

# Programmierkonzepte in der Physikalischen Chemie

## 8. Toolboxes

Albert-Ludwigs-Universität Freiburg



**UNI  
FREIBURG**

Dr. Till Biskup

Institut für Physikalische Chemie  
Albert-Ludwigs-Universität Freiburg  
Wintersemester 2013/14

## Wiederholung: Was ist eine Toolbox?

Konzept

Toolboxen in Matlab

## Struktur von Matlab-Toolboxen

Contents.m

Hilfe richtig ablegen

Besondere Verzeichnisse

## Infrastruktur in Matlab

Dokumentation

Versionsnummern

Informationen zur Toolbox

Installationsroutine

Dateien auf oberster Ebene

# Was ist eine Toolbox?

Kurze Wiederholung: Zwei Konzepte

Grundlegend zwei Konzepte:

- 1 ein Skript für jeden Datensatz
  - 2 eine Toolbox aus Funktionen, die generisch jeden Datensatz verarbeiten kann
- ☞ Beide Konzepte haben Vor- und Nachteile.

## Toolbox

In sich geschlossene Sammlung von Funktionen (Routinen) für eine bestimmte Aufgabe

# Was ist eine Toolbox?

Kurze Wiederholung: Werkzeugkasten



- ▶ Ein Werkzeug, eine Funktion
- ▶ Es gibt unterschiedliche Werkzeugkästen/Toolboxen für unterschiedliche Aufgaben.
- ▶ Ein gut sortierter Werkzeugkasten ist essentiell.

## Toolboxen in Matlab

- ▶ Matlab selbst ist sehr modular aufgebaut.
- ▶ Viele hilfreiche zusätzliche Funktionalität ist über Toolboxen realisiert.
- ▶ Toolboxen sind in sich geschlossene Sammlungen von Funktionen für eine bestimmte Aufgabe.
- ▶ Matlab bietet Unterstützung bei der Entwicklung eigener Toolboxen.
- ▶ Es gibt grundsätzlich zwei Arten von Toolboxen
  - 1 kommerziell (meist von MathWorks selbst)
  - 2 nichtkommerziell (meist frei im Netz verfügbar)

### Kommerzielle Toolboxen

- ▶ Optimization Toolbox
- ▶ Global Optimization Toolbox
- ▶ ...

### Nichtkommerzielle Toolboxen

- ▶ EzyFit Toolbox
- ▶ EasySpin
- ▶ DEER Analysis
- ▶ trEPR Toolbox, TA Toolbox
- ▶ ...

### Ein Wort zu kommerziellen Toolboxen

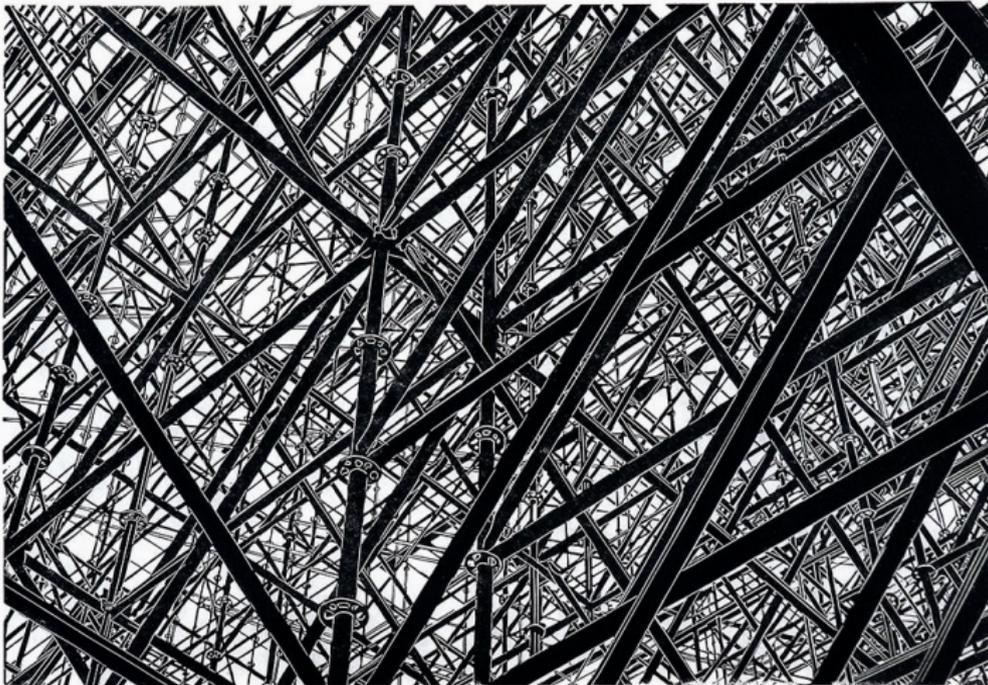
- ▶ Die Landeslizenz bietet Zugriff auf alle Matlab-Toolboxen.
  - ▶ Die Landeslizenz ist vorerst zeitlich limitiert.
  - ▶ Die Landeslizenz gilt nur für Baden-Württemberg.
  - ▶ Eine (akademische) Einzelplatzlizenz für eine Toolbox kostet ca. 300–400 EUR zzgl. jährlicher Lizenzkosten.
- 
- Sparsamer Einsatz kommerzieller Toolboxen erhöht die Portabilität und Wiederverwertbarkeit von Code.
  - Für den [AK Weber](#) gilt: Nur Optimization und Global Optimization Toolbox verwenden.

# Struktur von Matlab-Toolboxen

Hilfsmittel von Matlab



UNI  
FREIBURG



Philipp Hennevogel: Struktur (2009)

*Matlab unterstützt bei der Toolbox-Entwicklung*

- ☛ Zeit, das wiederholt gegebene Versprechen einzulösen

## Hilfsmittel von Matlab

- ▶ Hilfe zu den Dateien in einem Ordner
  - `Contents.m`
- ▶ Strukturierung der Hilfe im HTML-Format
  - `info.xml`, `helptoc.xml`, `demos.xml`
- ▶ Organisation der Dateien
  - `private`-Verzeichnis zum Verstecken von Routinen

## Die Datei `Contents.m`

### Listing 1: Einfachste Form einer `Contents.m`-Datei

```
1 % trEPR Toolbox
2 % Version 0.3.29 06-Feb-2014
```

### Listing 2: Zusätzliche Angabe eines Namens für eine Version

```
1 % TA Toolbox
2 % Version 0.0.31 (alpha30) 06-Feb-2014
```

- ▶ Besteht nur aus Kommentarzeilen
- ▶ Wird vom Befehl `ver` ausgewertet
- ▶ Zentraler Ort für die Ablage von Versionsnummern

## Hilfe im HTML-Format

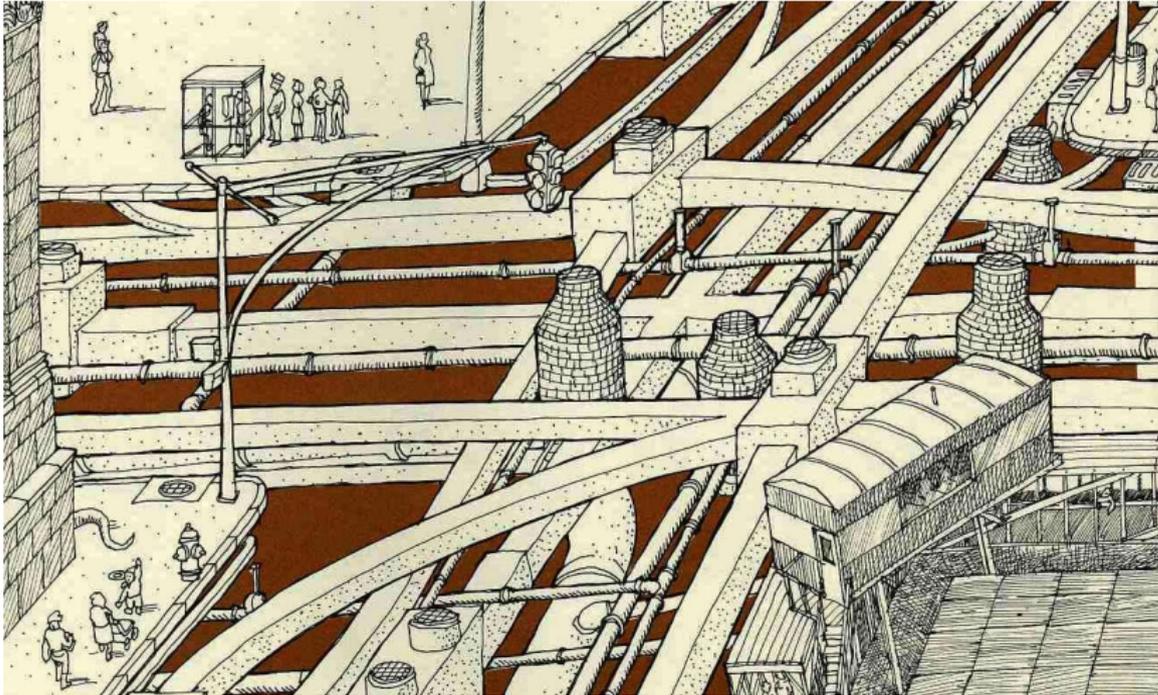
- ▶ Gesamte (eingebaute) Matlab-Hilfe im HTML-Format
  - Matlab-Hilfefenster (`doc`) ist ein HTML-Browser
  - HTML erlaubt auch komplexere Formatierungen
  - Matlab unterstützt nur ein Subset von HTML/CSS
- ▶ Drei Dateien zur Organisation der Hilfe
  - `info.xml`
  - `helptoc.xml`
  - `demos.xml`
- ▶ Hilfe durchsuchbar machen
  - `builddocsearchdb`

## Organisation der Dateien

- ▶ `private`-Verzeichnis
  - Dateien erscheinen nicht im Matlab-Suchpfad
  - Funktionen nur aus dem Verzeichnis darüber aufrufbar
  - Ähnlich dem "private"-Konzept der OOP
  
- ▶ Vorteile
  - Unterfunktionen lassen sich in eigene Dateien auslagern
  - Geeignet z.B. für die Programmlogik einer GUI
  - Mehrere Funktionen gleichen Namens möglich
  
- ▶ Grenzen
  - Funktionen nur aus dem Verzeichnis darüber aufrufbar
  - Modularität: DRY-Prinzip beachten!

# Infrastruktur

Oft unsichtbar, aber überlebenswichtig



David Macaulay: Unter einer Stadt. dtv, München, 1977

## Innerhalb von Matlab implementierte Infrastruktur

- ▶ Erweitert die von Matlab bereitgestellte Funktionalität
- ▶ Innerhalb von Matlab implementiert und nutzbar

## Konkrete Konzepte

- ▶ Dokumentation
  - ▶ Versionsnummern
  - ▶ Info-Funktion
  - ▶ Installationsroutine
  - ▶ Grundlegende Informationen (README, INSTALL, ...)
- ☞ Haben sich in der Praxis bewährt

## Dokumentation

- ▶ Ziel
  - Anwenderdokumentation direkt in der Toolbox
  - Beschreibung der Konzepte
  
- ▶ Umsetzung
  - HTML-Format
  - Im Matlab-Hilfebrower (`info.xml`, `helptoc.xml`)
  - GUIs: eigenes, eingebautes Hilfefenster
  
- ▶ Tipps aus der und für die Praxis
  - GUIs: Eigenes Hilfefenster mit HTML-Browser (Java)
  - HTML: logische Textauszeichnung; CSS: Formatierung
  - Hilfe über `info.xml/helptoc.xml` zusätzlich einbinden

## Versionsnummern

- ▶ Ziel
  - Eindeutige Versionsnummern und Veröffentlichungsdaten
  - Einfacher Zugriff mit „Matlab-Bordmitteln“
- ▶ Umsetzung
  - Zentrale Ablage in einer Datei: `Contents.m`
- ▶ Tipps aus der und für die Praxis
  - Versionsnummer in `Contents.m` immer aktuell halten (Revisionsnummer wird bei jeder Änderung erhöht!)
  - Eindeutiges und intuitives Versionierungsschema (z.B. „Semantic Versioning“)

## Informationen zur Toolbox

- ▶ Ziel
  - Ein Funktionsaufruf liefert alle relevanten Informationen
  
- ▶ Umsetzung
  - Info-Routine (liefert *struct* zurück)
  - Enthält alle Infos (außer Versionsnummer)
  
- ▶ Zurückgelieferte Informationen
  - Name, Version und Veröffentlichungsdatum (`Contents.m`)
  - Autor(en), ggf. mit Email-Adressen
  - Internetseite mit weiteren Informationen
  - URL des Bug-Trackers
  - URL des Repositorys (Versionskontrollsystem)

## Installationsroutine

- ▶ Ziel
  - Ein Funktionsaufruf erledigt alle Installationsaufgaben
- ▶ Umsetzung
  - Installationsroutine
- ▶ Mögliche Aufgaben
  - Matlab-Suchpfad aktualisieren
  - Durchsuchbare Dokumentation aufbauen
  - Kompilieren von Quellcode (Fortran/C/C++/...)
  - Anlegen notwendiger Verzeichnisse
  - Anlegen einer Konfiguration aus Konfigurationsvorlagen

## Dateien auf oberster Ebene

- ▶ Ziel
  - Schneller Zugang zu Aufgaben und Installation
  - Hinweise auf Copyright und ggf. Lizenz
  
- ▶ Umsetzung
  - README, INSTALL
  - COPYRIGHT, LICENSE
  
- ▶ Tipps aus der und für die Praxis
  - Reine Textdateien (ASCII 7-bit, *keine* Umlaute)
  - Kurz und prägnant halten
  - Dateinamen sind keine Empfehlung, sondern Konvention
  - Lizenz: möglichst keine eigene, sondern existierende

*...gleich geht's weiter: „Finale furioso“*

## Vorschau: [Programmentwicklung](#)

- ▶ Wohin soll die Reise gehen?
- ▶ Infrastruktur
- ▶ Konzepte
- ▶ Datenverarbeitung
- ▶ Nutzerschnittstellen
- ▶ Projektplanung