

Wichtige Konzepte in Matlab:

Matrix:

Matlab ist so aufgebaut, dass nahezu alle Daten in Form von Matrizen (also in tabellarischer Anordnung) dargestellt werden. Matrizen können prinzipiell beliebig viele Dimensionen haben. Bei einer zweidimensionalen Matrix bezeichnet der erste Index die Zeile, der zweite die Spalte, z.B.

m_{11} m_{12} m_{13}

m_{21} m_{22} m_{23}

m_{11} etc bezeichnet man als Einträge oder Komponenten der Matrix.

Vektor:

Ein Vektor ist ein Spezialfall einer Matrix, die entweder nur eine Zeile und beliebig viele Spalten (Zeilenvektor) oder nur eine Spalte und beliebig viele Zeilen (Spaltenvektor) hat.

Variable:

Ein Bezeichner für einen Inhalt, z.B. für eine Zahl, einen Vektor, oder eine Zeichenfolge. In Matlab bestehen Variablennamen aus Buchstaben, Ziffern und Unterstrichen, müssen aber immer mit einem Buchstaben anfangen.

Befehl:

Ein Befehl ist der Name eines Programms. Matlab bringt von sich aus eine grosse Menge an Befehlen mit, die man verwenden kann. Hinter jedem davon steckt ein Programm, viele kann man sich als m-Files ansehen (aber nicht alle - Matlab möchte sich nicht zu sehr in die Karten schauen lassen, um weiter verkauft zu werden). Wenn man selber m-Files schreibt, kann man diese anschließend genau wie die fest eingebauten Befehle in Matlab benutzen. (Aber vorsicht: die Befehle für selbst geschriebene Programme können nur in dem Verzeichnis aufgerufen werden, in dem sie auch abgespeichert sind.)

Syntax:

Die genau festgelegte Schreibweise, wie man Matlab dazu bekommt einen Befehl auszuführen, bezeichnet man als Syntax. Wenn man sich vertut oder vertippt und die Syntax nicht eingehalten wird (z.B. eine geöffnete Klammer nicht wieder geschlossen wird), gibt es einen Syntaxfehler, den Matlab durch eine rotgeschriebene Fehlermeldung anzeigt.

Kommandozeile:

Mit der Kommandozeile steuert man in Matlab die Ausführung von Befehlen. Wenn im grossen Eingabefenster das Zeichen >> zu sehen ist und dahinter der Cursor blinkt, ist Matlab eingabebereit. Während ein Programm ausgeführt wird, steht unten "busy" und es ist kein Cursor zu sehen.

Algorithmus:

Ein Algorithmus ist eine lineare Abfolge von Vorgängen, die nacheinander erfolgen. Ein Algorithmus ist zunächst unabhängig von der Programmiersprache, er kann in verschiedenen Sprachen in Programme umgesetzt (implementiert) werden.

Implementierung:

Algorithmen werden implementiert, indem sie in Abfolgen von Befehlen einer Programmiersprache umgesetzt werden. Das Endprodukt der Implementierung ist das fertige Programm.

Programm:

Die Umsetzung eines Algorithmus in eine Abfolge von Befehlen einer Programmiersprache. Der Aufruf des Programmnamens bewirkt, dass das Programm ausgeführt wird.

Workspace:

Der Workspace umfasst alle von der Kommandozeile aus zugreifbaren Variablen. Diese kann man sich im Fenster links oben ansehen. Damit man Daten verarbeiten kann, muss man sie zuerst in den Workspace bekommen, also z.B. aus einer Datei einladen oder eintippen.

Script:

Ein Script ist ein Programm ohne Eingabe- und Ausgabeparameter. Die Ausführung eines Scripts entspricht exakt einer Abfolge von Befehlen, die man in die Programmzeile des Workspace hintereinander eintippt und ausführen lässt. Der einzige Unterschied ist, dass man diese Abfolge bequemer durch Eintippen eines einzigen Befehls (des Scriptnamens) wiederholen kann. Entsprechend sind alle im Skript enthaltenen Variablen auch Variablen des Workspace und umgekehrt.

Kapselung:

Häufig möchte man nicht, dass die Variablen des Workspace durch eine Programmausführung verändert werden. In diesem Fall nutzt man die sogenannte Kapselung in einer Funktion, die ein vom Workspace abgekoppeltes Einzelleben mit eigenen Variablen führt.

Ein- und Ausgabeparameter:

Durch Ein- und Ausgabeparameter können Daten zwischen dem Workspace und Funktionen ausgetauscht werden. Eingabeparameter übertragen Werte aus dem Workspace in die Funktion, Ausgabeparameter von der Funktion in den Workspace.

Funktion:

Im Gegensatz zu einem Skript sind Variablen des Workspace nicht automatisch innerhalb einer Funktion bekannt und werden auch nicht automatisch durch eine Funktion verändert. Die Datenübertragung zwischen Workspace und Funktion erfordert Ein- und Ausgabeparameter. Nach Abarbeitung der Funktion sind nur diejenigen in einer Funktion verwendeten Variablen im Workspace vorhanden die als Ausgabeparameter übertragen wurden.

Die Syntax des Funktionskopfes lautet:

function Ausgabe=Funktionsname(Eingabe)

Der Aufruf (von der Kommandozeile oder aus einem Skript oder einer anderen Funktion heraus) lautet:

Ausgabe=Funktionsname(Eingabe)

Datenfluss:

Die Übertragung von Daten zwischen Funktionen und Workspace, sowie die Benutzung von Dateien und Benutzerabfragen kann bei komplexen Problemen recht kompliziert werden. In solchen Fällen sollte der Datenfluss zwischen den beteiligten Funktionen, Datenquellen (z.B. Benutzereingaben, Dateien) und Datensinken (z.B. Bildschirmausgaben, Dateien) sorgfältig geplant werden, um versehentlichen Datenverlust oder Programmfehler zu vermeiden. Eine grafische Darstellungsmöglichkeit des Datenflusses sind Datenflussdiagramme:

<http://de.wikipedia.org/wiki/Datenflussdiagramm>

Kontrollfluss:

Die zeitliche Reihenfolge der Abarbeitung von Befehlen eines Programms wird als Kontrollfluss

bezeichnet. Diese entspricht in Matlab normalerweise der Abfolge der Befehle im Programmcode, wird aber durch Kontrollstrukturen wie Fallunterscheidungen und Schleifen beeinflusst.

Eine gute Möglichkeit, den Kontrollfluss eines Programmes zu planen und darzustellen sind Flussdiagramme: <http://de.wikipedia.org/wiki/Programmablaufplan>

Fallunterscheidung:

Eine Fallunterscheidung besteht aus einer Bedingung und einer Abfolge von Befehlen. Die Bedingung muss so formuliert sein, dass sie entweder wahr oder falsch ist. Wenn sie wahr ist, wird der Befehlsteil einmal ausgeführt, sonst wird er im Programmablauf übersprungen. Matlab bietet als Möglichkeiten für Fallunterscheidungen "if - elseif - else" und "switch".

Schleife:

Eine Schleife besteht ebenfalls aus einer Bedingung und einer Abfolge von Befehlen, der Unterschied zur Fallunterscheidung ist dass die Befehle in Abhängigkeit von der Bedingung mehrfach ausgeführt werden können.

In einer Zählschleife, (meist als for-Schleife implementiert) wird als Bedingung eine Variable in gleichen Schritten von einem Startwert hoch- oder runtergezählt, bis ein Endwert erreicht wird.

In der allgemeineren Form einer bedingten Schleife (meist als while-Schleife implementiert) werden die Befehle solange immer wieder ausgeführt, bis die angegebene Bedingung nicht mehr erfüllt ist.

Datentypen:

Normalerweise wird Matlab hauptsächlich dazu benutzt, Zahlen zu verarbeiten, die von Matlab grundsätzlich als Fließkommazahlen (double) dargestellt werden. Es gibt aber auch Daten, für die es keinen Sinn macht, gebrochene Werte anzunehmen, und die man deshalb als ganze Zahlen (int) darstellt. Wahrheitswerte, die nur entweder wahr (1) oder falsch (0) sein können, heißen in Matlab logical. Wenn Text verarbeitet werden soll, werden dafür Buchstaben (char) als Datentyp verwendet. Außerdem werden wir im Kurs Strukturen als Datentypen kennenlernen.