Variablen

- In MATLAB vordefinierte Variablen
  - pi = 3.14…..
  - ans = Standard-Ausgabevariable
  - inf = unendlich
  - i = Imaginärteil einer Zahl,

    i = $\sqrt{-1}$

Variablen

- Zeichenketten Beispiele:
- A='Hallo';
- B='Welt';
- C=[A,' ',B];

  upper(C)

  lower(C)

Ausgabe der ASCII Zeichen

ascii = char(reshape(32:127,24,4)')

Arbeiten mit Variablen

- Kommentare verwenden
- Mit who werden alle aktuellen Variablen aufgelistet
- Ausführliche Auflistung mit whos
- Löschen einer lokalen Variablen mit clear Variable
- Ohne Variablennamen löscht clear alle lokalen Variablen

Arithmetische Operatoren

Hilfe zu den Operatoren mit help ops

=	Zuweisung
+, -	Addition, Subtraktion
, .	Multiplikation, Elementweise Multiplikation
/, \	rechte bzw. linke Matrix-Division
^	Exponentialfunktion

Logische Operatoren

&, |	logisches Und, Oder
<, > <=, >=	kleiner; größer kleiner gleich; größer gleich
==	Gleich
~	logisches Nicht

Vektoren und Matrizen

- Die Größe einer Matrix wird mit size(A) bestimmt
  - Ausgabe: zuerst Zeilen, später Spalten
- Die Länge von Vektoren wird mit length(A) angezeigt
- Selektion von Matrix Elementen erfolgt durch Indexzugriff
  - Bsp.: A(3,4) holt das Element aus der 3.Zeile, 4.Spalte
- Die Zählung der Indizes beginnt bei 1

Vektor und Matrix Operatoren

Hilfe zu Matrix Manipulation mit help elmat

( ), [ ]	runde Klammern für Argumente, eckige Klammern für Matrizen, mit Hilfe von eckigen Klammern können auch Matrizen von Matrizen erzeugt werden
, ;	Komma als Elementetrenner in Zeilen, Strichpunkt als Zeilenumbruch (in Matrizen)
.	Elementweise Durchführung, Dezimaltrenner
‘	Hochkomma für Transposition, Kennzeichnung von Text

Vektor und Matrix Operatoren

Der **:** Operator

- Wird als Inkrement oder Dekrement zum Aufbau von Vektoren benutzt,
  - Bsp.: 1:10
    baut einen Zeilenvektor auf, 1 2 3 4 5 6 7 8 9 10 0:2:10 => 0 2 4 6 8 10
  - **Kann als Auswahloperator dienen**
  - Bsp.: A(:,1) liefert die erste Spalte von A, A(2,:) liefert die zweite Zeile von A
  - **Weiterhin kann der : als Verknüpfung eingesetzt werden**
  - Bsp.: A(:) => an die erste Spalte werden alle folgenden angehängt

Vektor und Matrix Operatoren

- Die Transposition
  - Hierzu dienen die bekannten Operatoren **.** und **‘**
  - Vertauschen von Spalten und Zeilen
    - Bsp.: **A = B.‘**
    - Bsp.: A = B‘ => Elemente werden zusätzlich konjugiert komplex genommen (siehe Komplexe Zahlen)
- Matrix Funktionen
  - find(), any(), all(), linspace()
    - Bsp.: find(A <= 5 & A>=3)

MATLAB Funktionen

- Allgemeine Hilfe zu Funktionen help elfun
- Hilfe zu Funktionen help [function]
- Wichtige mathematische Funktionen
  - sin, cos, tan, cot
  - exp, log, sqrt
  - abs, angle, konj, imag, real
  - fix, floor, ceil, round, sign
  - fft, dft

MATLAB Funktionen

- Funktionen selbst erzeugen
  - neues M-file erstellen und Deklaration der eigenen Funktion mit Schlüsselwort function zu Beginn
  - Funktion eingeben und speichern
  - Aufruf aus Programm mit Funktionsnamen
  - Argumente können übergeben werden mit function(x,y)
  - Variablen in der Funktion sind lokale Variablen, ansonsten global davor schreiben

Programmsteuerung

- Bedingung, Verzweigung
  - if, elseif, else
  - switch, case
- Schleifen
  - for, while
- Schleifenoperationen
  - break, return, continue
- Schleifen werden in MATLAB häufig durch Vektoren ersetzt – mittels etime() kann man die Laufzeit messen

Programmsteuerung

- if, elseif, else
  - x = 1848
  - if x >= 1e9
  - [num2str(1e-9*x) 'G']
  - elseif x >= 1e6
  - [num2str(1e-6*x) 'M']
  - elseif x >= 1e3
  - [num2str(1e-3*x) 'k']
  - else
  - x
  - end

Programmsteuerung

- switch
  - Jahr=1438;
  - switch Jahr
  - case 1438, 'Melozzo da Forli'
  - case 1450, 'Lucca Signorelli'
  - case 1444, 'Sandro Botticelli'
  - case 1436, 'Andrea del Verrocchio'
  - end

  - Person='Sandro Botticelli';
  - switch Person
  - case 'Sandro Botticelli', 'geboren 1444'
  - case 'Fra Angelico', 'geboren 1387'
  - otherwise, [Person 'ist unbekannt']
  - end

Programmsteuerung

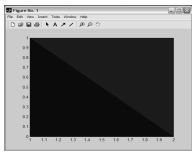
- for
  - sum=0;
  - n=10;
  - for ind=1:n;
  - sum=sum+ind^2;
  - end
  - Sum
- while
  - x=1;
  - q = (x^2-3)/(2*x);
  - while abs(q) > 1.0e-8
  - x = x -q;
  - q = (x^2-3)/(2*x);
  - end
  - [x q x^2]

Grafische Ausgabe

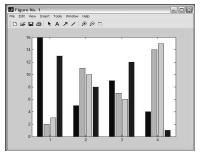
- Für die grafische Ausgabe zwei elementare Befehle – Hilfe mit help plot, stem…
  - disp (Variable, String …)
  - stem (x,y)
  - plot (x,y)
- Erweiterungen
  - grid (Gitternetz)
  - xlabel (‘x-Achse beschriften \rightarrow‘)
  - ylabel (‘y-Achse beschriften \leftarrow‘)
  - title (‘Titel der Grafik‘)

Grafische Ausgaben

- area(eye(2,2));
- area(rand(3,2));



- bar(rand(10,3))
- bar(magic(4))



- stairs([2 4 6 8 6 4 2 0]);

Grafische Ausgaben

Mesh & surf Grids:

- [X,Y]=meshgrid(-2:.1:2,-1:.1:1);
  Z=X.*exp(-X.^2-Y.^2);

- mesh(X,Y,Z);
- surf(X,Y,Z)

Grafische Ausgaben

- title = Überschrift : Beschriftung der Achsen
- legend: Legende
- text: Text in Grafik

- Beispiel:
- X=linspace(0,2*pi);
- Y=[sin(X);cos(X)];
- plot(X,Y)
- title('Trigonometrische Funktionen')
- xlabel('Winkel \alpha');
- text(pi/4,sin(pi/4),' sin \pi/4 = cos \pi/4')
- axis equal
- axis tight
- grid on
- legend(['sin \alpha';'cos \alpha'],-1)

Beispiel

```
t = 0:.01:15;          % time variable
xt = sin(t);           % sinus function
n  = 0:15;             % normalized time
Ts = pi/4;             % sampling interval
xn = sin(Ts*n);        % sinus sequence

% graphics
FIG1 = figure('Name',,MySinus','NumberTitle','off');
subplot(2,1,1), plot(t,xt)
grid, xlabel('t [s] \rightarrow'), ylabel('x(t) \rightarrow')
title('sinus function')
%second graphic
subplot(2,1,2), stem(n,xn)
grid, xlabel('n \rightarrow'), ylabel('x[n] \rightarrow')
title('sinus sequence')
% end
```