EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

MATHEMATISCH-
NATURWISSENSCHAFTLICHE FAKULTÄT
Fachbereich Geowissenschaften

Einführung in Matlab

5. Funktionen

Prof. Dr. Olaf Cirpka, Prof. Dr. Christiane Zarfl, Dipl.-Inf. Willi Kappler

Sie wissen bereits...

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- wie Sie durch Scripte Befehlsfolgen wiederverwertbar machen.
- wie Sie häufig vorkommende Berechnungen in Algorithmen formulieren.

Wie kann ich eigene Funktionen für verschiedene Eingaben erstellen und damit Berechnungsschritte wiederverwenden?

Nach diesem fünften Block...

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- können Sie interaktive Eingaben anfordern und Ausgaben erstellen.
- können Sie eigene Funktionen definieren und z.B. in Scripten aufrufen/wiederverwenden.

Ein- und Ausgabe

- Ausgabe eines Textes im Comand-Window:
  - `disp` steht für "display"
  - Text-Strings sind von Hochkommata umschlossen
  - `disp` versteht auch Vektoren: `disp(['He' 'llo' 'World'])`
- Interaktive Eingabe:
  - ```
    a=input('Bitte Halbwertszeit [Tage]
    für Photoabbau eingeben:  ')
    ```
  - erzeugt den Text auf dem Bildschirm und wartet, bis eine Eingabe abgeschlossen ist (Return beendet die Eingabe)
  - Der eingegebene Wert steht dann in der Variable `a`
    ```

Ein- und Ausgabe Fortsetzung

- Beispiel:
  - `jn = input('Abbruch?  (j/n) ','s')`
  - Zusatzargument 's' erklärt, dass Ergebnis als Text-String zu lesen ist (selbst wenn der String aus Ziffern besteht)
  - Umgang mit Fehlern der Nutzer ⇒ typische Schleife, bis richtige Antwort kommt:

```
1  jn = ''; % Initialisierung
2  while jn~='j' & jn~='n'
3    jn = input('Abbruch? (j/n) ','s')
4  end
5  disp('Ihre Eingabe:')
6  disp(jn)
```

Übung 5-1

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Überlagerung von zwei Sinusschwingungen (schwing.m)

- Darzustellende Funktion:
- $y(x) = a_1 sin(fx + \phi_1) + a_2 sin(fx + \phi_2)$
- Interaktive Eingabe:
  - gemeinsame Frequenz `f`
  - Amplitude der ersten Komponente `a1`
  - Amplitude der ersten Komponente `a2`
  - Phasenwinkel der ersten Komponente `phi1`
  - Phasenwinkel der zweiten Komponente `phi2`
- Graphische Ausgabe im Intervall $[-\pi, +\pi]$

Skripte und Funktionen

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- "Idee" eines Scriptes: ersetzt manuelle Eingabe einer Befehlsfolge:
  - Variablen stehen auch nach Ausführung des Scriptes zu Verfügung
  - Wenn Variablenwerte im Script verändert werden, sind sie auch außerhalb des Scriptes verändert
  - Ein Script kann ein anderes Script aufrufen
    - mit gemeinsamer Nutzung des Arbeitsspeichers
  - Nachteile eines Scriptes
    - Allgemeiner Arbeitsspeicher wird mit Zwischengrößen "zugemüllt"
    - Variablenbezeichnung muss über alle Scripte hinweg konsistent sein (Vorsicht! Seiteneffekte)

Skripte und Funktionen Fortsetzung

- "Idee" einer **Funktion**: "Input $\Rightarrow$ Output"-Beziehung:
  - Von der Außenwelt ist nur das bekannt, was als Input-Argument(e) übergeben wird
  - Der Außenwelt wird lediglich das Ergebnis (Output-Argument(e)) mitgeteilt
  - Innenwelt der Funktion ist von außen nicht einsehbar
    - Eine Funktion hat ihren eigenen Speicherbereich

Vorteile von Funktionen

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- Modularisierung
  - Unterteile ein großes Projekt in viele Einzelschritte und stelle Methoden für die Einzelschritte zur Verfügung
  - Eigener Namensraum, in verschiedenen Funktionen kann es die lokale Variable `i` oder `t` geben
- Wiederverwendbarkeit
  - Identischer Einzelschritt kann mehrfach, auch in anderen Projekten, verwendet werden
- Einkapselung
  - Wenn eine Funktion funktioniert, interessiert mich nicht ihr Innenleben, und ich will nicht mit Zwischenergebnissen belästigt werden
  - Black Box Prinzip (s.a. Information Hiding)

Funktionen erfordern eine Schnittstelle

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Funktionen Definition in Matlab:

- Definiert in der ersten Zeile
- `function [out1,out2]=myfunction(in1,in2)`
  - Erstes Wort = Schlüsselwort `function`
  - Liste der **Outputargumente** (in eckigen Klammern, also als Vektor)
  - Name der Funktion mit Liste der **Inputargumente** in runden Klammern
- Die Schnittstelle ist definiert über die Reihenfolge der Argumente, nicht über die Namen
  - **Vorteil**: Variablennamen im Innern können sich von den Variablennamen außen unterscheiden
  - **Nachteil**: Man muss sich die Reihenfolge merken
  - Voraussetzung für allgemeine Wiederverwertbarkeit