Kapitel 10

Zeichenketten

10.1 Grundlagen

Der MATLAB-Datentyp `char` dient zur Speicherung von ASCII-Zeichen. Dies kann auch in ein- bzw. mehrdimensionalen Feldern geschehen. Es besteht jedoch die Einschränkung, dass alle Zeilen die gleiche Anzahl von Zeichen enthalten müssen, ähnlich wie alle Zeilen einer Matrix die gleiche Anzahl von Elementen beinhalten müssen.

MATLAB speichert Zeichenketten, auch Strings genannt, als Felder von ASCII-Werten. Diese liegen z.B. zwischen `48` und `57` für die Zahlen 0 bis 9, zwischen `65` und `90` für die Großbuchstaben und zwischen `97` und `122` für die Kleinbuchstaben. Neben anderen Sonderzeichen hat der horizontale Tabulatur `HT` den Wert `9`, der Zeilenumbruch `LF` den Wert `10`, und das Leerzeichen `SP` den Wert `32`.

Die ASCII-Werte `A` und die Zeichenketten `S` können mit den Befehlen `S=char(A)` bzw. `A=double(S)` ineinander umgewandelt werden.

Die Erzeugung von Strings erfolgt mit `s1='sin'` bzw. mit `s2=char('(x)')`. Ein horizontales Aneinanderfügen erfolgt mit `[s1,s2]` bzw. mit `strcat(s1,s2)`, eine Anordnung in mehreren Zeilen kann in Prinzip wie bei Vektoren mit `[s1;s2]` erfolgen, wenn beide Strings gleich lang sind. Besser ist jedoch die Verwendung von `char(s1,s2)` oder `strvcat(s1,s2`, da hier zu kurze Zeilen am Zeilenende durch Leerzeichen aufgefüllt werden.

Ist das Auffüllen mit Leerzeichen nicht erwünscht, muss auf Objekte des Typs `cell` zurückgreifen.

```
z = {'Erste Zeile', ...
     'Zweite Zeile'}
```

Mit `z{1}` bzw. `z{2}` kann man dann wieder auf die einzelnen Elemente der Zelle zugreifen. Auch hier gibt es Funktionen zur Umwandlung, `s=char(z)` von der Zelle zum String, bzw. `z=cellstr(s)`.

Eine Zusammenstellung interessanter Umwandlungsroutinen für Strings:

`s = num2str(d)`	Zahlen in Strings
`s = num2str(d,n)`	Zahlen in Strings; n Stellen
`s = int2str(d)`	Integer in Strings (runden)
`s = mat2str(m)`	Matrizen in Strings mit []
`s = mat2str(m,n)`	Matrizen in Strings; n Stellen
`d = str2num(s)`	String in Zahlen
	Leeres Array `[]` falls keine Zahl
`d = str2double(s)`	String in eine double-Zahl
	`NaN` falls nicht möglich
`sm = str2mat2(s)`	String in Stringmatrix
	Leerzeichen erzeugt neue Zeile
`z = cellstr(s)`	Strings in Zellen
`s = char(z)`	Zellen in Stringmatrix
`A = double(s)`	Strings in ASCII Werte
`s = char(A)`	ASCII Werte in Strings

Darüber hinaus gibt es eine Reihe von Befehlen, die Strings umwandeln, in Strings suchen, Strings vergleichen usw.

`s=blanks(n)`	Erzeugt String der Länge n mit Leerzeichen
`s=deblank(s)`	Entfernt Leerzeichen am Beginn
`s=lower(s)`	Umwandlung in Kleinbuchstaben
`s=upper(s)`	Umwandlung in Großbuchstaben
`l=ischar(s)`	`TRUE` wenn String
`l=isletter(s)`	`TRUE` wenn Buchstabe
`l=strcmp(s1,s2)`	`TRUE` wenn gleich
`l=strcmpi(s1,s2)`	`TRUE` wenn gleich
	ignoriert Groß/Kleinschreibung
`l=strncmp(s1,s2,n)`	`TRUE` wenn die ersten n gleich
`l=strncmpi(s1,s2,n)`	`TRUE` wenn die ersten n gleich
	ignoriert Groß/Kleinschreibung
`i=strfind(s1,s2)`	Positionen von s2 im String s1
`i=findstr(s1,s2)`	Positionen des kürzeren im längeren
`i=strmatch(s,sm)`	Zeilen in Stringmatrix oder Zelle,
	die mit String s beginnen
`i=strmatch(s,sm,'exact')`	Zeilen in Stringmatrix oder Zelle,
	die mit String s exakt übereinstimmen