

Matrizen

Matlab> A=[1 3 5 7; 2 4 6 8] => $\begin{pmatrix} 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \end{pmatrix}$

Matlab> B = ones(2,4) => $\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$

Matlab> C = zeros(1,3) $\begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$

Matlab> D = 1:5 => $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \end{pmatrix}$

Matlab> E = 5:-0.5:4 => $\begin{pmatrix} 5 & 4.5 & 4 \end{pmatrix}$

Matlab> F = diag(1:3) => $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}$

Matlab> F = 5*ones(3,3) => $\begin{pmatrix} 5 & 5 & 5 \\ 5 & 5 & 5 \\ 5 & 5 & 5 \end{pmatrix}$

Matlab> A(2,3) = 5 => schreibt in die 2te Zeile in die 3te Spalte eine 5

Matlab> A(1:2,1:2) = ones(2,2) => 1/1, 1/2, 2/1, 2/2 werden mit 1 gefüllt

Matlab> A(1:end,[2 4]) = 9 => füllt Spalte 2 und 4 mit 9ern

Spalte löschen: Matlab> X(:,2) = [] => löscht zweite „column“

Matrix, in der jedes Element den Wert 10 hat: >> A = ones(5,5)*10;

Vektoren

Matlab> A = 1:3 => $\begin{pmatrix} 1 & 2 & 3 \end{pmatrix}$ => Matlab> B = A' => $\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$

Matlab> C = 1:5; => $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \end{pmatrix}$ => Matlab> C(3) = 100
=> $\begin{pmatrix} 1 & 2 & 100 & 4 & 5 \end{pmatrix}$

```
Matlab> A = (1:4).^2=> 4 9 16
```

```
Matlab> A = 100:-7:50=> 100 93 86 79 72 65 58 51
```

Um einen Vektor zurück in Matrixform zu verwandeln, wird reshape verwendet.

```
Zahlen kleiner 0 löschen: >> M = [-.2 1.0 1.5 3 -1 4.2 3.14]; >> M>=0 >> ans = 0 1  
1 1 0 1 1
```

Vergleichsoperatoren

<,> kleiner, größer

<=,>= kleinergleich, größergleich

== gleich

~= ungleich

Plotten

```
Matlab> t = 0:0.01:2*pi;=> Matlab> y = sin(t);=> Matlab> plot(t,y);
```

Liniefarbe: plot(x,y,'FARBE'); // 'c', 'm', 'y', 'r', 'g', 'b', 'w', and 'k'

Linietyt: plot(x,y,'TYP'); // '-', '--', ':', '-.'

Liniemarke: plot(x,y,'MARKER'); // '+', 'o', '*', 'x', 's', 'd', '^', 'v', '>', '<', 'p', 'h',
'none'

Achsenbeschriftung: xlabel('...') und ylabel('...')

Achsen unsichtbar machen: axis off // mit axis on sichtbar machen

Gitternetz unsichtbar machen: grid off/ mit grid on sichtbar machen

Graphen beschriften: legend('string1','string2',...)

Titel des Plots: title('...')

Text ausgeben: text(x,y,'string') //in string steht die Ausgabe drinnen

Mehrere Graphen auf ein mal und bunt: `x = 0:pi/100:2*pi;=> y = sin(x); => y2 = sin(x-.25); => y3 = sin(x-.5); => plot(x,y,x,y2,x,y3) => legend('sin(x)','sin(x-.25)','sin(x-.5))'`

Dargestellten Wertebereichs: `axis([xmin; xmax])` bzw. `axis([xmin; xmax; ymin; ymax])`

Achsen beschränken: `xlim([xmin xmax])` // z.b. `ylim([-40 10])`;

Matlab> `X = ...=> Matlab> Y = ... => Matlab> Z = ... => Matlab> mesh(X,Y,Z);`

Visualisierung

Darstellen: `imshow(I,[])`; //Stellt Matrix I als Graustufenbild da

Invertiert darstellen: `imshow(-I,[])`;

Bild laden: `A = imread('bildname.tif');` //und am besten mit `A = double(A);`
umwandeln z.b. `chaplin = double(imread('chaplin.tif'));`

Bestimmter Bildausschnitt anzeigen: `Matlab> imshow(B(117:146,100:194),[])`

Fenster

Neues Fenster: `figure`

Beschriftung: `title (,NAME')`

Mehrer „Funktionen“ in einem Fenster plotten und überlagern: `hold on`

Beispiel: `Matlab> x= linspace(0,2*pi,100);=> Matlab> y1 = sin(x); => Matlab> y2 = cos(x); => Matlab> plot(x,y1,'k-'); => Matlab> hold on; => Matlab> plot(x,y2,'k.');`

Ausgabefenster in y kreuz x Unterbereiche aufteilen: `subplot(y; x; index)`

Beispiel: Matlab> subplot(3,2,1); => Matlab> plot(x1,y1); => Matlab> title('...');
=> Matlab> ... => Matlab> subplot(3,2,2); => Matlab> plot(x2,y2); => Matlab> ...
=> Matlab> ... => Matlab> subplot(3,2,6); => Matlab> plot(x6,y6); => Matlab>
title('...');

Oft gebrauchte Funktionen

Funktion selbst schreiben: function y = gauss1d(x, sigma); => y = FORMEL =>
und nützen der Funktion indem man z.b. y1 = gauss1d(1.5,6) macht

Mittelwert: VAR = mean(VAR2) //bei Matrix mean2

Standardabweichung: VAR = std(VAR2) //bei Matrix std2

Bild c in 8x8 Blöcke teilen+ Mittelw.: fun = @mean2;=> b = blkproc(c, [8,8], fun);

2D Filter: h = fspecial(type,parameters) //'gaussian', 'sobel', 'prewitt', 'laplacian',
'log', 'average', 'unsharp'

Bild [falten](#): c = conv2(image, FILTER); //z.b. FILTER = fspecial('average',3);

Noise erzeugen: imnoise(IMAGE,type) //z.b. image2 = imnoise(image, 'salt &
pepper', 0.05);

[Medianfilter](#): MedFilter = medfilt2(BILDNAME);

fftshift: zero frequency in die mitte, höchste frequenzen nach aussen da
anscheinend nach fft2 höchste innen, niedrigste aussen.

Diskrete Fourier Transformation: dft = fft2(Image);

Bild rotieren: rotiert = imrotate(IMAGE,angle); // z.b tmp = imrotate(I,45);

Gradienten berechnen:[X, Y] = gradient(Image);

Vektor mit Pfeilen darstellen: quiver (VektorX, VektorY);