Mathematische Computer-Software

- Kommerzielle Computeralgebrasysteme (CAS)
  - Beispiele: Mathematica, Maple, …
    - Numerisches und symbolisches Verarbeiten von Gleichungen:
      - Grundrechenarten
      - Ableitung und Integration
      - Lösen von Gleichungssystemen
      - Durch Paket-Erweiterungen auch vieles mehr
    - Visualisierung von Graphen

Mathematische Computer-Software

- Umgebung zur Entwicklung math. Programme
- Graphische Oberfläche
  (teilw. mit Unterstützung math. Sonderzeichen)

- Open Source CAS
  - Beispiel: Maxima
    - Steht unter der GNU General Public License (GPL)
    - Auch sehr umfangreich, aber nicht so komfortabel
    - Graphische Oberfläche (wxMaxima)

Mathematische Computer-Software

- Weitere kommerzielle math. Software
  - Beispiel: MATLAB (MATrix LABoratory)
    - Auf numerische Berechnung mit Matrizen ausgelegt
    - Häufig eingesetzt für
      - Numerische Simulation
      - Datenerfassung
      - Datenanalyse und -auswertung
    - In Kombination mit
      - Simulink (zeitgesteuerte Simulation)
      - Stateflow (ereignisorientierte Simulation)
    - Sehr mächtig

Mathematische Computer-Software

- Open Source Alternative
  - Beispiel: GNU Octave (aktuell Version 3.6.4) [1]
    - Steht unter der GNU General Public License (GPL)
    - In weiten Teilen kompatibel zu MATLAB
    - Kommandozeilenbasiert (feste Entwicklungsumgebung und GUI ab Version 4.0 geplant);
      ABER: Für Windows gibt es z.B. GUI Octave [2]
    - Umfassende Dokumentation (online und als PDF) [3]

[1] http://www.gnu.org/software/octave/
[2] http://www.softpedia.com/get/Science-CAD/GUI-Octave.shtml
[3] http://www.gnu.org/software/octave/doc/interpreter/

Eine kurze Einführung in GNU Octave

- Variablen
  - Interpretierte Sprache, dadurch keine Deklaration notwendig (anders als in Java, C,...)
  - Erste Zuweisung eines Wertes erzeugt Variable
  - Variablennamen müssen mit Buchstaben anfangen und dürfen Buchstaben, Zahlen und Unterstriche enthalten
  - Unterscheidung zwischen Groß- und Kleinschreibung

Eine kurze Einführung in GNU Octave

- Variablen
  - Beispiele
    - Variable anlegen und Wert zuweisen

      ```
      >> a=1024
      ```
    - Variable ausgeben

      ```
      >> a
      ```
  - Ein Semikolon hinter der Wertzuweisung unterdrückt die Ausgabe

    ```
    >> a=1024;
    ```

Eine kurze Einführung in GNU Octave

- Variablen
  - In der Regel werden Variablen in Octave als nxm-Matrizen behandelt
    - Skalare sind 1x1-Matrizen
    - Zeilenvektoren sind 1xm-Matrizen
    - Spaltenvektoren sind nx1-Matrizen
- Kommentare
  - Ein Kommentar wird mit einer Raute (#) eingeleitet

Eine kurze Einführung in GNU Octave

- Vektoren
  - Beispiele
    - Zeilenvektor erzeugen

```
>> a=[1,2,3]
```

    - Spaltenvektor erzeugen

```
>> a=[1;2;3]
```

    - Spezielle Funktionen zum Füllen von Vektoren

```
>> a=1:5 # erstellt den Vektor a=(1 2 3 4 5)
>> a=1:2:5 # erstellt den Vektor a=(1 3 5)
>> a(i)=[] # löscht den i-ten Eintrag von a
```

Eine kurze Einführung in GNU Octave

- Vektoren
  - Beispiele
    - Länge eines Vektors

      >> length(a) # gibt die Länge des Vektors a aus
    - Zugriff auf Vektoren

      >> a(i) # greift auf die i-te Komponente des Vektors a zu

      >> a(1:3) # erstellt einen Vektor aus den ersten drei Einträgen des Vektors a
    - Transponieren von Vektoren

      >> a' # wandelt Spalten- in Zeilenvektoren um und umgekehrt

Eine kurze Einführung in GNU Octave

- Matrizen
  - Beispiele
    - Matrix erstellen

```
>> A=[1,2,3;4,5,6;7,8,9] # erstellt 3x3-Matrix A
>> v=[1,2,3]; A=[v;v] # erstellt 2x3-Matrix A mit Zeilen v
```

    - Spezielle Befehle für Matrizen

```
>> eye(n) # erstellt nxn-Einheitsmatrix
>> zeros(n) # erstellt nxn-Nullmatrix
>> zeros(n,m) # erstellt nxm-Nullmatrix
>> ones(n) # erstellt nxn-Matrix aus Einsen
>> ones(n,m) # erstellt nxm-Matrix aus Einsen
```

Eine kurze Einführung in GNU Octave

- Matrizen
  - Beispiele
    - Größe einer Matrix

```
>> size(A) # gibt für eine nxm-Matrix A einen Zeilenvektor (n m) aus
```

    - Zugriff auf Matrizen

```
>> A(i,j) # greift auf den i,j-ten Eintrag von Azu
>> A(i,:) # greift auf die i-te Zeile von A zu
>> A(i,:)=[] # löscht diej-te Zeile von A
>> A(:,j) # greift auf die j-te Spalte von A zu
>> A(:,j)=[] # löscht die j-te Spalte von A
```

Eine kurze Einführung in GNU Octave

- Arithmetische Operatoren
  - Octave kennt die Operatoren + (Addition),- (Subtraktion),* (Multiplikation),/ (Division) und ^ (Potenz)
  - Operatoren werden als Matrix-Operationen interpretiert
  - Das Voranstellen eines Punktes erzwingt die komponentenweise Interpretation des Operators
  - Beispiel

```
>> # Seien A und B nxn-Matrizen
>> A*B # führt die Matrixmultiplikation der Matrizen A und B durch
>> A.*B # multipliziert den (i,j)-ten Eintrag der Matrix A mit dem (i,j)-ten Eintrag der Matrix B für alle i,j zwischen 1 und n
```

Eine kurze Einführung in GNU Octave

- Vergleichsoperatoren
  - Octave kennt die Operatoren == (gleich), ~= (ungleich), > (größer), < (kleiner), >= (größer gleich), <= (kleiner gleich)
  - Das Ergebnis ist wahr (1) oder falsch (0)
  - Beispiele

```
>> x=3;
>> x==2 # liefert 0 (falsch)
>> x~=2 # liefert 1 (wahr)
```

Eine kurze Einführung in GNU Octave

- Logische Operatoren
  - Octave kennt die Operatoren | (oder), & (und) und ~ (nicht)
  - Das Ergebnis ist wahr (1) oder falsch (0)
  - Beispiele

```
>> x=3;
>> (x>=3)&(x<5) # liefert 1 (wahr)
>> ~(x>2) # liefert 0 (falsch)
```

Eine kurze Einführung in GNU Octave

- Funktionen
  - Beispiele

```
>> abs(·) # Absolutbetrag, bei Matrizen komponentenweise
>> sqrt(·) # Wurzelfunktion
>> exp(·) # Exponentialfunktion
>> sin(·), cos(·), etc. # trigonometrische Funktionen
>> sum(·) # Summe der Einträge eines Vektors
```

- Hilfen und Basisfunktionen
  - Beispiele

```
>> who # zeigt die Variablenbelegung an
>> help <function> # gibt Hinweise zur Funktion aus
>> doc # ausführliche Dokumentation zu Befehlen
>> clear <name> # löscht die Variable
```

Eine kurze Einführung in GNU Octave

- Editor
  - Ausführung zuvor in einem Editor erstellter Skripte (alternativ zur kommandozeilenorientierten Eingabe)
  - Speicherung der Octave-Befehle in einer Datei (m-Files <name>.m)
  - Wiederverwendbarkeit von Funktionen

Eine kurze Einführung in GNU Octave

- Schleifen und Verzweigungen
  - for-Schleife
    - Syntax

```
for <Laufindex> = <Schleifenbeginn>:<Schleifenende>
<Befehl 1> ...
<Befehl n>
end
```

    - Beispiel

```
>> sum=0;
for k=1:20
sum=sum+k;
end
>> sum # gibt die Summe der Zahlen von 1 bis 20 aus
```

Eine kurze Einführung in GNU Octave

- Schleifen und Verzweigungen
  - while-Schleife
    - Syntax

```
while <zu erfüllende Bedingung>
<Befehl 1> ...
<Befehl n>
end
```

    - Beispiel

```
>> sum=20;
while sum>0
sum=sum-1;
end
>> sum # gibt 0 zurück
```

Eine kurze Einführung in GNU Octave

- Schleifen und Verzweigungen
  - if-Verzweigung
    - Syntax (kann um die Kontrollstruktur elseif ergänzt werden)

```
if <zu erfüllende Bedingung>
<Befehl 1>
else
<Befehl 2>
end
```

    - Beispiel

```
>> x=2; y=1;
if (x>y) z=1;
else z=0;
end
>> z # gibt 1 zurück, da x>y wahr ist
```

Eine kurze Einführung in GNU Octave

- Schleifen und Verzweigungen
  - Allgemeine Befehle

```
>> break # setzt die Bearbeitung direkt nach der Schleife fort
>> continue # springt aus der Schleife direkt zurück zur Überprüfung der Bedingung
```

Viel Spaß beim Ausprobieren!