Eine kurze Einführung in GNU Octave (Teil 2)

- Strings
  - Strings stehen in Anführungszeichen
    - Beispiel

```
>> string="abc"; # erzeugt den String abc
>> s(1)="a"; s(2)="b" # s=ab
```

  - Octave kennt nützliche Funktionen für Strings
    - Beispiele

```
>> string="acb"; sort(string) # sortiert einen String, liefert also abc
>> strcmp(s1,s2) # prüft ob die Strings s1 und s2 gleich sind
>> num2str(n) # konvertiert eine Zahl in einen String
>> str2num(s) # konvertiert einen String in eine Zahl
>> findstr("aba","a") # liefert [1 3], also die Stellen an denen a steht
```

Eine kurze Einführung in GNU Octave (Teil 2)

- Strukturierte Datentypen
  - Mit Octave lassen sich Datentypen auch strukturieren
    - Beispiel

```
>> x.a="Dies ist eine Struktur";
>> x.b=3.3;
# liefert
x=
{
    a=Dies ist eine Struktur
    b=3.3000
}
```

Eine kurze Einführung in GNU Octave (Teil 2)

- Grafische Ausgabe
  - plot(x,y) visualisiert zu einem Vektor von Stützstellen x die zugehörigen Funktionswerte

```
>> x=0:0.01:10;
>> plot(x,x.^2)
```

    - Beschriftung des Plots
      - Titel title(“Titel des Plots“);
      - Achsen xlabel(“Beschriftung der x-Achse“);
      - ylabel(“Beschriftung der y-Achse“);
      - Legende legend(“plot1“,“plot2“,...);
    - figure(i) öffnet mehrere Ausgabefenster
    - hold on ermöglicht die Ausgabe mehrer Plots in einem Fenster

Eine kurze Einführung in GNU Octave (Teil 2)

- Skripte und Funktionen, M-Files
  - Skripte sind Dateien der Form <name>.m in denen Octave-Befehle gesammelt sind
  - Um eine Datei aus dem Verzeichnis zu laden muss der Pfad gesetzt sein

```
>> addpath("C:\Users\...\Octave"); # setzt einen Pfad
```

  - Dateien mit Endung *.m werden dann automatisch geladen und Variablen/Funktionen stehen zur Verfügung
  - Laden und Speichern einer Datei

```
>> load "<name>.m" # lädt die Datei
>> save "<name>.m" # speichert die Datei
```

Eine kurze Einführung in GNU Octave (Teil 2)

- Funktionen
  - Können direkt definiert werden oder in M-File gespeichert werden (dann kein load notwendig)
  - Funktionsaufbau

```
function [<out1>,...,<outN>]=<name>(<in1>,...<inM>)
<Funktionsbefehle>
end
```

  - M-File wird unter dem Funktionsnamen <name>.m gespeichert
  - Wichtig: Datei muss in dem Verzeichnis gespeichert werden, in dem sie ausgeführt werden soll

Eine kurze Einführung in GNU Octave (Teil 2)

- Funktionen
  - Bestehen aus drei Teilen
    - Funktionskopf
      - Eingabevariablen <in1>,...,<inM>
      - Ausgabevariablen <out1>,...,<outN>
      - Funktionsname <name>
    - Funktionsrumpf
      - Funktionsbefehle
      - Setzung der Ausgabevariablen
    - Funktionsende end

Eine kurze Einführung in GNU Octave (Teil 2)

- Funktionen
  - Beispiel (Liste der ersten Fibonacci-Zahlen)

```
>> function [x]=fibonacci(n)
>>         x(1)=0;
>>         if n>0
>>                 x(2)=1;
>>                 for i=2:1
>>                         x(i+1)=x(i)+x(i-1);
>>                 end
>>         end
>> end
>> fibonacci (10) # liefert [0 1 1 2 3 5 8 13 21 34 55]
```

Eine kurze Einführung in GNU Octave (Teil 2)

- Integration
  - Numerische Integration
    - In Octave möglich
    - Nutzt standardmäßig als Verfahren die Gauß-Quadratur
    - Zuvor muss die Funktion definiert werden
    - Beispiel

```
>> function y=f(x) # erstellt die Funktion f(x)=sin(x)
>> y=sin(x)
>> endfunction
>> quad("f",pi,2*pi) # liefert den Wert -2 des Integrals der Funktion sin(x) in den Grenzen von π bis 2π
```

    - Anleitung zur Integration in Octave [1]

[1] http://www.gnu.org/software/octave/doc/interpreter/Functions-of-One-Variable.html

Eine kurze Einführung in GNU Octave (Teil 2)

- Integration
  - Symbolische Integration
    - In Octave erst einmal nicht möglich
    - Zusatzpaket „Symbolic toolbox“ [1] ermöglicht symbolische Integration;
      - Jedoch nicht mehr aktuell (letzter Stand: 18.10.2011), daher ggf. inkompatibel mit neueren Versionen von Octave
      - Folglich nicht zu empfehlen
    - Auch in MATLAB nicht im Standardfunktionsumfang; erst mit der „Symbolic Math Toolbox“ [2] durchführbar

[1] http://octave.sourceforge.net/symbolic/index.html

[2] http://www.mathworks.de/products/symbolic/

Eine kurze Einführung in GNU Octave (Teil 2)

- Integration
  - Symbolische Integration
    - Alternative: Integration in Maxima [1]
      - Beispiel

        Maxima --> integrate(sin(x),x);
        liefert: -cos(x)

        Maxima --> integrate(sin(x),x,pi,2*pi);
        liefert: cos(π)-cos(2 π)

        Maxima --> integrate(sin(x),x,%pi,2*%pi);
        liefert: -2
      - Ausführliche deutschsprachige Dokumentation zu Maxima [2]

[1] http://maxima.sourceforge.net/

[2] http://maxima.sourceforge.net/docs/manual/de/maxima_toc.html