

1 Matrizen

1.1 Ohne Klammern

$$\begin{array}{ccc} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{array} \quad (1.1)$$

1.2 Runde Klammern

$$\begin{pmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{pmatrix} \quad (1.2)$$

1.3 Eckige Klammern

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix} \quad (1.3)$$

1.4 Geschweifte Klammern

$$\begin{Bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{Bmatrix} \quad (1.4)$$

1.5 Betragsstriche

$$\left| \begin{array}{ccc} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{array} \right| \quad (1.5)$$

1.6 Doppelte Betragsstriche (Norm)

$$\left\| \begin{array}{ccc} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{array} \right\| \quad (1.6)$$

2 Matlab Quellcode

2.1 Was wird benötigt

Für Listings (Quellcode-Fenster) benötigt man lediglich das Package:

```
\usepackage{listings}
```

Möchte man den Quellcode einfärben, benötigt man zusätzlich:

```
\usepackage{color}
```

2.2 Direkt in Latex

```
a = 1;  
b = 2;  
c = a + b;
```

2.3 Aus einer Datei

```
% Optimierung:
```

```
clear tabelle;  
clear werte;
```

```
untere_grenze = 0;  
genauigkeit = 0.01;  
obere_grenze = 100;
```

```
anzahl_schritte = 10;
```

```
anzahl_durchgaenge = log((obere_grenze - untere_grenze  
)/anzahl_schritte/genauigkeit)/log(anzahl_schritte  
/2);
```

```
anzahl_durchgaenge = ceil(anzahl_durchgaenge);
```

```
prozent = 0;
```

```
for anzahl = 1:anzahl_durchgaenge
```

```
    intervall = (obere_grenze - untere_grenze)/  
        anzahl_schritte;
```

```
    werte = untere_grenze:intervall:obere_grenze;
```

```

    for i = 1:anzahl_schritte+1
        korrekturfaktor = werte(i);
        sim('testbench_v01_1');
        tabelle(i) = min(abs(simout1(2000:end)));

        clc;
        prozent = prozent + 100/(anzahl_durchgaenge+1)
            /anzahl_schritte;
        fprintf('%f_%%',prozent);
    end

    [C,I] = min(tabelle);
    obere_grenze = werte(I) + intervall;
    untere_grenze = werte(I) - intervall;

end

korrekturfaktor = werte(I)

```

2.4 Quellcode im Fließtext

Quellcode kann auch mitten im Fließtext eingebettet werden, wie z. B. hier:
print "hello world".

2.5 Mit Zeilennummern

```

1 a = 1;
2 b = 2;
3 c = a + b;

```

2.6 Mit Kasten

```

1 a = 1;
2 b = 2.5;
3 c = a + b;
4 ceil(c);

```

2.7 Mit Farben

```

1 a = 1;
2 b = 2.5;
3 c = a + b;
4 ceil(c)

```

2.8 Mit anderen Zeilennummern

```
23 a = 1;  
24 b = 2.5;  
25 c = a + b;  
26 ceil(c)
```

2.9 Kleinere Schriftart

```
23 a = 1;  
24 b = 2.5;  
25 c = a + b;  
26 ceil(c)
```

2.10 Nochmal alles zusammen

```
59 % Optimierung:  
60  
61 clear tabelle;  
62 clear werte;  
63  
64 untere_grenze = 0;  
65 genauigkeit = 0.01;  
66 obere_grenze = 100;  
67  
68 anzahl_schritte = 10;  
69  
70 anzahl_durchgaenge = log((obere_grenze - untere_grenze)/  
71   anzahl_schritte/genauigkeit)/log(anzahl_schritte/2);  
72 anzahl_durchgaenge = ceil(anzahl_durchgaenge);  
73  
74 prozent = 0;  
75  
76 for anzahl = 1:anzahl_durchgaenge  
77     intervall = (obere_grenze - untere_grenze)/anzahl_schritte;  
78  
79     werte = untere_grenze:intervall:obere_grenze;  
80  
81     for i = 1:anzahl_schritte+1  
82         korrekturfaktor = werte(i);  
83         sim('testbench_v01_1');  
84         tabelle(i) = min(abs(simout1(2000:end)));  
85  
86         clc;  
87         prozent = prozent + 100/(anzahl_durchgaenge+1)/anzahl_schritte  
88         ;  
89         fprintf('%f%%',prozent);  
90     end  
91  
92     [C,I] = min(tabelle);  
93     obere_grenze = werte(I) + intervall;  
94     untere_grenze = werte(I) - intervall;  
95 end
```

```
96 |
97 | korrekturfaktor = werte(I)
```

2.11 Und nochmal Quellcode im Fließtext

Jetzt sieht auch der Quellcode im Fließtext anders aus, z.B. hier: `print "hello world".`

2.12 Quellen

- Allgemein: http://wiki.infostudium.de/wiki/Quelltext_in_LaTeX
- Andere Zeilennummern: http://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings