

Programmierkonzepte in der Physikalischen Chemie

2. Allgemeines

Albert-Ludwigs-Universität Freiburg



**UNI
FREIBURG**

Dr. Till Biskup

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

„Kenne Deine Programmiersprache“

- Dokumentation zur Hand haben und nutzen
- Weitere Informationsquellen kennen

Der richtige Editor

Verzeichnisstrukturen

„Matlab in fünf Minuten“

- Bezugsquellen und Verfügbarkeit
- Das Matlab-Fenster
- Hilfe und Dokumentation
- Toolboxen

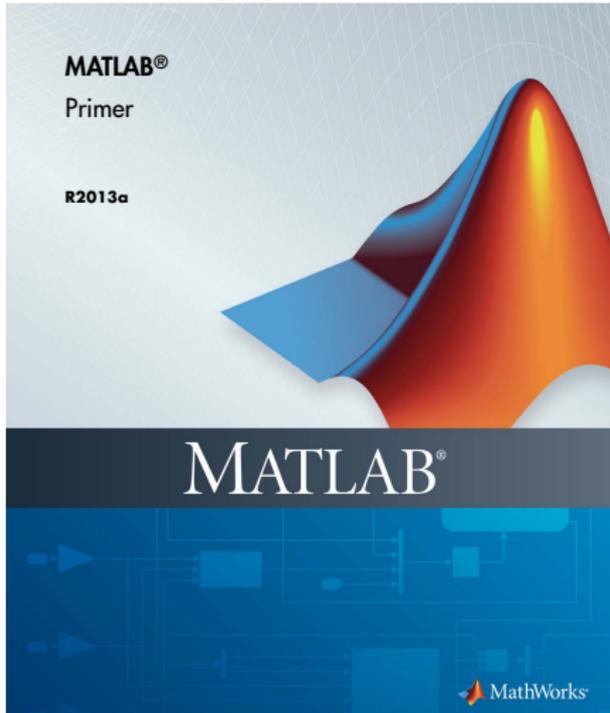
Satz

Man muss nicht alles wissen, sollte aber wissen, wo es steht.

- ▶ Programmieren lernen ist wie eine Sprache lernen.
 - ▶ Grundlegende Sprachkonzepte müssen bekannt sein.
 - ▶ Details können in der Dokumentation nachgeschlagen werden.
-
- 👉 Kenntnis der vorhandenen Dokumentation und wie man sie nutzt.

„Kenne Deine Programmiersprache“

Dokumentation zur Hand haben und nutzen



Inhalte

- ▶ Quick Start
- ▶ Language Fundamentals
- ▶ Mathematics
- ▶ Graphics
- ▶ Programming

Es gibt eine Vielzahl **weiterer Informationsquellen**:

- ▶ Bücher und gute Webseiten zu Programmierkonzepten
 - ▶ Einführende Veranstaltungen an der Universität
 - ▶ Kollegen, Betreuer, Freunde
 - ▶ Matlab: Matlab Central, Matlab File Exchange
-
- ☛ Programmieren grundlegend zu erlernen geht am Besten alleine mit einem guten Buch.
 - ☛ Eigeninitiative und Interesse sind essentiell.

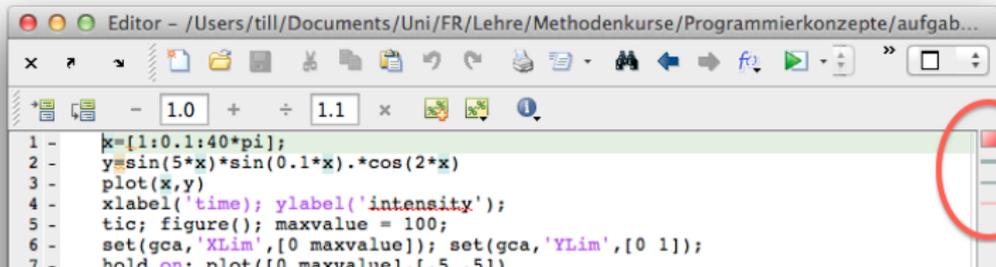
Vorteile moderner Editoren

- ▶ automatische Codevervollständigung
 - ▶ Codeüberprüfung während der Eingabe
 - ▶ Hilfe aus dem Editor heraus erreichbar
 - ▶ Syntaxhervorhebung („Syntax highlighting“)
 - ▶ automatische Codeeintrückung
 - ▶ Zusammenfalten von Codeteilen („Code folding“)
 - ▶ Refaktorisierung („Refactoring“)
- ☛ Der Matlab-Editor unterstützt die meisten der genannten Kriterien mittlerweile recht gut.

Codeüberprüfung im Matlab-Editor

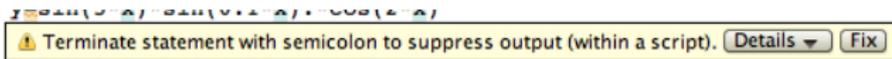
- ▶ Der Matlab-Editor zeigt drei Zustände an.
(alles in Ordnung, Warnungen, Fehler)
 - ▶ Für Warnungen und Fehler können zusätzliche Hinweise angezeigt werden.
 - ▶ Häufig wird für Warnungen und Fehler eine automatische Behebung angeboten („Autofix“).
 - ▶ Warnungen können ignoriert/abgeschaltet werden
(im Einzelfall sinnvoll).
- ☛ Warnungen und Fehler sollten *in jedem Fall* ernst genommen und deren Ursache behoben werden.

Codeüberprüfung im Matlab-Editor



```
1 - x=[1:0.1:40*pi];
2 - y=sin(5*x)*sin(0.1*x).*cos(2*x)
3 - plot(x,y)
4 - xlabel('time'); ylabel('intensity');
5 - tic; figure(); maxvalue = 100;
6 - set(gca,'XLim',[0 maxvalue]); set(gca,'YLim',[0 1]);
7 - hold on; plot(0,maxvalue,1.5 .5)
```

Autofix im Matlab-Editor



Codeeintrückung: Allgemeine Aspekte

- ▶ Automatische Codeeintrückung erhöht die Lesbarkeit.
- ▶ Beginn und Ende von Schleifen sind einfach erkennbar.
- ☛ Saubere Codeeintrückung ist nicht optional.

Codeeintrückung im Matlab-Editor

- ▶ Der Editor beherrscht automatische Codeeintrückung.
- ▶ Codebereiche können im Nachhinein automatisch eingerückt werden.

Refactoring (Refaktorisierung, Restrukturierung)

Fortentwicklung einer Programmierschnittstelle, die keine Änderungen in den Anwenderprogrammen nach sich zieht.

Beispiele

- ▶ Umbenennung von Variablen und Funktionen
- ▶ Auslagerung von Code in eigene Funktionen
- ☞ Der Matlab-Editor beherrscht momentan nur rudimentäres Refactoring (Variablenumbenennung).

- ☛ Moderne Editoren sind „Integrierte Entwicklungsumgebungen“ (IDEs)

Beispiele weiterer IDEs

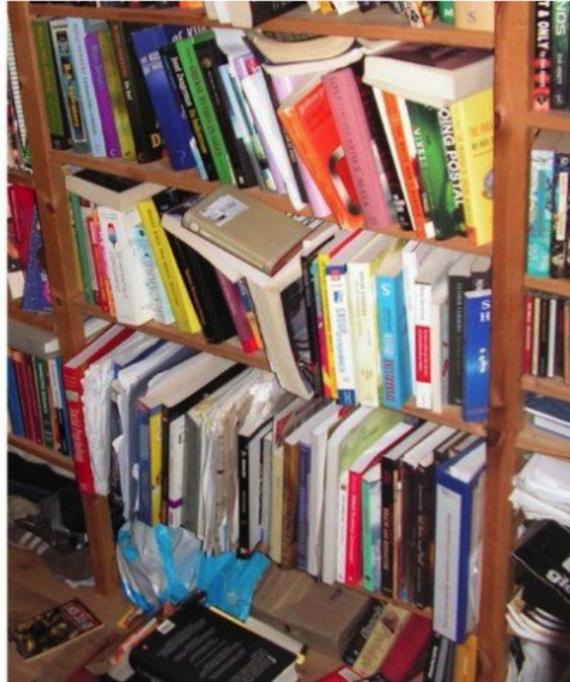
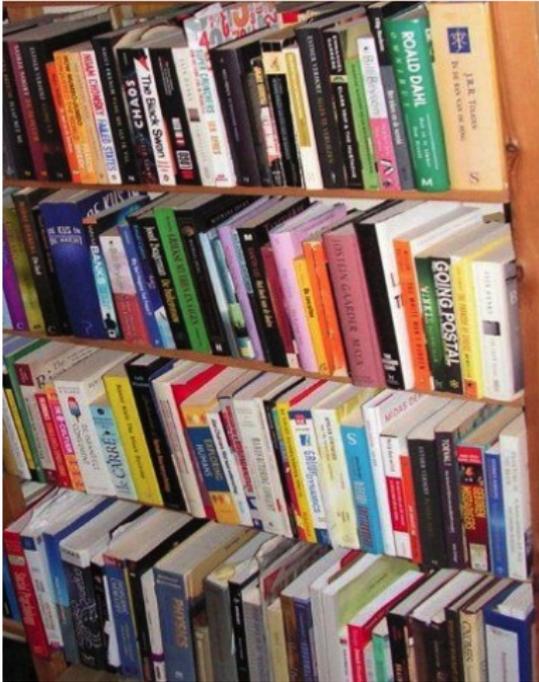
- ▶ Eclipse, IntelliJ, NetBeans
 - sehr modular
 - für eine Vielzahl an Programmiersprachen verwendbar
- ▶ Kile, TeXshop, ...
 - spezifisch für die Verwendung mit \LaTeX
- ☛ Letztlich ist die Wahl des Editors bzw. der IDE eine Frage persönlicher Vorlieben.

Verzeichnisstrukturen

„Ordnung ist das halbe Leben“



UNI
FREIBURG



Warum Strukturen?

- ▶ Matlab hat einen (manuell zu pflegenden) Suchpfad
 - Matlab kennt nur, was in diesen Verzeichnissen liegt.
- ▶ Struktur sorgt für Übersichtlichkeit
 - Übersicht ist wichtig für die Nachvollziehbarkeit.
 - Übersichtlichkeit kann manche Dokumentation ersetzen.
 - Übersichtlichkeit erleichtert die Übernahme der Pflege des Codes durch andere Personen.
- ☞ Nachvollziehbarkeit und Reproduzierbarkeit sind wichtiges Gebot in den Wissenschaften.
- ☞ Struktur ist letztlich **Verpflichtung, nicht Option.**

Grundprinzipien

- ▶ Trennung von Routinen und Daten
- ▶ Jede Toolbox in einem eigenen Verzeichnis
- ▶ Alle Matlab-Funktionen möglichst in einem Verzeichnisbaum

Struktur innerhalb einer Toolbox

- ▶ Zusammengehörige Funktionen in ein Verzeichnis
- ▶ Dokumentation und Beispiele

Verzeichnisstruktur einer (realen) Toolbox:

- IO allgemeine Ein- und Ausgaberoutinen
- processing Verarbeitung der Rohdaten
 - analysis Analyse der vorverarbeiteten Rohdaten
 - CLI textbasierte Nutzerschnittstelle
 - GUI graphische Nutzerschnittstelle
- common allgemeine Funktionen
- internal interne Funktionen (Installationsroutine, ...)
- doc Dokumentation
- examples Beispiele



Inhalte

- ▶ Bezugsquellen (Landeslizenz)
- ▶ Übersicht über das Matlab-Fenster
- ▶ Hilfe und Dokumentation
- ▶ Konzept der Toolboxen (kommerziell und frei)

- ☛ Keine Einführung in Matlab und dessen grundlegende Sprachkonzepte.
- ☛ Dafür gibt es einen eigenen Methodenkurs:
„Anwendungen von Mathematica und Matlab in der PC“



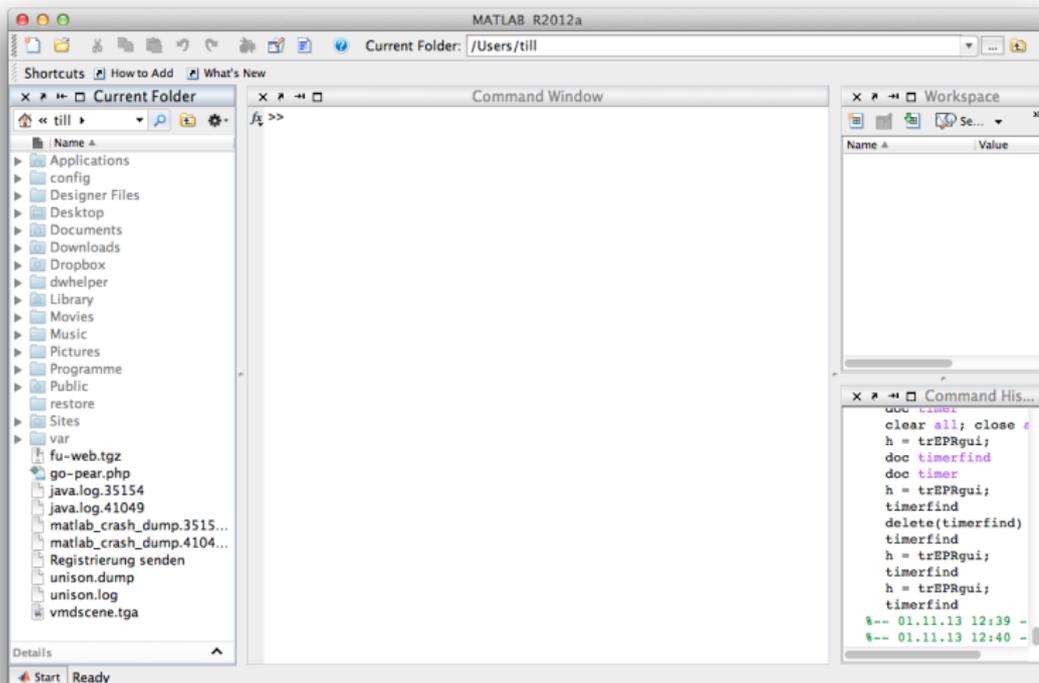
- ▶ Matlab ist in Baden-Württemberg über eine **Landeslizenz** an allen Universitäten verfügbar.
- ▶ Die Lizenz erlaubt die Installation auf **privaten Computern**.
- ▶ Zum Bezug muss auf der MathWorks-Seite ein persönliches Konto angelegt werden.
 - Dazu ist eine Uni-Email-Adresse zwingend notwendig.
- ▶ Details auf den Seiten des Rechenzentrums

Seite des Rechenzentrums

<https://www.rz.uni-freiburg.de/services/beschaffung/software/matlab-landeslizenz>

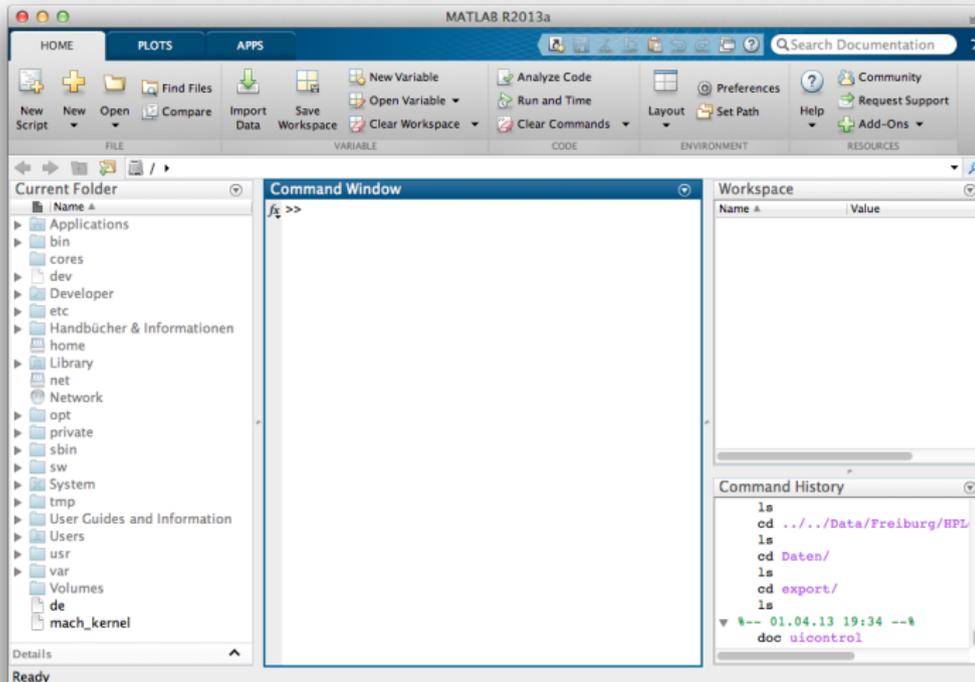
„Matlab in fünf Minuten“

Das Matlab-Fenster (2012a)



„Matlab in fünf Minuten“

Das Matlab-Fenster (2013a)



Elemente des Matlab-Fensters

- ▶ **Command Window**
 - „Kommandozeile“
 - Das Fenster, in das Befehle direkt eingetippt werden.
 - ▶ **Workspace**
 - Übersicht über die momentan definierten Variablen
 - ▶ **Current Folder**
 - Anzeige des aktuellen Verzeichnisses
 - ▶ **Command History**
 - Übersicht über die Befehlshistorie
- ☞ Es gibt darüber hinaus noch weitere Fenster und Elemente, z.B. den Editor, Abbildungen, ...

Offline verfügbar

- ▶ **Eingebaute Hilfe in Matlab**
 - `help <Befehlsname>` (auf der Kommandozeile)
 - `doc <Befehlsname>` (eigenes Fenster, ausführlicher)
- ▶ **Handbücher zu Matlab und kommerziellen Toolboxes**
 - mittlerweile nur noch elektronisch als PDF-Dokumente
 - Zugriff nur mit Konto bei der MathWorks-Seite
- ▶ **Bücher**
- ▶ **Kollegen, Betreuer, Freunde**

Online verfügbar

- ▶ MathWorks-Webseite
 - Webcasts (kleine Filme) zur Einführung
 - Matlab Central
 - Matlab File Exchange

- ▶ <http://undocumentedmatlab.com/>
 - Richtet sich eher an Experten
 - Sehr viele Interna zu Matlab

- ▶ (manche) Kollegen, Betreuer, Freunde

- ▶ <http://lmgty.com/>

„Matlab in fünf Minuten“

Hilfe und Dokumentation: Matlab Central



Trendy
Connect the dots

Cody
Let the games begin



MathWorks
Careers

Find Code
Solve Problems

File Exchange

Recent Files

- Image Registration App *Brett Shelton*
- Maximum Weight Independent Set instance *Richard*
- Multidimensional path-generator *Erwin Torreao Dassen*
- MIMO aliamouti *Leila nasraoui*
- psotoolbox *Sandeep Solanki*
- CIRCLE FIT IN HEART SHAPE *Prashant Somani*

Cody

Recent Problems

- Kaggle: Reverse Game of Life - Single Move to One Cell Case *Richard Zapor*
- Grid traversal *Ziko*
- Finding neighbors of [-1:1] in a matrix... *Chris E.*

MATLAB Answers

Recent Questions

- I need matlab code for probabilistic box counting (PBC) algorithm. *Vinay Shrivastava*
- Help plotting interpolation polynomial. *Yuvof*
- How to scan a hex file and then search for the required byte and display *Prashant*
- Simscape - transforming a rotational motion into an oscillating translational motion *Rano*
- How to fix code? *John Foster*
- how to convert grayscale image to rgb image *taushend jan*

Trendy

Popular Plots

Files on the File Exchange *Ned Gallely*

Blogs

Recent Updates

- Guy and Seth on Simulink *30 Oct 2013*
View archive
- MATLAB Spoken Here *29 Oct 2013*
View archive
- Cleve's Corner: The Intel Hypercube, part 1 *28 Oct 2013*
View archive
- File Exchange Pick of the Week *25 Oct 2013*
View archive
- Doug's MATLAB Video Tutorials *24 Oct 2013*

<http://www.mathworks.com/matlabcentral/>

File Exchange

- Files
- Categories
- Authors
- Tags
- Comments

Submit a File

About File Exchange

Search Files | Advanced Search

Browse

- Functions
- Apps
- Examples
- Simulink Models
- Videos
- Instrument Drivers
- Hardware Support Packages

Most Recent (see all)

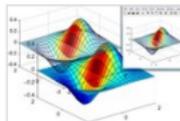
Image Registration App by Brett Shoelson



UI environment for registering a Moving image to a Fixed image

Most Popular (see all)

export_fig by Oliver Woodford



Exports figures nicely to a number of vector & bitmap formats.

<http://www.mathworks.com/matlabcentral/fileexchange/>

Ein Wort zu „Google-Lösungen“

- ▶ Code immer erst verstehen und dann einsetzen
- ▶ Schwarm-Intelligenz sorgt meist nicht für bessere Code-Qualität.
- ▶ Google verhilft zu schnellen Lösungen – aber:
Oft sind „offizielle“ oder spezifische Quellen besser.
- ☞ Viele Wege führen nach Rom.
Man kann von anderen viel lernen, sollte sich aber immer die Mühe machen, deren Code zu verstehen.

- ▶ 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 viel 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.

So long, and thanks for all the fish.

Vorschau: Modularität, Dokumentation im Code

▶ Modularität

- einzelne Routinen statt „Spaghetti-Code“
- eine Aufgabe, eine Routine
- klar definierte Schnittstellen

▶ Dokumentation im Code

- keine Datei ohne Dokumentationskopf
- grundlegende Struktur einer Dokumentation
- Dokumentationshilfen