

# Anwendung von (Mathematica und) Matlab in der Physikalischen Chemie

## 2. Einleitung – Matlab

Albert-Ludwigs-Universität Freiburg

Dr. Till Biskup

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



**UNI  
FREIBURG**

Einsatzgebiet in der Physikalischen Chemie

Stärken und Schwächen von Matlab

Allgemeines zu Matlab

Toller Taschenrechner – und Programmiersprache

Was man mit Matlab u.a. alles machen kann

Bezugsquellen und Verfügbarkeit

Matlab – eine praktische Einführung

Das Matlab-Fenster

Der Matlab-Editor

Abbildungen in Matlab

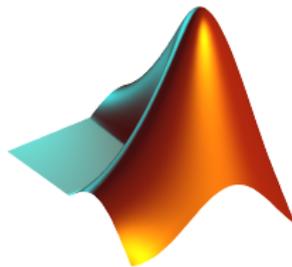
### Mathematica



- ▶ (analytische) Lösungen komplexer Gleichungen
- ▶ Symbolische Mathematik

### Matlab

- ▶ Datenverarbeitung
- ▶ Numerische Simulationen
- ▶ Komplexere Programmierungen



## Motivation zur Verwendung von Matlab

- ▶ Wir haben Daten gemessen und wollen diese Daten auswerten und (sinnvoll) darstellen.

## Warum Matlab

- ▶ Komplexe Auswertungen
  - Nicht mehr einfach bzw. per Hand durchführbar
  - Automatisierung durch Programmierung einzelner Schritte
  - Reproduzierbarkeit und Nachvollziehbarkeit
- ▶ Einfache Erlernbarkeit von Matlab
  - Programmiersprache mit Ähnlichkeiten zu C und Pascal
  - Grafische Oberfläche
  - Relativ einfach und schnell erste Erfolge

### Ein reales Beispiel: Fluoreszenz-Versuch aus dem PCG

- ▶ Ausgangslage
  - Daten wurden alle gemessen
  - Daten liegen als Textdateien (ASCII) vor
  
- ▶ Zielstellung
  - Vollständige Auswertung gemäß Fragestellung
  - Abbildungen, die den Assistenten zufriedenstellen (und den wissenschaftlichen Standards entsprechen)
  
- ▶ Vorgehen
  - 1 Pflichtenheft erstellen (was muss getan werden?) ✓
  - 2 Notwendige Grundlagen von Matlab aneignen
  - 3 Auswertung gemäß Pflichtenheft in Matlab programmieren

## Stärken

- ▶ Relativ einfach erlernbar
- ▶ Mächtig (vollwertige Programmiersprache)
- ▶ Gut geeignet für „Rapid Prototyping“:  
Schnell (halbwegs) vernünftige Ergebnisse

## Schwächen

- ▶ Kommerziell
  - Akademische Einzelplatzlizenz: ca. 1000 EUR
  - Pro kommerzieller Toolbox zusätzlich ca. 300-400 EUR
  - Zzgl. jährlicher Lizenzkosten
- ▶ Langsam

# Allgemeines zu Matlab

Ein toller Taschenrechner – und noch viel mehr



Thales Patent Rechenmaschine, ab ca. 1935



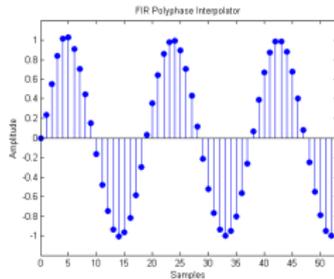
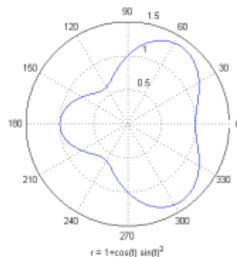
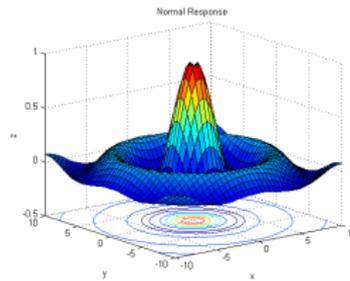
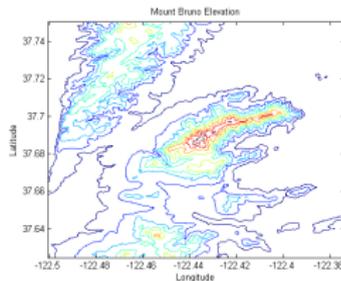
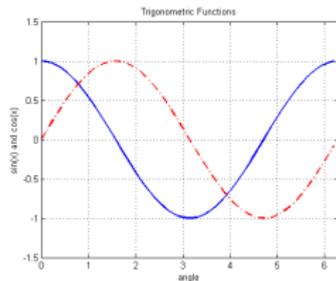
## Toller Taschenrechner – und vollwertige Programmiersprache

- ▶ Toller Taschenrechner
  - Auf der Matlab-Kommandozeile
  - Matlab beherrscht viele mathematische Funktionen
  - Ausführliche Befehlshistorie
- ▶ Vollwertige Programmiersprache
  - An Pascal und C angelehnt
  - Erlaubt die Programmierung komplexer Auswertungen
  - Unterstützung moderner Programmierkonzepte in jüngerer Zeit dazugekommen (z.B. OOP)

## Was man mit Matlab u.a. alles machen kann

- ▶ Abläufe automatisieren
  - Befehlsfolgen in Skripten oder Funktionen
  - Gleiche Verarbeitung ähnlicher Daten
- ▶ Abbildungen erstellen
  - Matlab unterstützt viele verschiedene Abbildungstypen
  - (Halbwegs) publikationsfähige Abbildungen
- ▶ Komplexe Auswerteprogramme schreiben
  - „Toolboxen“
  - Nutzerschnittstellen (CLI und GUI)

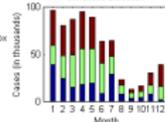
## Galerie



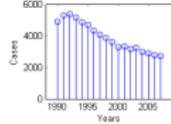
Childhood Diseases



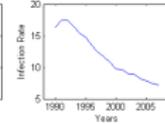
Childhood Diseases



Tuberculosis Cases



Tuberculosis Cases



<http://www.mathworks.de/discovery/gallery.html>

## Toolboxen zur Simulation von Spektren



### EasySpin – *by Stefan Stoll*

MATLAB toolbox for simulating and fitting Electron Paramagnetic Resonance (EPR) spectra.

- ▶ *De-facto*-Standard für die EPR-Spektrensimulation
- ▶ Komplette kommandozeilenbasiert
- ▶ Implementiert viele verschiedene Algorithmen
- ▶ Gemeinsame Schnittstelle für alle Simulationen

# Allgemeines zu Matlab

Was man mit Matlab u.a. alles machen kann



## Toolboxen mit grafischen Schnittstellen





- ▶ 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

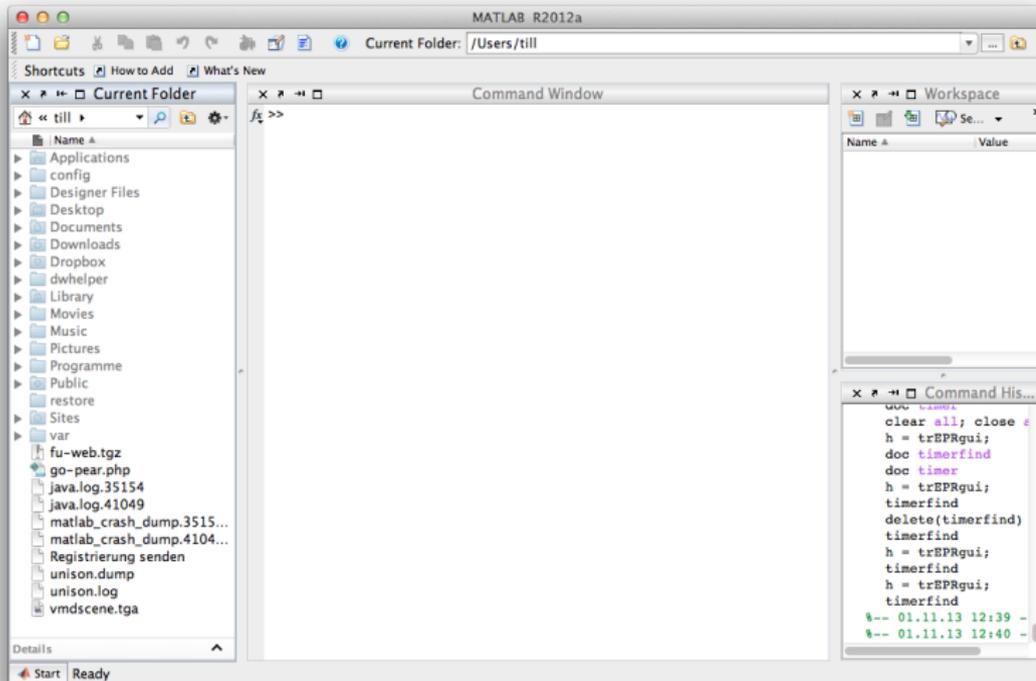
`https://www.rz.uni-freiburg.de/services/beschaffung/software/matlab-landeslizenz`

### Ein erster Blick auf das Arbeiten mit Matlab

- ▶ Matlab ist eine „integrierte Entwicklungsumgebung“ (IDE)
  - Bringt alles zur Arbeit Notwendige mit
  - Das Meiste spielt sich in einem Fenster ab.
- ▶ Einzelne „Teile“ von Matlab
  - Hauptfenster (mit Kommandozeile, ...)
  - Editor (für die Programmierung)
  - Abbildungen (Grafikfenster)
- ▶ Zwei Arbeitsmodi
  - Befehle direkt auf der Kommandozeile eingeben
  - Befehlslisten in Skripten/Funktionen

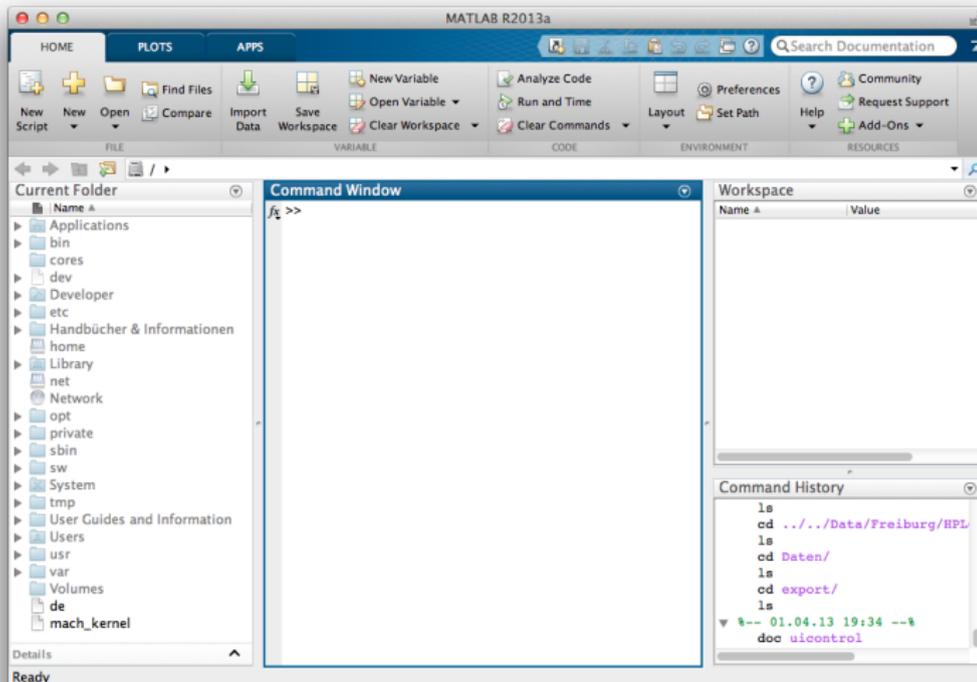
# Matlab – eine praktische Einführung

## Das Matlab-Fenster (2012a)



# Matlab – eine praktische Einführung

## 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, ...

# Matlab – eine praktische Einführung

## Der Matlab-Editor



Editor - /Users/till/Documents/Uni/FR/Lehre/PCG/Fluoreszenz/Alt/Daten/PCG-TB/LS45ASCIIread.m

EDITOR PUBLISH VIEW

New Open Save Find Files Compare Print Insert Comment Indent Go To Breakpoints Run Run and Time Run and Advance

```
function [data,status,warnings] = LS45ASCIIread(filename,varargin)
% LS45ASCIIREAD Read ASCII files written by PerkinElmer LS45 spectrometer
%
% Usage:
% data = LS45ASCIIread(filename)
% [data,status,warnings] = LS45ASCIIread(filename)
%
% filename - string
%           Name of file to read
%
% data      - struct
%           structure containing data and header
%
% status    - scalar
%           Return value for the exit status:
%           0: command successfully performed
%           -1: file unreadable
%
% warnings  - cell array
%           Contains warnings/error messages if any, otherwise empty
%
% (c) 2013, Till Biskup
% 2013-05-02
%
status = 0;
warnings = cell(0);
% Some internal configuration
separator = '\t';
```

Ln 1 Col 1

### Was muss ein guter Editor können?

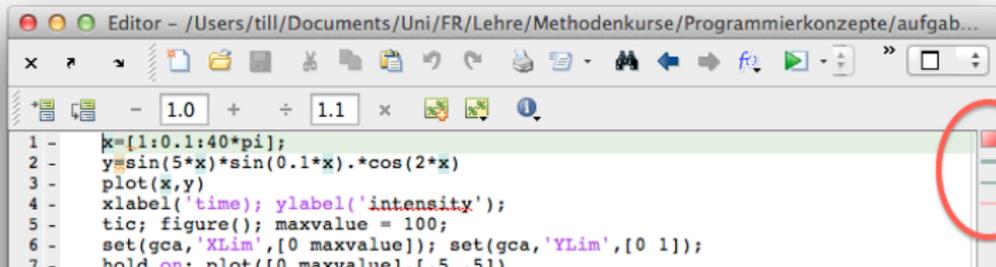
- ▶ automatische Codevervollständigung
- ▶ Codeüberprüfung während der Eingabe
- ▶ Hilfe aus dem Editor heraus erreichbar
- ▶ Syntaxhervorhebung („Syntax highlighting“)
- ▶ automatische Codeeintrückung
- ▶ Zusammenfalten von Codeteilen („Code folding“)
- ▶ Refaktorisierung („Refactoring“)

☛ Der Matlab-Editor unterstützt die meisten der genannten Kriterien mittlerweile recht gut.

## Codeüberprüfung im Matlab-Editor

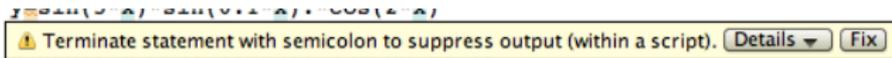
- ▶ Der Matlab-Editor zeigt drei Zustände an.  
(alles in Ordnung, Warnungen, Fehler)
  - ▶ Für Warnungen und Fehler können zusätzliche Hinweise angezeigt werden.
  - ▶ Häufig wird für Warnungen und Fehler eine automatische Behebung angeboten („Autofix“).
  - ▶ Warnungen können ignoriert/abgeschaltet 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



⚠ Terminate statement with semicolon to suppress output (within a script). [Details](#) [Fix](#)

### Codeeintrückung: Allgemeine Aspekte

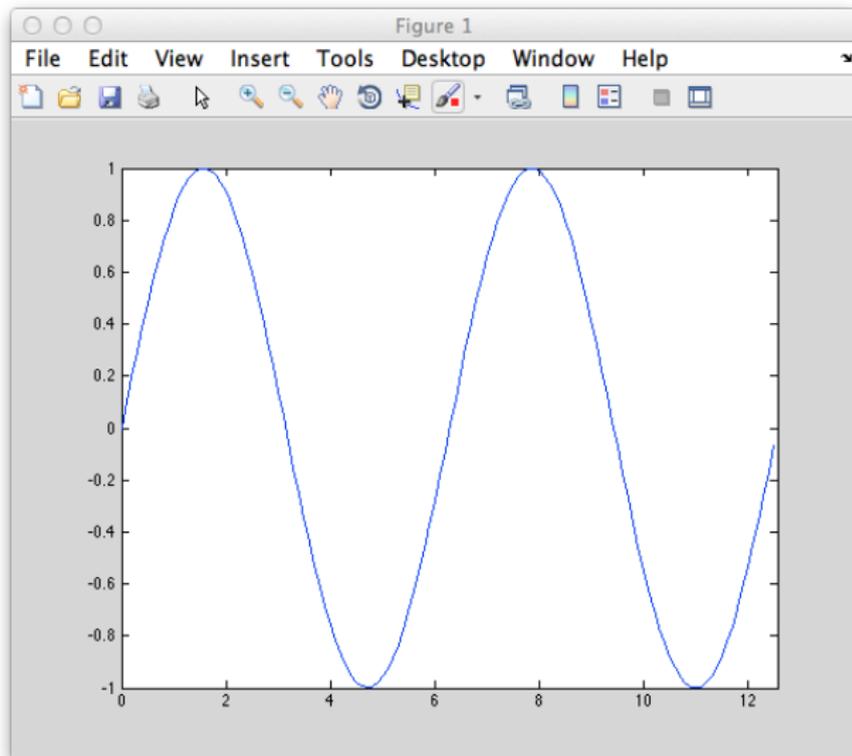
- ▶ Automatische Codeeintrückung erhöht die Lesbarkeit.
- ▶ Beginn und Ende von Schleifen sind einfach erkennbar.
- ☛ Saubere Codeeintrückung ist nicht optional.

### Codeeintrückung im Matlab-Editor

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

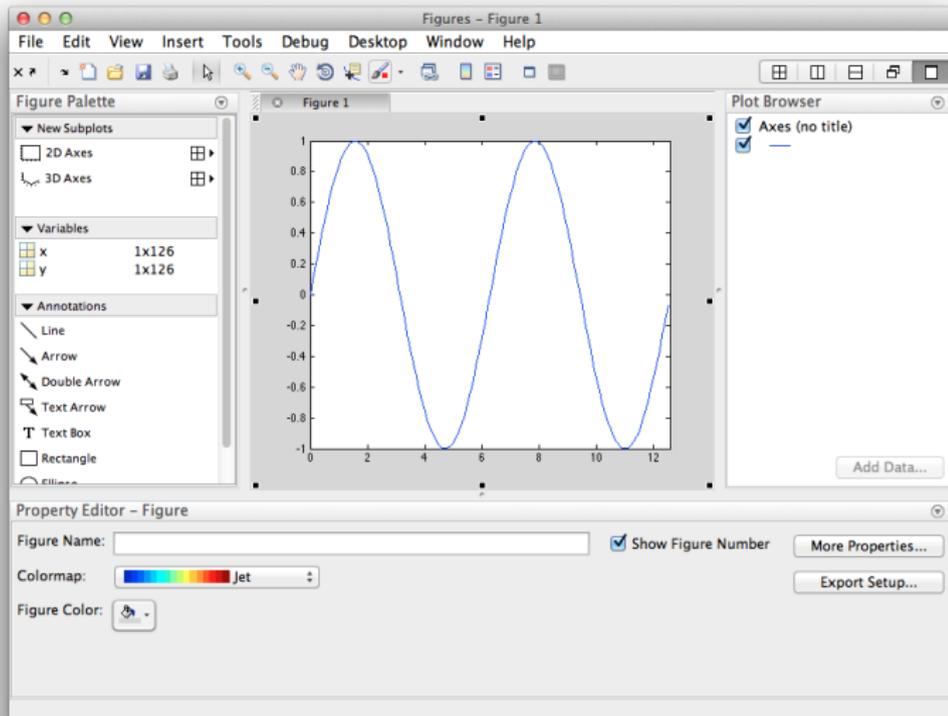
# Matlab – eine praktische Einführung

## Das Matlab-Abbildungsfenster



# Matlab – eine praktische Einführung

## Das Matlab-Abbildungsfenster



## Das Matlab-Abbildungsfenster

- ▶ Alle Eigenschaften über Kommandozeile steuerbar
  - Hilfreich für die automatisierte Erstellung von Abbildungen
  - Manches weder intuitiv noch trivial
- ▶ Achsenbeschriftungen
  - Auf korrekte Formatierung achten
  - Matlab unterstützt grundlegende  $\text{\LaTeX}$ -Befehle
- ▶ Export
  - Matlab unterstützt Export in Bitmap- und Vektorformate
  - Viele Einstellungen weder intuitiv noch trivial

## Man lernt nur durch Anwendung

- ▶ Matlab macht es dem neuen Nutzer leicht
  - Alles in einer grafischen Oberfläche
  - Schnelle Erfolge gewährleistet
  - Einfache Syntax
  - Einfach zu erlernende Sprache
  
- ▶ Ab jetzt wird es praktisch
  - Matlab ist auf den Rechnern des CIP-Pools installiert
  - Jedes Mitglied der Universität hat Zugriff via Landeslizenz
  
- ▶ Praktisches Beispiel: Auswertung Fluoreszenz-Versuch
  - 1 Pflichtenheft erstellen (was muss getan werden?) ✓
  - 2 Notwendige Grundlagen von Matlab aneignen (✓)
  - 3 Auswertung gemäß Pflichtenheft in Matlab programmieren

*...gleich geht's weiter*

### Vorschau: [Grundlegende Sprachkonzepte](#)

- ▶ Grundlegende Sprachkonzepte von Programmiersprachen
- ▶ Befehle/Funktionen
- ▶ Hilfe zur Selbsthilfe