

Befehle können direkt, nach der Eingabeaufforderung `-->` in das Kommandofenster eingegeben werden. Oder eine ganze Abfolge von Befehlen kann erst in einer Datei mit der Endung `.m` gespeichert werden, z.B. in `mytest.m` und dann mit dem Befehl `mytest` ausgeführt werden. Hierbei ist darauf zu achten, daß das Verzeichnis, in dem die Datei gespeichert wurde, ausgewählt ist.

Vektoren

<code>a=[1;2;3]</code>	Spaltenvektor $(1, 2, 3)^T$
<code>b=[4,5,6];</code>	Zeilenvektor. Semikolon am Ende: Keine Bildschirmausgabe
<code>x=0:0.1:10</code>	Zeilenvektor mit Einträgen von 0 bis 10, Schrittweite 0.1
<code>x=linspace(0,1,11)</code>	Zeilenvektor mit 11 äquidistanten Einträgen von 0 bis 1
<code>x'</code>	Transponieren, Zeilenvektor \longleftrightarrow Spaltenvektor
<code>n=0:10</code>	Zeilenvektor mit Einträgen 0 bis 10
<code>e=ones(1,5)</code>	Zeilenvektor mit 5 Einsen
<code>zeros(1,7)</code>	Zeilenvektor mit 7 Nullen
<code>x(3)</code>	Dritte Komponente des Vektors x
<code>x(3:5)</code>	Komponenten 3 bis 5
<code>x([1 3 7 2])</code>	Komponenten 1, 3, 7 und 2 des Vektors x

Rechnen mit Vektoren

<code>norm(a)</code>	euklidische oder 2-Norm des Vektors a
<code>norm(a,1)</code>	1-Norm des Vektors a
<code>norm(a,inf)</code>	Maximums- oder ∞ -Norm des Vektors a
<code>[1;2;3]'*[4;5;6]</code>	Skalarprodukt der beiden Vektoren $(1, 2, 3)$ und $(4, 5, 6)$
<code>[1;2;3]*[4;5;6]'</code>	dyadisches Produkt der beiden Vektoren $(1, 2, 3)$ und $(4, 5, 6)$
<code>y=2*x+1;</code>	$y_k = 2x_k + 1$ für alle Komponenten
<code>z=x.^2+y</code>	$x.^2 = x_k^2$ <i>punktweise</i> , d.h. komponentenweise Quadrieren
<code>w=1./(1+x.^4)</code>	$w_k = 1/(1 + x_k^4)$ <i>punktweise</i> , d.h. komponentenweise Division
<code>r=sqrt(n)</code>	Wurzelfunktion
<code>v=sin(2*pi*x)</code>	Funktionen wie <code>sin</code> , <code>cos</code> , <code>exp</code> , <code>log</code> , <code>sqrt</code> werden komponentenweise auf Vektoren angewandt
<code>log(n)</code>	ergibt <code>-Inf</code> in der ersten Komponente wegen Nulleintrag im Vektor n
<code>n>=3</code>	Komponentenweiser Vergleich. Ergibt 1 falls wahr, 0 sonst
<code>n==4</code>	<code>==</code> und nicht <code>=</code> testet auf Gleichheit
<code>n~=5</code>	Test auf Ungleichheit
<code>(n<2) (n>8)</code>	Test auf $n < 2$ oder $n > 8$
<code>(n>4) && (n<7)</code>	Test auf $n > 4$ und $n < 7$
<code>m=length(a)</code>	Länge des Vektors a
<code>s=sum(a)</code>	Summe der Komponenten $s = \sum_{i=1}^m a_i$
<code>c=cumsum(a)</code>	kummulierte Summe $c_j = \sum_{i=1}^j a_i$
<code>prod(b)</code>	Produkt der Komponenten
<code>cumprod(b)</code>	kumuliertes Produkt der Komponenten
<code>max(b)</code>	Maximale Komponente des Vektors, <code>min</code> analog

Plotten in 2d

<code>plot(x,v);</code>	Zeichnet Linienzug durch Datenpunkte (x_k, v_k)
<code>plot(x,y,'b-', x,z,'r--*');</code>	Ersten Datensatz mit blauer Linie, zweiten Datensatz rot gestrichelt mit Sternsymbol

Matrizen

<code>A=[0, 1; -1, 0]</code>	eine 2×2 -Matrix
<code>E=eye(3)</code>	3×3 -Einheitsmatrix
<code>M=ones(2,3)</code>	2×3 -Matrix mit nur Einsen
<code>R=rand(3,3)</code>	3×3 -Matrix mit Zufallszahlen zwischen 0 und 1
<code>R'</code>	transponierte Matrix, d.h. R^T
<code>size(M)</code>	Dimension der Matrix M
<code>R(:,1)</code>	erste Spalte von R
<code>R([2,1],:)</code>	zweite und erste Zeile von R

Rechnen mit Matrizen

<code>R+E</code>	Summe der beiden Matrizen
<code>R-4*E</code>	Linearkombination $R_{ij} - 4E_{ij}$
<code>A*M</code>	Matrixprodukt $AM \in \mathbb{R}^{2 \times 3}$
<code>M*A</code>	ergibt Fehlermeldung, da Matrizengrößen inkompatibel
<code>A^2</code>	Matrixprodukt $A^2 = AA$
<code>A.^2</code>	komponentenweises Quadrieren a_{ij}^2
<code>diag(R)</code>	Diagonalelemente von R
<code>det(A)</code>	Determinante von A
<code>eig(A)</code>	Eigenwerte von A
<code>[V,lambda]=eig(A)</code>	V enthält spaltenweise die Eigenvektoren, λ enthält Eigenwerte auf der Diagonalen
<code>A\[1;2]</code>	Lösung des LGS $Ax = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$
<code>inv(A)</code>	inverse Matrix

Funktionen und Kontrollstrukturen

<code>f=@(x) exp(-x.^2);</code>	selbstdefinierte Funktion $f(x) = e^{-x^2}$
<code>f(1)</code>	Funktionswert $f(1)$
<code>f(-3:3)</code>	Funktionswerte für $x = -3, -2, \dots, 3$
<code>if (x>3)</code> <i>Anweisung 1;</i> <code>else</code> <i>Anweisung 2;</i> <code>end;</code>	Führt die <i>Anweisung 1</i> aus, falls die <i>Bedingung</i> $x > 3$ wahr ist, sonst wird die <i>Anweisung 2</i> ausgeführt.
<code>for i=1:10;</code> <i>Anweisung;</i> <code>end;</code>	Die Schleife mit der <i>Anweisung</i> wird für i von 1 bis 10 durchlaufen.
<code>while (n<4);</code> <i>Anweisung;</i> <code>end;</code>	Die <i>Anweisung</i> wird solange ausgeführt, wie die <i>Bedingung</i> $n < 4$ wahr ist.
<code>help while</code>	Hilfetexte zum Befehl <code>while</code>
<code>helpwin</code>	Hilfefenster mit der Online-Hilfe

In der Online-Hilfe finden sich weitere nützliche Befehle z.B. zur Interpolation, Erzeugen, Drucken und Speichern von 3d-Grafiken, etc. Alle hier vorgestellten Befehle sind auch unter MATLAB verfügbar.

Eine Garantie für die Richtigkeit der Berechnungen kann nicht gegeben werden!