Einführung in MATLAB
für Maschinenbauer im ersten Semester

Autor: Dr. Christian Karpfinger et al.
Stand: 5. Dezember 2015

Inhaltsverzeichnis

1 Grundlagen

1.1 Die Benutzeroberfläche

Willkommen im Einführungskurs MATLAB. In der ersten Lektion sollen zunächst einmal die Grundlagen des Programms erklärt werden. Als erstes werden hierzu die verschiedenen Arbeitsflächen vorstellt.

Nach dem Öffnen von MATLAB erscheint folgende Benutzeroberfläche:

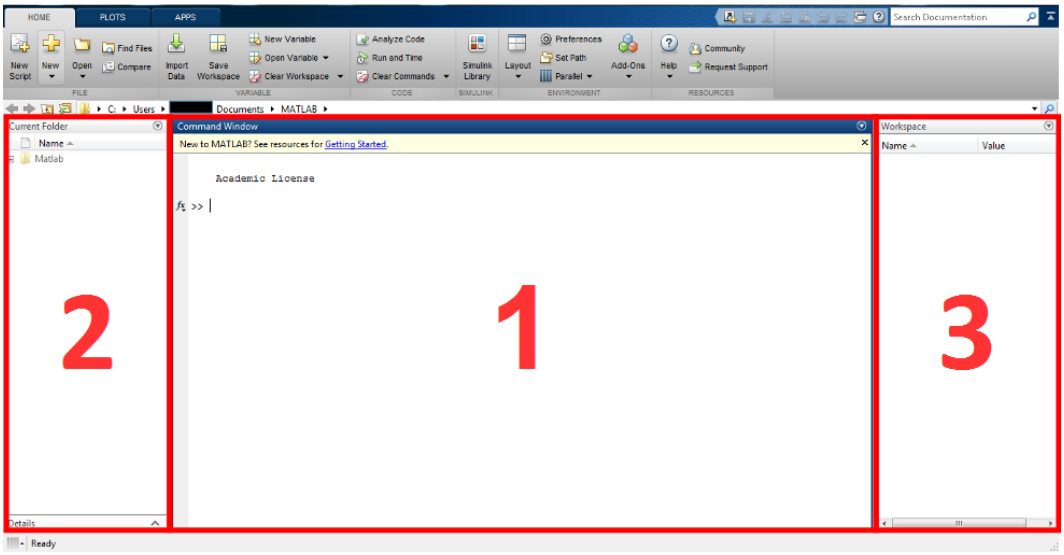

Abbildung 1.1: Command Window (1), Current Folder (2) und Workspace (3)

Im mittleren Bereich befindet sich das Command Window (Abbildung 1.1, Nummer 1). Hier werden einzelne bis mehrzeilige Befehle eingetragen. Bei mehrzeiligen Rechnungen werden Zeilenumbrüche durch gleichzeitiges Drücken von [Shift] und [Enter] eingegeben.

Die linke Arbeitsfläche (Abbildung 1.1, Nummer 2) zeigt an, in welcher Datei im Moment gearbeitet wird. Dabei werden die aktuelle Datei sowie der benutzte Dateipfad angegeben. Für eine genauere Erklärung siehe Lektion BLABLA.

Die letzte Arbeitsfläche ist der sogenannte Workspace (Abbildung 1.1, Nummer 3). Hier werden die verwendeten Variablen mit ihren Werten und Größen angezeigt.

1.2 Zuweisen und Löschen von Variablen

Eingegebene Befehle können durch das Drücken der Taste [Enter] an MATLAB geschickt werden. Wir betrachten zunächst die Eingabe der Zahl 2 in das Command Window. Man tippt also 2 ein und drückt dann auf [Enter].

Im Folgenen sehen wir die Darstellung von Code, die in dieser MATLAB-Einführung immer wieder verwendet wird: Im linken (dunkelgrauen) Bereich befindet sich die Eingabe, hier der Wert 2. Im rechten (hellgrauen) Bereich ist das zu sehen, was MATLAB daraufhin im Command Window ausgibt.

`>> 2`	**ans** = 2

MATLAB weist also der Variablen *ans* den Wert 2 zu und gibt das auch aus. Jedem Wert, dem kein eigener Variablenname zuordnet wurde, wird mit dem Variablennamen *ans* abgespeichert, was für *answer* steht. Wenn man mehreren Werten keine Variable zuordnet, dann wird in der Variablen *ans* jedes Mal der neue Wert gespeichert und die alten Werte gehen verloren.

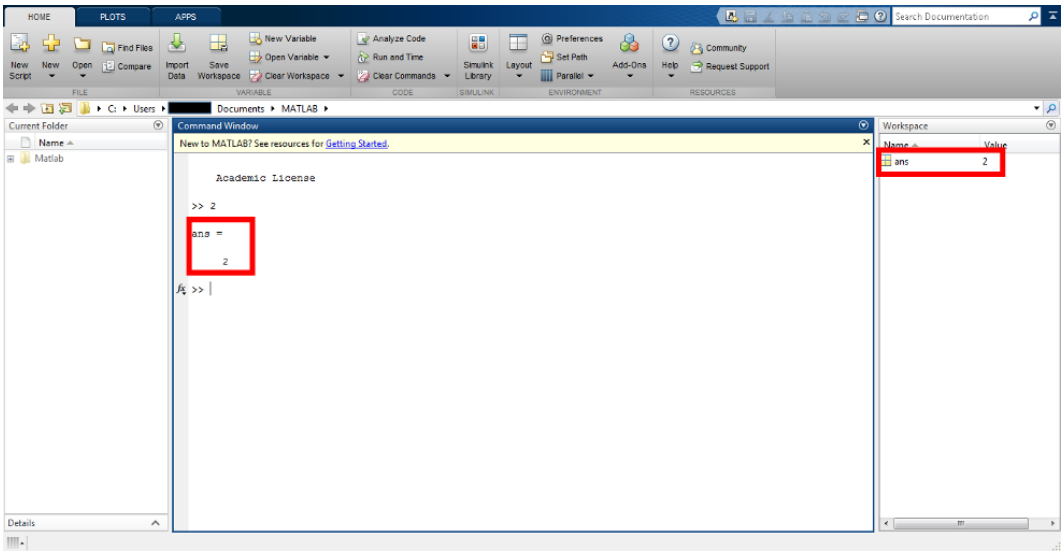

Abbildung 1.2: Defintion der Variablen *ans* mit dem Wert 2

In der Abbildung 1.2 sehen wir, dass die Variable *ans* mit dem Wert 2 im Workspace gespeichert wird. Geben wir nun 5 in das Command Window ein. Jetzt ändert sich die Variable *ans* im Workspace auf 5 und der Wert 2 geht verloren.

Um dies zu verhindern, können Variablennamen vergeben werden. Als Beispiel betrachten wir die Zuweisung $a = 2$, die der Variablen a den Wert 2 zuweist. Diese Gleichung $a = 2$ wird also in das Command Window eingetippt und es wird wieder [Enter] gedrückt, um den Befehl abzuschicken.

```
>> a=2
```

```
a=2
```

Nun ist im Workspace der Variablen a der Wert 2 zugeordnet worden und dies wird auch im Command Window angezeigt. Die Variable ans mit dem Wert 5 ist dabei weiterhin im Workspace vorhanden.

Man muss aufpassen, in welcher Reihenfolge eine Variablenzuordnung stattfindet. Links muss die Variable angegeben werden, der etwas zugeordnet wird und rechts der Wert, der von dieser Variablen angenommen werden soll. So gibt zum Beispiel $2 = a$ eine Fehlermeldung aus, obwohl dies mathematisch äquivalent zu $a = 2$ ist. Aber für MATLAB ist nun 2 der Variablenname und a der Wert. Dies ist aber nicht möglich, weswegen eine Fehlermeldung erscheint.

```
>> 2=a
```

```
2=a
 ↑
Error: The expression to the left of
    the equals sign is not a valid
    target for an assignment.
```

Das Anzeigen einer Fehlermeldung im Command Window ist sehr praktisch. Aber das Anzeigen des Variablennamen mit dem Wert reicht manchmal auch nur im Workspace und muss nicht noch einmal im Command Window geschehen. Um dies zu erreichen, muss hinter die Variablenzuweisung ein Semikolon geschrieben werden. Wird zum Beispiel in das Command Window $b = 7$; eingegeben, wird nur im Workspace der Variablen b der Wert 7 zugeordnet. Die Variablenzuweisung erscheint jedoch nicht noch einmal im Command Window, die Ausgabe des Ergebnisses wird unterdrückt.

```
>> b=7;
```

Nun sind der Workspace und das Command Window mit Werten und Text befüllt.

Möchte man das Command Window leeren, muss `clc` in die Kommandozeile geschrieben werden. Hierbei bleiben die Variablen im Workspace erhalten.

Wenn man den Workspace leeren möchte, kann man durch `clear` und dem Variablennamen einzelne Variablen löschen. So wird zum Beispiel mit `clear a` die Variable a gelöscht. Um alle Variablen im Workspace zu löschen, gibt man `clear` in das Command Window ein.

1.3 Aufrufen der Hilfe

Sucht man Erklärungen zu bestimmten Befehlen in MATLAB, so können diese auf mehreren Wegen aufgerufen werden. Es kann zum Beispiel in das Command Window `help` eingeben werden. Dann werden die verschiedenen Themen der Hilfe angezeigt und es kann zu jedem Befehl eine Erklärung aufgerufen werden.

Möchte man eine Erklärung zu einem bestimmten Befehl, so muss dieser hinter `help` geschrieben werden. Um zum Beispiel herauszufinden, was `clc` macht, geben wir in das Command Window `help clc` ein:

```
>> help clc
```

```
clc     Clear command window.
    clc clears the command window and
        homes the cursor.

    See also home.
```

Eine weitere Möglichkeit, Erklärungen zu Befehlen zu erhalten, ist das Anklicken von Getting Started.

Es öffnet sich ein Pop-up-Fenster (Abbildung 1.3), in dem alle Erklärungen sowie Beispiele dazu nachgelesen werden können. Diese Hilfe gibt übrigens es auch online unter http://de.mathworks.com/help/.

1.4 Einfaches Rechnen mit MATLAB

Bis jetzt wurde also erklärt, wie MATLAB aufgebaut ist und wie man in MATLAB Variablen zuweist und löscht. Nun folgt das einfache Rechnen mit MATLAB, also Berechnungen ähnlich denen, die auch ein normaler Taschenrechner kann.

Matlab kann Zahlen und Variablen addieren, subtrahieren, multiplizieren und dividieren,

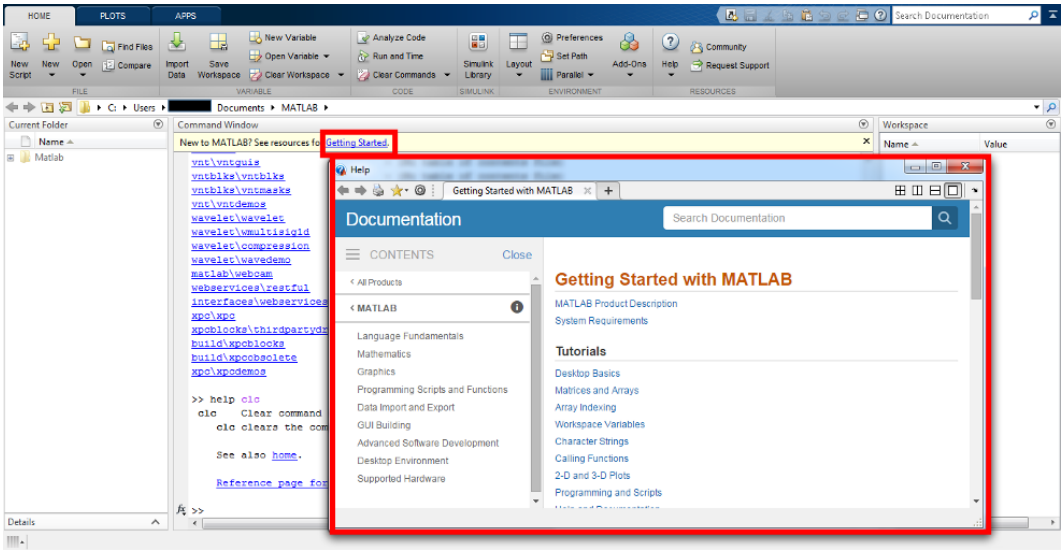

Abbildung 1.3: Erklärungen zu den Befehlen mit Hilfe des Getting Started-Hyperlinks

zum Beipiel:

>> 7+2	**ans** = 9

Dem Ergebnis kann auch wieder eine Variable zugewiesen werden, indem man vor die Rechnung die Variable schreibt:

>> a=8/2+2*9	a = 22

MATLAB beachtet bei allen Rechnungen Punkt vor Strich sowie Klammern. Wir beobachten, dass das Ergebnis 22 im Workspace der Variablen a zugewiesen wurde.

Statt mit Zahlen zu rechnen, kann man auch mit Variablen rechnen. So kann man Variablen Werte zuweisen und dann mit den Variablen Berechnungen durchführen. Ein Beispiel ist folgendes: Der Variablen x wird der Wert 56 zugeordnet und y der Wert 8. Dann wird x durch y geteilt.

>> x=56 >> y=8 >> z=x/y	x = 56 y = 8 z = 7

Nun ist auch im Workspace zu erkennen, dass $x = 56$, $y = 8$ und $z = 7$ gilt.

Die Rechnung x/y kann nun auch mit anderen Werten durchgeführt werden. Möchten wir zum Beispiel $56/7$ berechnen, ändern wir durch $y = 7$ den Wert von y auf 7. Der Wert von z ist jedoch noch immer 7. Also muss die Berechnung von x/y erneut durchgeführt werden. Dies funktioniert, indem man [Pfeiltaste nach oben] drückt und den Befehl dann wie gewohnt durch [Enter] abschickt. Durch die Pfeiltasten nach oben/unten können bereits benutzte Befehle durchsucht und erneut aufgerufen werden. Mit Hilfe dieses Tricks lassen wir nun noch einmal $z = x/y$ berechnen. Nun sind im Workspace alle geänderten Werte zu sehen.

Es gibt auch die Möglichkeit, alten Code in einem eigenen Fenster anzuzeigen.

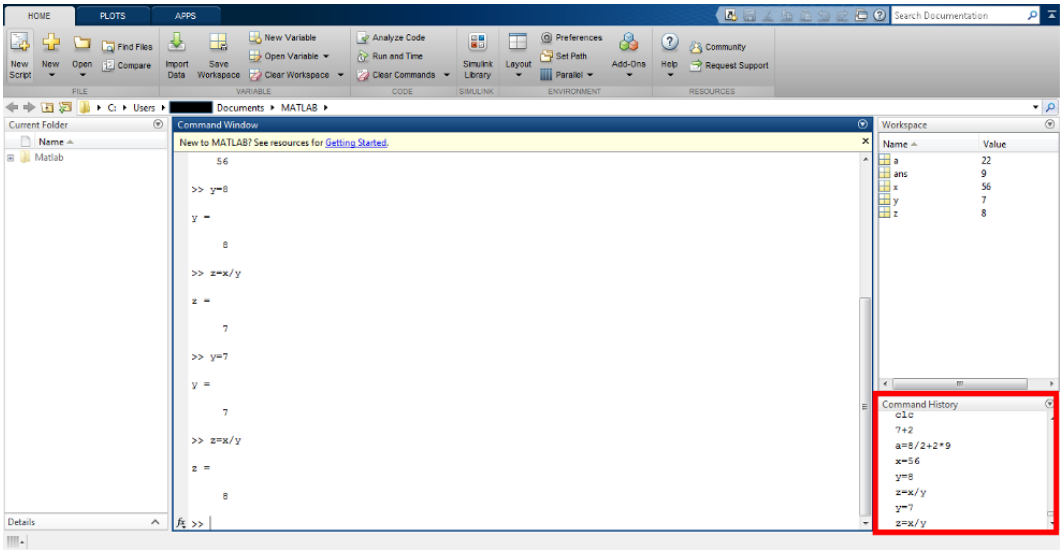

Abbildung 1.4: Eingefügte Command History

Dieses Fenster (Abbildung 1.4) kann man sich unter

$$Home \rightarrow Layout \rightarrow CommandHistory \rightarrow Docked$$

anzeigen lassen und dann an einen gewünschten Ort verschieben.

1.5 Ein paar Rechenbeispiele

Wir wollen im Folgenden einfache Berechnungen durchführen und dabei einige wichtige Befehle kennen lernen.

Man kann Potenzen durch [^] berechnen, zum Beispiel 3 im Quadrat und 4 hoch 3:

```
>> 3^2
>> 4^3
```

```
ans = 9
ans = 64
```

Durch `sqrt(x)` kann die Wurzel aus einer Zahl x gezogen werden. Der Befehl ist übrigens eine Abkürzung für *square root*. Ein Beispiel ist die Wurzel aus 2:

```
>> sqrt(2)
```

```
ans = 1.4142
```

Um die Fläche eines Kreises mit dem Radius $r = 3$ zu berechnen, gehen wir wir folgt vor:

```
>> r=3;
>> A=r^2*pi
```

```

A = 28.2743
```

Wir wollen Sinus und Kosinus des Winkels $\frac{\pi}{3}$ bestimmen. Nach Pythagoras ist deren Summe gleich eins.

```
>> s=sin(pi/3)
>> c=cos(pi/3)
>> s^2+c^2
```

```
s = 0.8660
c = 0.5000
ans = 1
```

Eine Liste weiterer nützlicher Befehle befindet sich im Anhang dieses Skripts.

1.6 Aufgaben

Aufgabe 1. Starten Sie MATLAB und legen Sie drei verschiedene Variablen an. Löschen Sie eine davon wieder aus dem Workspace. Schließen Sie MATLAB. Wie können

die angelegten Variablen bei einem erneuten Programmstart wieder verfügbar gemacht werden? Als Hilfestellung können sie `help save` betrachten.

Was macht der Befehl `whos`?

Aufgabe 2. Legen Sie eine Variable an, die als Wert eine positive ganze Zahl hat, und nehmen Sie die Quadratwurzel daraus. Wiederholen Sie dies dann, ohne den gleichen Befehl zu benutzen. Wiederholen Sie es noch einmal, diesmal ohne mehr als zwei verschiedene Tasten zu drücken.

Aufgabe 3. Welche Ausgabe erwarten Sie bei der Eingabe `n=10, 2*n;` ? Stimmt das Ergebnis mit Ihrer Erwartung überein?

Aufgabe 4. Legen Sie die Variablen $x = 2$, $y = 3$ und $z = -4$ an. Lassen Sie sich nun folgenden Ausdruck von MATLAB berechnen:

$$y - \frac{x}{y + \frac{y+x}{xz}}$$

Aufgabe 5. Finden Sie mit `help` heraus, was der Befehl `floor` macht. Lassen Sie sich danach durch `doc fix` die Dokumentationsseite zum Befehl `fix` anzeigen. Geben Sie eine Zahl a an, für welche die Ausgaben von `floor(a)` und `fix(a)` nicht gleich sind.

Aufgabe 6. Berechnen Sie $\sin 42°$, $\cos(\frac{\pi}{7})$ und den Arkustangens von 1 mit MATLAB.

Aufgabe 7. Welche Zahl erzeugt die MATLAB-Eingabe `7.2e8`? Geben Sie $0,0054$ in dieser Schreibweise an.

Aufgabe 8. Suchen Sie durch Aufrufen der Hilfe eine Funktion, die ein magisches Quadrat anlegt, wobei Sie die Größe des Quadrats festlegen können. Erstellen Sie solch ein magisches Quadrat.

A Befehlsübersicht

Hier sind die wichtigsten Befehle in MATLAB aufgelistet.

Lektion 1: Grundlagen

`atan(a)`
: Der Arkustangens von `a`.

`clc`
: Das Command Window wird geleert.

`clear`
: Löschen aller Variablen aus dem Workspace.

`clear item`
: Löschen von `item` aus dem Workspace.

`cos(a)`
: Der Kosinus von `a`.

`exp(a)`
: Die Exponentialfunktion an der Stelle `a`, also e^a.

`fix a`
: Rundet `a` zur 0 hin, also zur betragsmäßig kleineren Zahl hin.

`floor a`
: Rundet `a` zur $-\infty$ hin, also zur kleineren Zahl hin.

`help`
: Die Hilfe wird geöffnet.

`help name`
: Die Hilfe zum Befehl `name` wird aufgerufen.

`load('dateiname.mat')`
: Laden des Inhalts von `dateiname.mat`.

`log10(a)`
: Der 10er-Logarithmus von `a`.

`magic n`
: Erzeugt ein magisches `n`x`n`-Quadrat.

`pi`
: Die Konstante π.

`save dateiname.mat`
: Speichert alle angelegten Variablen in der Datei `dateiname.mat`

`sin(a)`
: Der Sinus von `a`, wobei `a` in Radiant angegeben ist.

`sind(a)`
: Der Sinus von `a`, wobei `a` in Grad angegeben ist.

`sqrt(a)`
: Die Quadratwurzel von `a`.

`tan(a)`
: Der Tangens von `a`.

`whos`
: Informationen über alle angelegten Variablen.

`whos a`
: Informationen über die Variable `a`, beispielsweise ihre Größe, ihre Klasse und ihren Speicherplatzbedarf.