

---

# MATLAB – Funktionen

## Funktionszeiger

### 1. Zeiger

---

Zeiger (in der *MATLAB*-Notation `@<zeigername>`) sind Variablen, die als Inhalt (Wert) Adressen (z.B. von anderen Variablen oder von Funktionen) haben. Sie werden verwendet, um den indirekten Zugriff auf Variablen oder Funktionen zu ermöglichen. Die Verwendung von Zeigern kommt in den Sprachen C, C++ und weniger in C# eine besondere Bedeutung bei der Speicherverwaltung zu. Da die Zeigertechnik erhebliche Sicherheitsprobleme aufwirft, wird sie in modernen objektorientierten Sprachen wie C# oder Java zurückgedrängt. In *MATLAB* besteht im Zusammenhang mit parametrisierten Funktionsaufrufen die Notwendigkeit, Zeiger auf Funktionen zu deklarieren und als Funktionsparameter zu verwenden. Diese Funktionszeiger werden auch als *handles* bezeichnet.

### 2. Deklaration von Zeigern

---

Die Deklaration von Zeigern erfolgt mit dem Operatorsymbol `@`, so stellt z.B.

```
@fcn
```

einen Zeiger auf die Funktion mit dem Funktionskopf

```
function fcn (Parameterliste)
```

dar. Ein Aufruf

---

```
>> fdemo (@fcn)
```

ruft nun nicht die Funktion **fdemo** mit einem Wert auf, sondern mit einem Ausdruck, der von der Funktion syntaktisch und semantisch ausgewertet werden kann.

### 3. Wertetabelle der Sinusfunktion

---

Wir wollen mit der Funktion **feval** die Wertetabelle der Sinusfunktion berechnen. Dazu müssen wir der Funktion **feval** einen Funktionszeiger als Parameter übergeben. Die Formatierung der Parameterübergabe wird durch die mögliche Überladung

```
feval (<Funktionszeiger>, <Vektor>)1
```

der Funktion **feval** erzwungen. Der erste Parameter ist ein Zeiger auf den auszuwertenden Funktionsterm, der zweite Parameter der Wertebereich, auf dem diese Funktion ausgewertet werden soll. Wir konstruieren auf der Kommandoebene einen Vektor **x**, der den Wertebereich einer Wertetabelle der *MATLAB* Funktion **sin(x)** darstellt und rufen von der Kommandoebene mit

```
>> x = (0:4)
>> feval (@sin, x)
```

die Funktion **feval** mit dem Verweis auf die Sinusfunktion auf.<sup>2 3</sup> Die Übergabe eines Funkti-

---

<sup>1</sup> Die Metanotation **<ausdruck>** bedeutet, daß der Term **ausdruck** durch einen Term des terminalen Vokabulars der *MATLAB*-Grammatik zu ersetzen ist.

<sup>2</sup> Andere Funktionsaufrufe, wie z.B. der direkte Aufruf der Funktion **sin(x)**

```
>> feval (sin(x), x)
```

sind hier unzulässig und führen zur Fehlermeldung

```
>> feval(sin(x),x)
```

onszeiger ist auch dann erforderlich, wenn eine selbstdefinierte Funktion einen Parameter übernimmt, der nachfolgend als Parameter an eine Funktion weitergegeben werden soll, die an der betreffenden Stelle einen Parameter vom Typ *Zeiger* erwartet. Soll z.B. die Funktion **wtab** eine Wertetabelle mit dem internen Aufruf der Funktion **feval** berechnen, dann muß auch der Funktionsparameter der Funktion **wtab** (*implizit* oder *explizit*) als vom Typ *Zeiger* deklariert werden.

### Listing 1

```
function wert = wtab(funk)

    x = (0:4)
    wert = feval (funk, x)
end
```

Die Funktion **wtab** kann von der Kommandoebene aufgerufen werden. Der entsprechende Funktionsaufruf zur Berechnung von Werten der Funktion **sin(x)** lautet

```
>> wtab (@sin)
```

```
??? Error using ==> feval
Argument must contain a string or function_handle.
```

<sup>3</sup> Alternativ ist auch der Funktionsaufruf

```
>> feval ('sin')
```

möglich. Bei diesem Funktionsaufruf erfolgt eine implizite Typkonvertierung von **'sin'** vom Typ *'string'* zu **@sin**, d.h. in den Typ *'Zeiger'*,

---

Die Ausgabe der Funktionswerte erfolgt dann auf der Kommandoebene.

#### 4. Berechnung der Parabel $f(x) = a x^2 + b x + c$

---

Funktionszeiger können von Funktionen an andere Funktionen weitergeben werden. In nachfolgendem Beispiel soll die selbstdefinierte Funktion **Main()** die selbstdefinierte Funktion **par()** aufrufen. Die Funktion **par()** berechnet die Funktionswerte einer Parabel mit den Koeffizienten **a**, **b** und **c** auf dem durch den Vektor (*hier*) **x = [0 1 2 3 4 5]** definierten Wertebereich.. Der Funktionsaufruf

```
y = par()
```

ruft die parameterlose Funktion **par()** auf. Soll diese Berechnung mit der Funktion **feval** ausgeführt werden, muß die aufrufende Funktion den Zeiger **@par** übergeben, der entsprechende Funktionsaufruf ist

```
w = wtab (@par)
```

Nachfolgendes Programm zeigt den hierarchischen Funktionsaufruf

```
>>Main (wtab (feval(@par())))
```

mit dem Funktionszeiger **z = @par.**

**Listing 2**

```
function Main()

    % Variablen zurücksetzen
    % Konsole löschen

    clear all
    clc

    % Funktionszeiger deklarieren

    z = @par

    % Funktionsaufruf mit Zeiger

    w = wtab(z)
end

function f = par(x)

    % Deklaration des Wertebereichs

    % Deklaration /Initialisierung der
    % Koeffizienten

    a = 1
    b = 0
    c = 1

    % Rückgabe des Funktionswerts

    f = (a.*(x.*x) + b.*x + c);
end

function wert = wtab(fHnd)

    x = (0:5)
    wert = feval (fHnd, x);
end
```

Die Funktion **Main()** ruft die Funktion **wtab** mit dem Funktionszeiger **@par** als Parameter auf. Die Funktion **wtab** übernimmt den Funktionszeiger **@par** als Wert des Parameters **fHnd**. **fHnd** wird nun in den Typ *Zeiger* (implizit) konvertiert und wertet den Funktionsterm, auf den der Zeiger **@par** weist, aus. Die Ausgabe erfolgt in der Funktion **Main()**.