

Matlab-Übungen zu Deterministische Signale und Systeme 1. Übung

1. Aufgabe

Abbildung 1 zeigt eine typische Ansicht der Matlab-Oberfläche. Die rechte Hälfte des Fensters wird durch die Kommandozeile belegt, in der Sie interaktiv arbeiten können. Links unten ist eine Liste der zuletzt benutzten Befehle zu finden. Weiter oben ist eine Auflistung der Dateien im aktuellen Verzeichnis zu finden, die Sie durch Klicken auf »Workspace« in eine Ansicht der im Moment von Matlab verwendeten Variablen umschalten können. Die Dateiliste stellt auch zahlreiche Funktionen eines Dateimanagers bereit und über die Variablenliste können Sie Variablen anzeigen lassen und manipulieren.

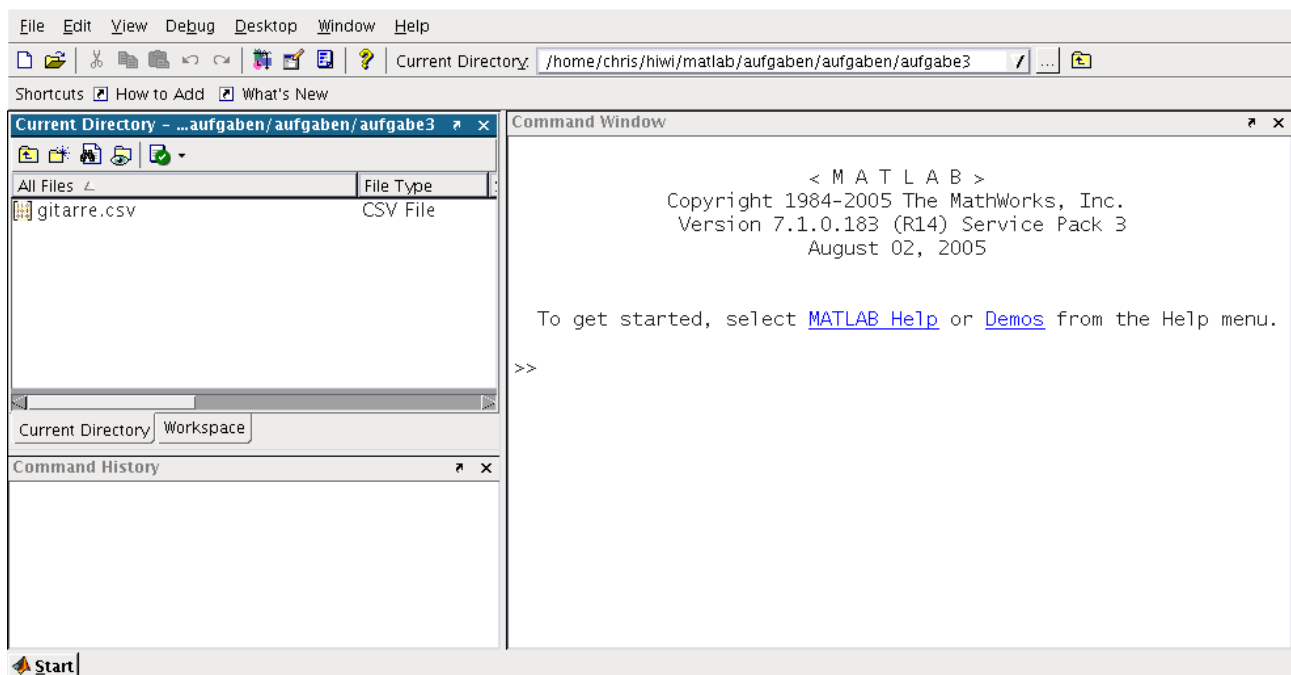


Abbildung 1: Die grafische Oberfläche von Matlab.

Dem »aktuellen Verzeichnis« oder »Arbeitsverzeichnis« (engl. working directory) kommt in Matlab eine besondere Bedeutung zu. Wenn Matlab Funktionen oder Skripte sucht (etwa weil

sie auf der Kommandozeile aufgerufen wurden – siehe dazu auch später in dieser Übung), schaut es zuallererst in diesem Verzeichnis, danach durchläuft es einen Suchpfad (mehr dazu in einer späteren Übung). Sie sollten immer darauf achten, daß das Arbeitsverzeichnis das Verzeichnis ist, in dem Ihre Daten und Matlab-Funktionen liegen, die Sie gerade verwenden (die »eingebauten« Funktionen findet Matlab natürlich auch so).

Sie können sich das Arbeitsverzeichnis auf der Kommandozeile anzeigen lassen, indem Sie **pwd** (»*print working directory*«) eingeben. Mit **ls** (»*list*«) geben Sie den Inhalt des Arbeitsverzeichnisses auf der Kommandozeile aus. Wenn Sie das Arbeitsverzeichnis wechseln wollen, können Sie das Kommando **cd <pfad>** (»*change directory*«) verwenden, welches in den angegebenen Pfad wechselt. **<pfad>** kann dabei eine relative Angabe (etwa **verzeichnis/unterverzeichnis**) oder eine absolute Angabe (etwa **/verzeichnis/unterverzeichnis** – unter Windows bezieht sich diese absolute Angabe immer auf das aktuelle Laufwerk, **/Dokumente** entspricht also etwa **c:\Dokumente**) sein. Die Angabe **..** hat dabei eine Sonderbedeutung, sie wechselt eine Verzeichnisebene tiefer. Ist man etwa im Verzeichnis **/home/user/matlab-uebung**, wechselt **cd ../..** nach **/home**. Mit dieser Syntax folgt Matlab den Konventionen aus der UNIX-Welt, wer z. B. mit Linux arbeitet, kennt diese Befehle vielleicht schon.

Tipp: Lassen Sie sich auch unter Windows nicht von den Schrägstrichen als Pfadtrenner ins Bockshorn jagen, die können zwar auch DOS-typische Pfadangaben verwenden (und etwa den Befehl **dir** statt **ls**), in Matlab steckt jedoch einiges der UNIX-Philosophie und Sie tun sich später leichter, wenn Sie sich das gleich angewöhnen. (Matlab wurde erst relativ spät auch auf Windows portiert, daher diese UNIX-Affinität.)

Nun haben sie schon ein paar »Werkzeuge« kennengelernt und es kann mit den Aufgaben losgehen.

- 1.1. Matlab ist eine auf den Umgang mit Matrizen und Vektoren optimierte Programmiersprache und das ist auch einer der Haupt-Gründe, warum sich dieses Tool in so vielen Wissenschaftsdisziplinen als Standardwerkzeug etabliert hat.

Erzeugen Sie mehrere Variablen, die unterschiedliche Datentypen enthalten, nach dem folgenden Beispiel (verwenden Sie dazu den Zuweisungsoperator **=**). Bei Zuweisungen steht immer links vom Operator der Variablenname und rechts der zugewiesene Wert):

```
a = 3
b = 5;
c = [1 2 a 4 b]
d = c.'
e = [a b; b a]
f = [b, a; 1, 2];
g = 'Hallo'
h = '␣Welt!';
i = [g h]
j = 0:.1:1
% k = j
```

Sie können sich den Inhalt der entstandenen Variablen ansehen, indem Sie einfach den Variablennamen auf der Kommandozeile eingeben (und dann die Eingabetaste drücken).

Was enthalten die Variablen **a** bis **k**? Nennen Sie den Matlab-Datentyp und die mathematische Bezeichnung (falls möglich).

Unterscheidet Matlab zwischen Skalaren, Vektoren und Matrizen?

Finden Sie heraus, was die Funktionen **who** und **whos** machen und erklären Sie, wofür das Semikolon am Ende einer Zeile und das Prozentzeichen stehen.

- 1.2. Im letzten Aufgabenteil haben sie mehrere Vektoren und Matrizen in Variablen gespeichert. Finden Sie heraus, wie man Spaltenvektoren direkt anlegt (ohne Transponierung).
- 1.3. Speichern sie das aktuelle Arbeitsverzeichnis (also die Ausgabe von **pwd**) in der Variablen **pfad** und lassen Sie sich auf der Kommandozeile sowohl **pfad** als auch **pfad.** ausgeben. Was für ein Datentyp ist die Ausgabe von **pwd** und in welcher Form speichert Matlab diesen?







2. Aufgabe

Mit **clear all** können Sie den Workspace, also alle Variablen, die Matlab momentan verwendet, löschen. Nutzen Sie diesen Befehl zu Beginn jeder Aufgabe, um das System auf den »Ausgangszustand« zurückzusetzen.

Gegeben seien der Vektor $\vec{b} = (1 \ 2)$ und die Matrizen

$$A = \begin{pmatrix} 1 & 7 \\ 2 & 3 \end{pmatrix}, B = \begin{pmatrix} 5 & 3 \\ 4 & 2 \end{pmatrix} \text{ und } C = B^T.$$

- 2.1. Ordnen Sie die Beschreibung den entsprechenden Rechenoperationen zu, indem Sie die oben angegebenen Vektoren und Matrizen in Variablen speichern und mit diesen dann ausprobieren, was die angegebenen Rechenoperationen bewirken (eine Zuordnung ist schon angegeben). Die grauen Kästchen dienen als Platzhalter für die einzusetzenden Variablen.

	Addition
	Transponierung
	Elementweise Division
	Elementweise Multiplikation
	Multiplikation, je nach Konstellation
	Produkt zweier Matrizen, Skalarprodukt etc.
	Subtraktion

- 2.2. In Matlab sind die Operatoren \backslash und $/$ definiert, die es, zumindest auf Vektoren und Matrizen angewendet, in der Mathematik so nicht gibt. Finden sie mit Hilfe der Matlab-Hilfe heraus, was diese Operatoren machen und beschreiben Sie es kurz.

3. Aufgabe

In dieser Aufgabe geht es um die Darstellung von mathematischen Funktionen, also dem Umgang mit etwas wie $f(x) = x^2$, nicht zu verwechseln mit Funktionen im Sinne von »Unterprogrammen« bei der Programmierung, welche wir später kennenlernen werden.

- 3.1. Speichern Sie Werte von -2 bis 2 im Abstand von $0,1$ in einem Zeilenvektor **x1** und die Werte -2 , 0 und 2 in einem weiteren Zeilenvektor **x2**. Speichern Sie nun in einem Zeilenvektor **y1** das Quadrat von **x1** und in **y2** das Quadrat von **x2**. Stellen Sie die Funktionen $y_1 = x_1^2$ bzw. $y_2 = x_2^2$ mit

```
plot(x1, y1, x2, y2)
```

dar. Um die richtige Zuordnung der entstandenen Graphen zu erleichtern, können Sie mit

```
legend('y_1=f(x_1)', 'y_2=f(x_2)')
```

eine Legende einblenden.

- (3.1.1) Warum unterscheiden sich die beiden Darstellungen, obwohl beide die Funktion $f(x) = x^2$ darstellen?
- (3.1.2) Welche Werte bekommt der **plot**-Befehl von Matlab übergeben?
- (3.1.3) Wie arbeitet der Befehl **legend**?
- (3.1.4) Schätzen Sie durch Ablesen aus dem Graphen ab, wie groß der Fehler gegenüber dem genauen Wert der Funktion $f(x) = x^2$ bei der Darstellung von $y_2 = f(x_2)$ bei $x = 1$ ist (Sie können die Rechengenauigkeit von Matlab in dieser Aufgabe vernachlässigen!). Der befehl **grid** könnte Ihnen dabei helfen.

- 3.2. Haben Sie eine Idee, wie Sie den tatsächlichen Fehler bei $x = 1$ bestimmen können? Wenn ja, setzen Sie diese Idee um und bestimmen Sie damit den Fehler.

- 3.3. Was passiert, wenn Sie beim Plotten einer einzelnen Funktion den Vektor für die x -Werte weglassen. Was passiert, wenn Sie beim Plotten beider Funktionen die x -Werte weglassen?

- 3.4. Mit den Befehlen **xlabel**, **ylabel** und **title** können Sie Diagramme in Matlab beschriften. Finden Sie mit Hilfe der Hilfe heraus, wie die Befehle aufgerufen werden und geben Sie dem Diagramm aus den vorangegangenen Aufgabenteilen eine sinnvolle Beschriftung.

- 3.5. Wenn Sie

```
x = [ 1 0 -1 0 1 ];  
y = [ 0 1 0 -1 0 ];
```

```
figure;  
plot(x, y);
```

in die Matlab-Eingabeaufforderung eingeben, erzeugt das ein auf einer Ecke stehendes Quadrat. Erzeugen Sie die Darstellung des Einheitskreises.

- 3.6. Der Umgang und die Darstellung von Signalen geschieht in Matlab auf die gleiche Weise wie bei Funktionen. Stellen Sie das Signal $y(t) = \sin(2 \cdot \pi \cdot f_0 t)$ für $0 \leq t \leq 1$ s mit $f_0 = 5$ Hz dar. Wählen Sie dazu zuerst eine geeignete Zeitachse.

4. Aufgabe

Matlab kann von Hause aus mit komplexen Zahlen umgehen. Um diese eingeben zu können, kennt Matlab die imaginäre Einheit i bzw. j .

Um die Lösung eines Polynoms zu bestimmen, kennt Matlab die Funktion **roots**, welche als Eingabe einen Vektor erwartet, der die Koeffizienten des Polynoms in absteigender Reihenfolge enthält. Die Gleichung $5x^3 + 7x^2 - 2x + 1 = 0$ läßt sich also mit

```
roots([ 5 7 -2 1 ])
```

numerisch lösen.

Die Aufgaben sind teilweise aus **mtut2** entnommen.

4.1. Berechnen Sie j^n mittels einer **for**-Schleife für $n = 1, 2, \dots, 10$, $n = 12$ und $n = 17$. Welche Variable sollten Sie hier besser nicht für die Zählschleife verwenden?

4.2. Ermitteln Sie die Lösung(en) der folgenden Gleichungen.

(4.2.1) $z^4 - 1 = 0$

(4.2.2) $z^4 + 1 = 0$

(4.2.3) $z^5 = 32j$

Stellen Sie die Lösungen in einem Diagramm, das die komplexe Ebene darstellt, dar.

4.3. Gegeben sei der Vektor $\vec{a} = (1 \ 2j \ 3 \ 4j)$. Benennen Sie den Unterschied zwischen den Operationen **a'** und **a.'** in Matlab.

4.4. Berechnen Sie mit den komplexen Zahlen $u = 3 + 2j$ und $v = 4 - 5j$ die Werte von

(4.4.1) $u + v$

(4.4.2) $2u - 3v$

(4.4.3) $u + v^*$

(4.4.4) $u^* + v^*$

(4.4.5) $(u \cdot v)^*$

4.5. Rechnen Sie die Zahlen $Z_1 = 3,247 + 1,254j$ und $Z_2 = -3,247 - 1,254j$ in die Darstellung $Z_n = a_n \cdot e^{j\phi_n}$ um. Tipp: Verwenden Sie die Funktionen **abs**, **atan**, **real** und **imag**. Überprüfen Sie die berechneten Werte zeichnerisch (mit Papier, Bleistift und Geo-Dreieck). Sind die Ergebnisse korrekt? Wenn nicht, welches Ergebnis ist falsch und warum?

4.6. Bringen Sie die Zahl $Z_3 = -3,247 - 6,125j$ in die Exponentialdarstellung. Verwenden Sie diesmal statt **atan** die Funktion **atan2**.

4.7. Mit **a == b** können Sie überprüfen, ob zwei Werte den gleichen Inhalt haben. Falls ja, ist das Ergebnis 1, wenn nein, dann 0. (Mit Strings funktioniert diese Überprüfung nicht, sollten Sie das benötigen, können Sie die Funktion **strcmp** bemühen.) Es sei $Z_4 = \pi + je$, wobei e die Eulersche Zahl darstellt. Überprüfen Sie die folgenden Aussagen auf die beschriebene Weise.

$$(4.7.1) \quad Z_4 \cdot Z_4^* = |Z_4|^2$$

$$(4.7.2) \quad Z_4 \cdot Z_4^* = |Z_4|^2$$

$$(4.7.3) \quad e^{j\pi} = -1$$

Entsprechen die Ergebnisse dem, was Sie erwartet haben? Wenn nein, versuchen Sie herauszubekommen, warum das Ergebnis von dem von Ihnen erwarteten Ergebnis abweicht.