

Einführung in die Informatik I

Kapitel I.8: Funktionen

Prof. Dr.-Ing. Marcin Grzegorzek
Juniorprofessur für Mustererkennung im Institut für Bildinformatik
Department Elektrotechnik und Informatik
Fakultät IV der Universität Siegen

28.11.2012

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

Funktionen

- Funktionen (auch Prozeduren, Unterprogramme) sind ein weiteres programmiersprachliches Hilfsmittel, um den Programmablauf zu steuern.
- Mit Hilfe von Funktionen können Rechenvorschriften auf eine wiederverwendbare Art verfügbar gemacht werden.
- Eine solche Vorschrift **f** berechnet **m** Ausgaben (Ergebnisse) auf der Grundlage von **n** Eingaben:

$$\mathbf{f} : (\mathbf{x1}, \mathbf{x2}, \dots, \mathbf{xn}) \longrightarrow (\mathbf{y1}, \mathbf{y2}, \dots, \mathbf{ym})$$

- Funktionen werden in Bibliotheken gesammelt.

Funktionen

- Die MATLAB-Funktionsbibliothek stellt bereits eine Fülle von Funktionen bereit:

<code>X = sin(y)</code>	<code>% 1 Eingabe , 1 Ausgabe</code>
<code>Rest = rem(20, 3)</code>	<code>% 2 Eingaben , 1 Ausgabe</code>
<code>plot(1, 1)</code>	<code>% 2 Eingaben , 0 Ausgaben</code>
<code>[Z, N] = rat(x)</code>	<code>% 1 Eingabe , 2 Ausgaben</code>

Wiederholung über Skripte

- Wiederverwendung einer Berechnungsvorschrift mit den bisher bekannten Mitteln (Skripte):

Skript A1

```
% Skript zum wiederholten  
% Aufruf von Skript A2  
  
x=0;  
  
while x<=1  
    SkriptA2;  
    disp(y);  
    x=x+0.1;  
end;
```

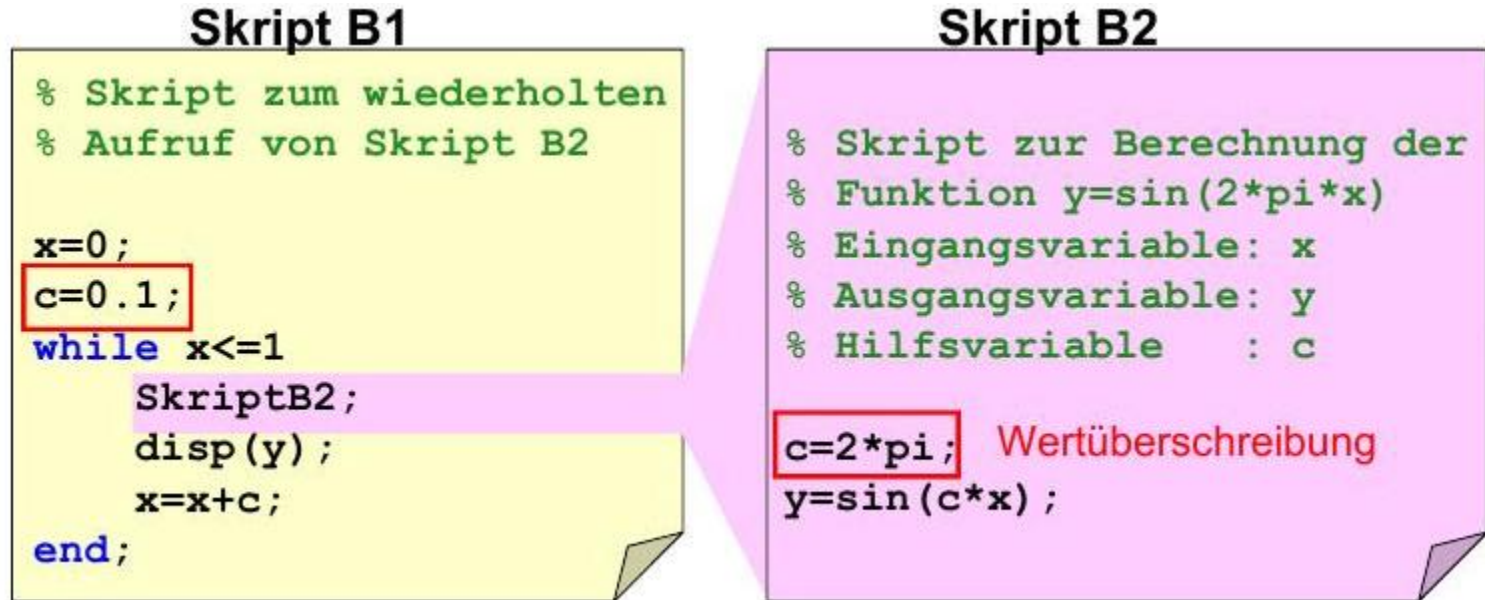
Skript A2

```
% Skript zur Berechnung  
% der Funktion  $y=\sin(2\pi x)$   
% Eingangsvariable: x  
% Ausgangsvariable: y  
  
y=sin(2*pi*x);
```

- Der Vorteil: auch komplexe Berechnungen können so mehrfach wiederverwendet werden.

Probleme mit Skripten

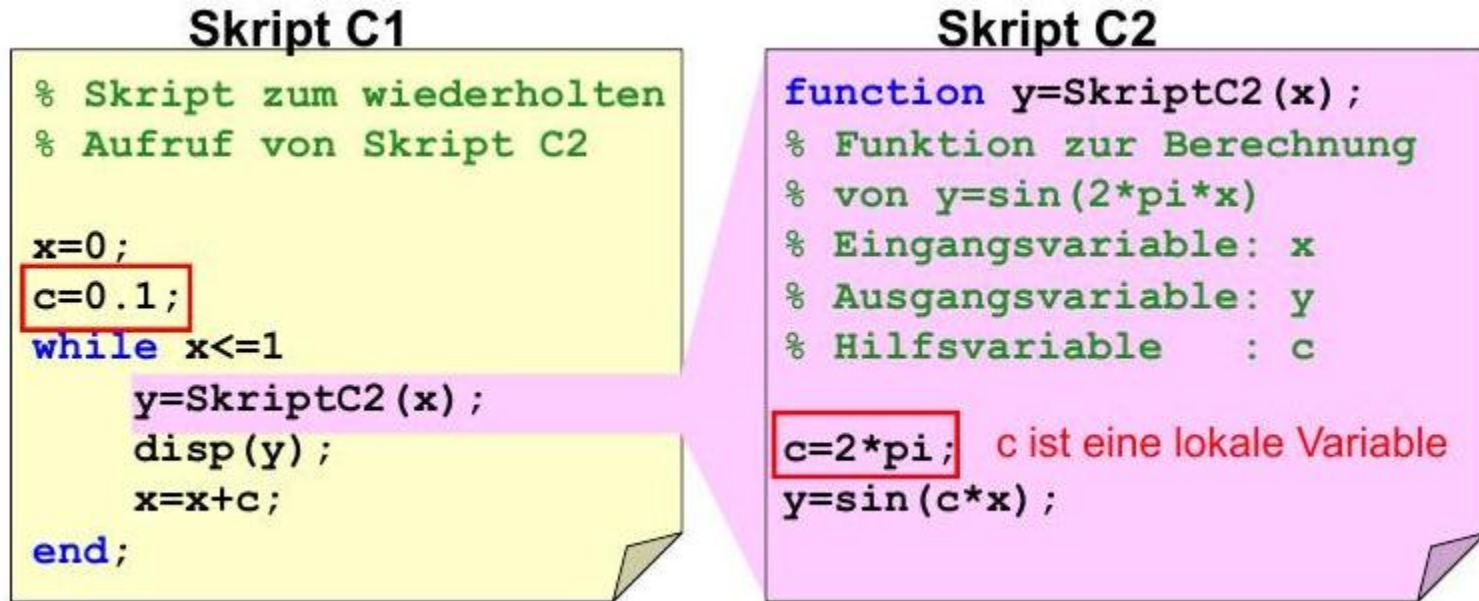
- Allerdings tritt bei dieser Methode ein schwerwiegendes Problem auf:



- Das Konzept produziert Fehler, wenn aufrufendes und aufgerufenes Skript gleiche Variablen verwenden.
- In der Praxis ist dieses Vorgehen daher nicht praktikabel.

Funktionen lösen das Problem

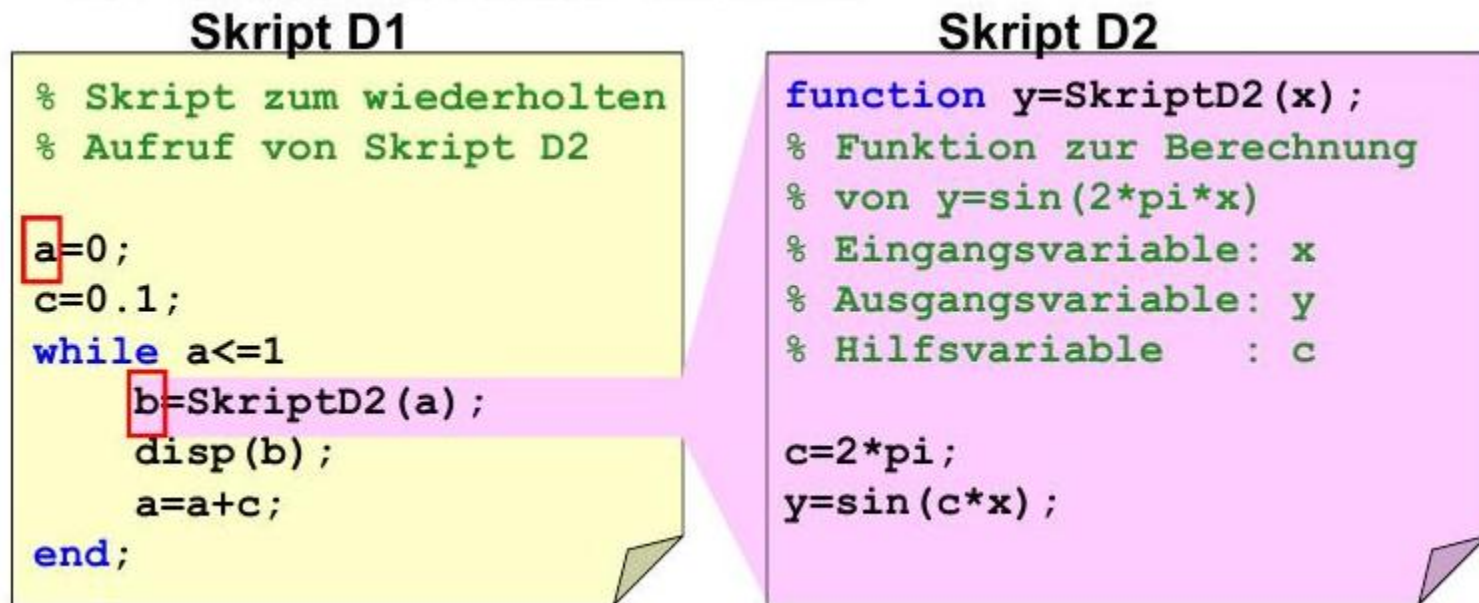
- Eine Funktion versteckt die in ihr eingeführten Variablen vor dem Zugriff aus dem aufrufenden Skript:



- Die in einer Funktion eingeführten Variablen heißen lokale Variablen. Für diese wird neuer Speicherplatz angelegt.

Übergabeparameter von Funktionen

- Die Übergabevariablen und Rückgabevariablen einer Funktion müssen nicht dieselben Namen haben wie im Aufruf.
- Sie sind ebenfalls lokale Variablen:



- Beim Aufruf werden die Variablenwerte aus dem Aufruf für die Parameter der Funktion eingesetzt. Hier $a \rightarrow x$
- Ebenso bei der Rückgabe des Ergebnisses. Hier $v \rightarrow b$

Nachträgliche Erweiterung

- Ändert man eine Funktionsimplementierung, so ändert sich ihr Verhalten überall auf die gleiche Weise, d.h. der Code der Anwendung muss nicht verändert werden.

Übergabe-Parameter

- Funktionen können in MATLAB n Variablen als Eingabewert haben und natürlich auch m Werte zurückgeben.
- Grundsätzlich muss für jeden Eingabeparameter im Aufruf auch ein Wert bereitgestellt werden. Sonst gibt es eine Fehlermeldung.
- Die Fehlermeldung lautet:

??? Input argument <variable> is undefined.

- ❖ Der Fehler tritt erst dann auf, wenn die entsprechende Variable im Funktionsrumpf gebraucht wird.

Funktion mit mehreren Ergebnissen

- Funktionen mit mehreren Ergebnissen haben die Form:

```
function [Resultat1, Resultat2, ... ]  
    = Funktionsname (Parameter1, Parameter2, ...)  
Funktionsrumpf  
  
end
```

Texte in MATLAB

- Texte werden in der Programmierung auch als Zeichenketten oder Strings bezeichnet.
- In MATLAB stehen Texte in Hochkommata. Z.B.:
 - `'Geben Sie eine Zahl ein: '`
 - `'Falsche Eingabe '`
 - `'Versuchen Sie noch einmal '`
 - ...
- Strings können wie Gleitkommazahlen in Variablen gespeichert werden. Z.B.:
 - `Text = 'Einführung in die Informatik I '`
 - `Alex = 'Wie alt ist Alex '`

input Befehl

- Mit dem input-Befehl können wir in MATLAB die Werte INTERAKTIV eingeben und definieren.
- Die Struktur des input-Befehles ist folgendes:

```
<variable> = input('Beliebiges Text')
```

Z.B.:

```
Alter = input('Wie alt sind Sie ?')
```

In Workspace wird das Text angezeigt und das Programm wartet solange, dass Sie einen Wert dem eingeben, und nach der Eingabe läuft das Programm normal weiter.

- ❖ Hier ist zu erläutern, dass die Texte werden OHNE die Hochkommata in dem Workspace angezeigt.

disp Befehl

- Mit dem disp-Befehl können wir ein beliebiges Text in Workspace erscheinen lassen. Genauer gesagt, wenn das Programm den Befehl durchführt, wird das Text auf dem Workspace angezeigt.
- Damit können sowohl Texte als auch Variablen in dem Workspace erscheinen lassen.
- Die Struktur des disp-Befehles ist folgendes:

disp ('Belibiges Text ')

oder auch

disp (<variable>)

- ❖ Hier ist auch zu erläutern, dass die Texte werden OHNE die Hochkommata in dem Workspace angezeigt.

Informationen

Quelle :

- [1] Vorlesungsskript Einführung in die Informatik I, Prof. Reichhardt, Universität Siegen, 2009.
- [2] A Guide to MATLAB for Beginners and Experienced Users, Brian R. Hunt, Ronald L. Lipsman, Jonathan M. Rosenberg, Cambridge University Press, 2001.
- [3] An Introduction to Programming and Numerical Methods in MATLAB, S.R. Otto and J.P. Denier, Springer, 2005.

Personen:

- Prof. Dr. Marcin Grzegorzek
 - Address: Research Group for Pattern Recognition Department ETI, University of Siegen Hoelderlinstr. 3, H-F 016, D-57076 Siegen.
- Dr. -Ing. Andreas Hoffmann
 - Address: Research Group for Pattern Recognition Department ETI, University of Siegen, Hoelderlinstr. 3, H-B 8405, D-57076 Siegen.