



Einführung in Matlab

1. Grundlagen

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



1 Einleitung

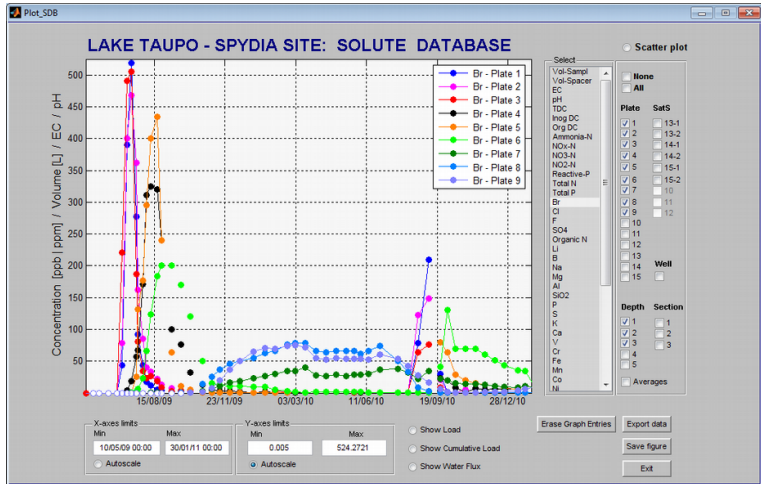
2 Grundlagen



- Einfache Rechnungen
- Datenaufbereitung & -speicherung
- Visualisierung von Daten & Simulationsergebnissen
- einfaches Lösen von (linearen) Gleichungssystemen, Differentialgleichungen (DGL), DGL-Systemen...
- Wiederverwendung von häufig gebrauchten Berechnungen (Programmierung)
- Datenanalyse (z.B. Regression) & Statistik
- Komplexe Modellierung von Umweltsystemen

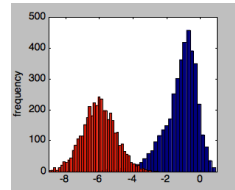
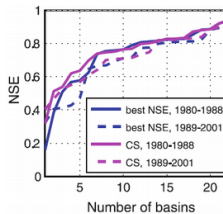
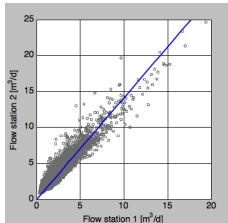
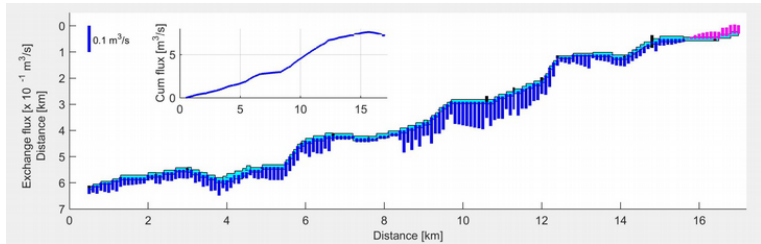


Visualisierung von Daten & Simulationsergebnissen



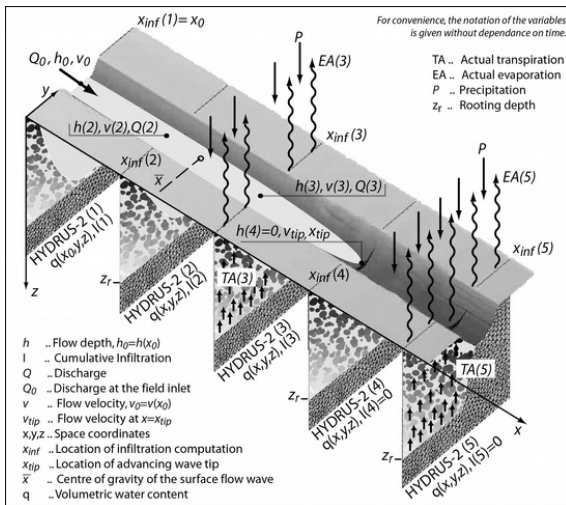


Datenanalyse & Statistik





Modellierung von Umweltsystemen





- Hardware:
 - BYOD (eigenes Gerät mitbringen)
 - Geo-Notebooks (Raum S245)
 - CIP Pool Rechner (Raum S310)
- Software:
 - Auf Institutshardware bereits vorinstalliert
 - ZDV: Matlab [▶ hier](#) herunterladen
 - GNU Octave [▶ Webseite](#) (Open Source, aber nicht 100% kompatibel)
- Auf Institutshardware bitte zuerst ein eigenes Verzeichnis anlegen!



- “*Matlab* ist eine kommerzielle Software des Unternehmens *The MathWorks, Inc.* zur Lösung mathematischer Probleme und zur grafischen Darstellung der Ergebnisse.” (Quelle: [▶ Wikipedia](#)).
- Matlab leitet sich ab von **MAT**rix **LAB**oratory.
- Wir benutzen Matlab als (numerische) Programmiersprache.
- Wie ein Taschenrechner oder Excel arbeitet Matlab numerisch (mit Zahlenwerten, also nicht symbolisch wie ein [▶ CAS](#)).
- Anders als bei einem Taschenrechner können Zahlenwerten Variablennamen zugewiesen werden.
- Im Programm werden die Variablennamen als Platzhalter für die Werte verwendet.



- kennen Sie den Aufbau der Oberfläche der Software Matlab.
- benutzen Sie die Matlab-Hilfe, um für Sie nützliche Funktionen und Informationen selbstständig zu finden.
- führen Sie einfache Rechnungen mit Matlab durch.
- können Sie Variablen in Matlab definieren und verwenden.
- kennen Sie die Vorteile der Verwendung von Vektoren und können diese in Matlab definieren und für Rechnungen verwenden.



gegenwärtiges Verzeichnis


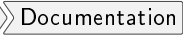

The screenshot shows the MATLAB R2014a environment. The 'Current Folder' pane on the left lists files: 'Hausaufgaben', 'Woche1', 'Woche2', 'Woche3', and 'template.m'. An arrow points from the text 'Dateien im gegenwärtigen Verzeichnis' to this pane. The 'Editor' pane in the center shows the 'template.m' script. A yellow highlight covers the code defining variables 'x', 'y1', 'y2', and 'y3'. An arrow points from the text 'Variablen im Arbeitsspeicher' to this highlighted code. The 'Workspace' pane on the right shows these variables: 'x' (1x51 double), 'y1' (1x51 double), 'y2' (1x51 double), and 'y3' (1x51 double). An arrow points from the text 'Eingabeverlauf' to this pane. The 'Command Window' at the bottom shows the prompt '»' and the command 'template'. An arrow points from the text 'Eingabefenster mit "Prompt" (»)' to this window.

↑
Dateien im
gegenwärtigen
Verzeichnis


↑
Variablen im
Arbeitsspeicher

↑
Eingabeverlauf

↑
Eingabefenster mit
"Prompt" (»)

- Niemand kann alle Befehle kennen, deshalb ist die (ausführliche) Hilfe in Matlab so wichtig.
- Allgemeines Hilfe-Fenster:   ()
- Information zu einem Befehl:
 - `doc <Befehlsname>` (Info in einem extra Fenster)
 - `help <Befehlsname>` (Info im Befehlsfenster)
- Beispiele: Im Prompt eingeben:
 - `help sin`
 - `help exp`
 - `doc plot`



- Alle Anweisungen werden nach dem Prompt (`>>`) eingegeben und mit  (Return) bestätigt. Matlab nennt das Ergebnis `ans` (für answer):

```
1 12/3 + 7*5 - 1
```

```
1 ans = 38
```

- Addition (\oplus), Subtraktion (\ominus), Multiplikation (\otimes) und Division (\oslash) wie im Taschenrechner (Matlab kennt “Punkt vor Strich-Rechnung”).
- Braucht man einen **Ausdruck öfters**, so kann man ihn als **Variable** definieren:

```
1 a = 48/3 - 3^2
```

```
1 a = 7
```

- Variablen werden im Arbeitsspeicher (Workspace) gespeichert (s. Arbeitsspeicher-Fenster).



- Ein Semikolon (;) am Ende der Eingabezeile unterdrückt die Ausgabe des Ergebnisses.
- Matlab **unterscheidet** zwischen Groß- und Kleinbuchstaben!
- Potenzieren wird **vor** einer **Multiplikation** oder **Division** ausgewertet, sonst gilt “Punkt-vor-Strich”; runde Klammern “(” und “)” um die Reihenfolge der Berechnung zu steuern.
- Mehrere Anweisungen in einer Zeile sind zulässig:
 - Sind sie durch ein Komma getrennt, so folgt eine Ausgabe.
 - Werden sie durch ein Semikolon getrennt, so folgt keine Ausgabe:

```
1 b = (3+5)*6;  
2 c = (b/3)^2, d = 1/c^2
```

```
1 c =    256  
2 d =    1.5259e-05
```