

## Matlab Vektoren 1

### Zusammenfassung von brunowe, exklusiv für das goMatlab- Forum

#### **Das Erstellen von Vektoren:**

##### **Wie lassen sich Zeilen-/ bzw. Spaltenvektoren erzeugen?**

###### a) Zeilenvektoren

U=[1 2 3]	% Einzelne Elemente durch Space getrennt
V=[1,2,3]	% Einzelne Elemente durch Komma getrennt
A=[1:2:9]	% Mit Colon Operator: Startwert, Schrittweite (optional) Standard= 1, Endwert
A=1:2:9	% Auf [ ] kann hier verzichtet werden
B=linspace(pi/10, pi, 11)	% linspace ist immer vorteilhaft wenn Anzahl der Elemente angegeben werden soll. (Startwert, Endwert, Anzahl der Elemente (optional, Standard= 100))
C=logspace(0,3,50)	% logarithmische Abstände zw. den Werten, Startwert:10^0, Endwert: 10^3, Anzahl der Elemente (optional, Standard= 50)
A = B'	% Durch Umwandlung (transponieren) eines Spaltenvektors (hier B)
A=ones(1,5)	% Durch Verwendung einer vorgefertigten Matlab Funktion (z.B. ones, zeros, rand, randn, magic, ...) und Begrenzung auf eine Zeile

###### b) Spaltenvektoren

W=[1;2;3]	% Trennzeichen Semikolon
W=[1 2]	% Durch Zeilenumbruch
A = B'	% Durch Umwandlung (transponieren) eines Zeilenvektors (hier B)
A=ones(5,1)	% Durch Verwendung einer vorgefertigten Matlab Funktion (z.B. ones, zeros, rand, randn....) und Begrenzung auf eine Spalte
B= C(:)	% Durch Zuweisung mittels linearer Indizierung (C kann hier Spalten-, Zeilenvektor oder eine Matrix sein)

Des Weiteren können Vektoren auch als Ergebnis einer Rechenoperation, einer Matrixumwandlung mit reshape, durch Zusammenfügen aus anderen Vektoren (hier A, B) (z.B.: C=[A B]), durch Ausschneiden aus bestehenden Vektoren oder Matrizen ( B= C(1,:) ) entstehen, oder eingelesen werden (z.B. von einem Excelsheet, via RS232 usw...)

#### **Transponieren**

X=U'	<p>% Unter Transponieren versteht man das Umwandeln von Spalten in Zeilenvektoren, bzw. umgekehrt. Dies lässt sich durch ein Hochkommata erreichen.</p> <p><b>Achtung beim Transponieren von Vektoren mit komplexen Elementen!</b></p> <p>Diese werden durch das Hochkommata zu konjugiert komplexen Elementen, wird das nicht gewünscht, verwende: .'</p>
------	--

## Überprüfen auf Gleichheit

```
U==V %ans = 1 1 1 hier werden alle Elemente einzeln auf Gleichheit mit dem  
entsprechenden Element des 2.ten Vektors verglichen. Die logische "1"  
kennzeichnet hier die Gleichheit der einzelnen Elemente.  
all(U==V) %ans= 1 Das Schlüsselwort "all" gibt, wenn alle Elemente mit den  
entsprechenden Werten des 2ten Vektors identisch sind, eine logische "1" zurück.  
%Alternativ: isequal(U,V)
```

**Achtung:** Vektoren müssen die gleiche Orientierung haben (ggf. vorher Transponieren) und die gleiche Anzahl an Elementen

% isequal akzeptiert auch ungleich lange Vektoren, liefert dann aber stets ans=0, also Vektoren sind ungleich

```
any(U==V) %ans = 1 Gibt, wenn mindestens ein Element des Vektors 1 mit dem  
entsprechendem Element des Vektors 2 identisch ist, eine logische "1" zurück.
```

**Tip:** soll nur überprüft werden, ob die Elemente von Vektoren gleich sind (unabhängig von ihrer Position im Vektor), so sortiere die Vektoren vor der Überprüfung mit sort.

**Tip:** sollen ungleich lange Vektoren überprüft werden, so begrenze die Überprüfung auf die Länge des kürzeren Vektors. Also z.B.: `U==V(1:length(U))` % U sei hier der kürzere Vektor

Auf diese Weise lassen sich auch bestimmte Ausschnitte von Vektoren überprüfen, z.B.:

```
[A(1:3) A(5:7)]==B(1:6)
```

**Tip:** Beim Test auf Gleichheit von Vektoren/ Matrizen lassen sich auch vorteilhaft die Befehle:

ismember, setdiff und unique verwenden. S.h. Matlab- Hilfe. Evtl. bespreche ich diese Befehle auch in einem folgenden Tutorial.

## Informationen über Vektoren abfragen

```
whos('U') % Wenn Klammern verwendet werden, setze Namen des Vektors in Hochkommata!  
% ans: Name size Bytes class  
U 1x3 24 double
```

```
% Alternativen: whos U  
% whos 'U'
```

%Die Angabe size: 1x3 bedeutet: der Vektor besteht aus 1ner Zeile und 3 Spalten, U ist also ein Zeilenvektor

```
numel(U) %Anzahl der Elemente des Vektors U (bei Matrizen die Gesamtzahl über alle  
Dimensionen)
```

```
length(U) %Höchste Anzahl der Elemente in einer Dimension, bei Vektoren gleich  
numel(U)
```

```
size(X) % ans: 3 1 bedeutet hier: 3 Zeilen, 1ne Spalte
```