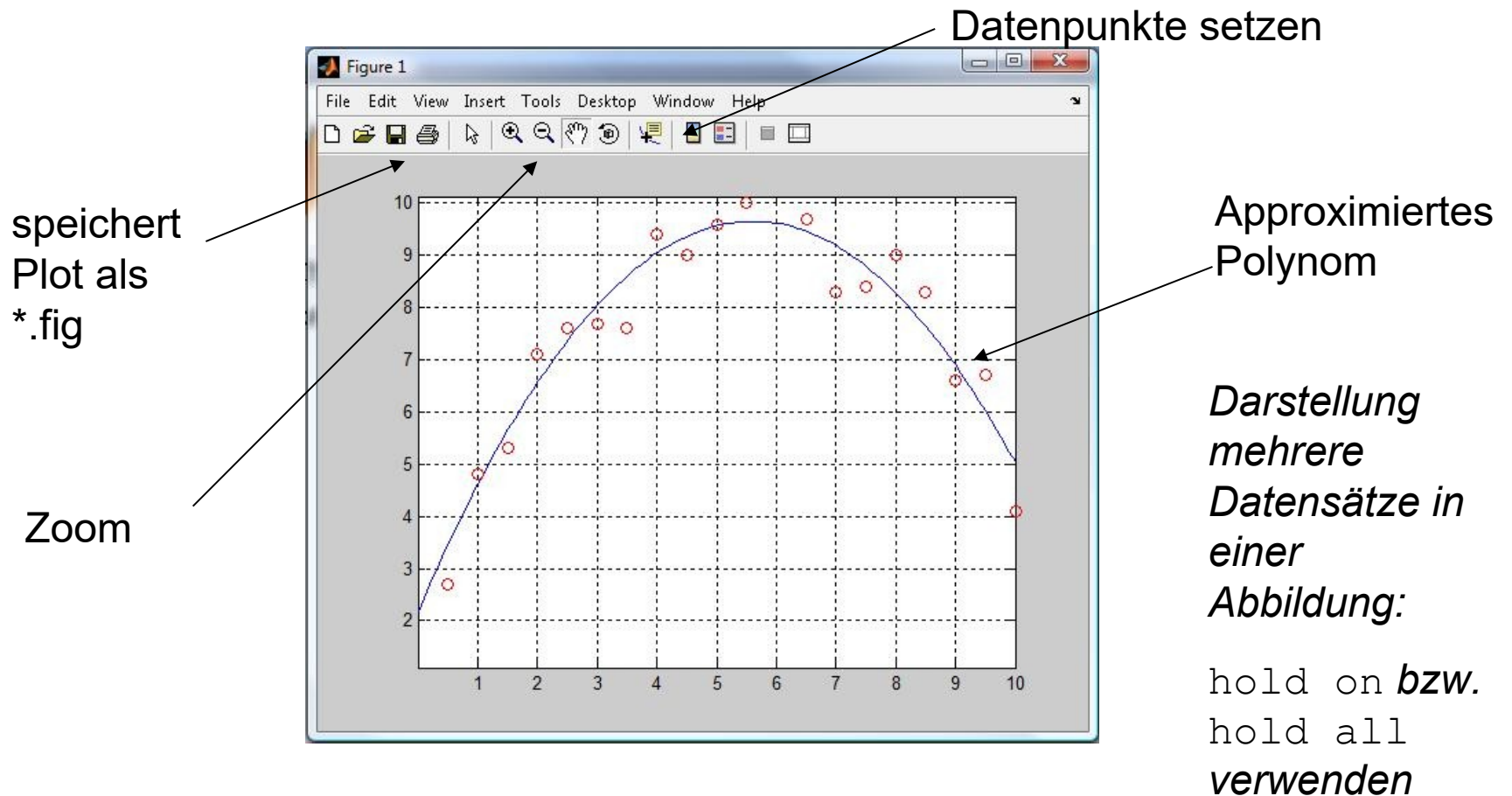
Berechnet neue y-Werte aus dem Polynom **p1** und der Zahlenfolge **x_i**

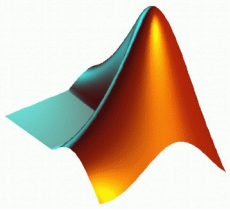
Befehl zur Erstellung des Diagramms



Numerische Berechnungen

- Plotausgabefenster der Approximation:





Symbolische Berechnungen

Anforderung an MATLAB

- Symbolic Math Toolbox
- Extended Symbolic Math Toolbox

Deklaration einer Variable als Symbol

```
>> syms x;  
>> h = 2*x^2 + x
```

```
h =  
      2*x^2 + x
```

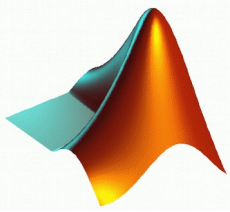
```
>> subs(h,x,2)
```

```
ans =  
      10
```

← konvertiert **x** in ein Symbol

← Aushabe der Funktion **h** in Symbolen

← substituiert in der Funktion **h** das Symbol **x** mit dem Wert **2** und errechnet das Ergebnis



Symbolische Berechnungen

Berechnung eines Grenzwertes

$$y_{\rightarrow 0} = \lim_{x \rightarrow 0} \left[\frac{1}{x} \right] \quad \text{ist in MATLAB}$$

```
>> y_0 = limit(1/x, x, 0)
```

Lösen von Gleichungen

$$0 = ax + b \quad \text{ist in MATLAB nach } a \text{ aufgelöst}$$

```
>> solve('a*x+b','a')
```

```
ans =
```

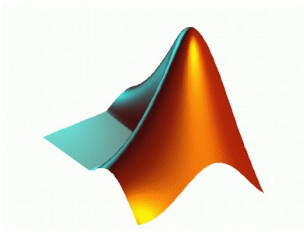
```
-b/x
```

Lösen von Differentialgleichungen

$$\frac{dy}{dt} + y = t \cdot e^t \quad \text{Lösung ist in MATLAB}$$

```
>> y = dsolve('Dy+y = t*exp(t)')
```

```
y =  
1/2*t*exp(t)-1/4*exp(t)+exp(-t)*C1
```



Kontrollstrukturen

„for“-Schleife

Wiederholt eine Anweisung
für die Indizes
 $i=a, i=a+1, \dots i=b$

```
>> for i = a:b
```

```
    Anweisung
```

```
end
```

„if“-Schleife

Nach Überprüfung einer
Bedingung wird die jeweilige
Anweisung ausgegeben

```
>> if Bedingung  
    Anweisung
```

```
    else  
    Anweisung
```

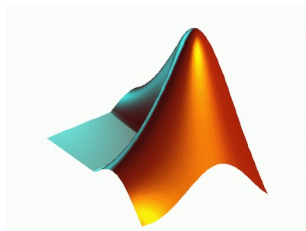
```
end
```

„while“-Schleife

Wiederholt eine Anweisung solange „Bedingung“ erfüllt ist

```
>> while Bedingung  
    Anweisung
```

```
end
```



M-Files

- Öffnen einer neuen Datei mit 'strg-N' oder über 'New – Script' bzw. 'New – Function'
- M-Files werden bei der Ausführung zeilenweise abgearbeitet

Schließen aller offenen Plot-Fenster

Bereinigung von Workspace und Command Window

A screenshot of the MATLAB Editor window titled "Editor - Untitled*". The window contains a script with the following code:

```
1 close all;  
2 clear all;  
3 clc;  
4  
5 %Kommentar  
6  
7 a=2;  
8 b=3;  
9  
10 x=[0:0.1:1];  
11 y=a*x+b;  
12  
13 plot(x,y)
```

Three arrows point from external text to specific parts of the code: one points to the '+' button in the tab bar, another points to the 'clear all;' line, and a third points to the '%Kommentar' line.

Kommentare mit % werden bei der Ausführung nicht berücksichtigt