



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

informatik
die zukunft

Seminar «Modellierung und Simulation»

Thema
Autor
Datum

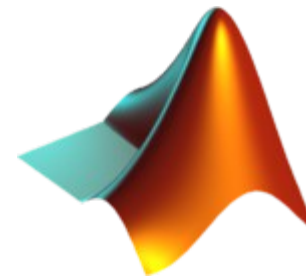
MATLAB und Simulink
Anna Fuchs
28. Januar 2013

Überblick

- Eckdaten
- Einsatzgebiet
- Einführung und Vorführung
 - Interaktiv
 - Programmieren
- Simulink
- Visualisierung
- Möglichkeiten und Grenzen
- Alternativen
- Fazit
- Quellen

Eckdaten - MATLAB

- **Matrix Laboratory**
- Kommerzielle Software zur Lösung mathematischer Probleme
- Gleichnamige Programmiersprache
- Ende 1970-er von Cleve Moler von der Universität New Mexico entwickelt (FORTRAN)
- Einst für Studenten ohne FORTRAN Kenntnisse entwickelt
- 1984 Unternehmen «The MathsWorks» gegründet
 - In C umgeschrieben
 - MATLAB wird kommerziell
- Im Netz eine freie Version von 1982 in Fortran zu finden
- Aktuell in C und Java geschrieben
- Geschätzt über 1 Mio. Benutzer unter Wissenschaftlern
- Sehr verbreitet in der Lehre
- Akademische Lizenz: 500\$ MATLAB + 500\$ Simulink



Logo

- Aktuelle Version: 8.0 (R2012b) vom 11.09.2012
- Unterstützte Betriebssysteme: Win, Mac OS, Linux, Unix, (ehemals Solaris)
- Sprache: Englisch und Japanisch (?)

Einsatzgebiet

- Primär für numerische (näherungsweise) Berechnungen entwickelt
- Auch symbolische (exakte) Berechnungen möglich
- Entwicklung von Algorithmen
- Datenerfassung und -bearbeitung
- Modellierung und Simulation (technischer Probleme)
- Wissenschaftliche und technische Darstellungen
- Grafische Visualisierung

Einführung 1

- **Kern**

- Nicht änderbar
- Hauptbestandteile
 - Programmiersprache
 - Arbeitsumgebung
 - Grafiksystem
 - Funktionsbibliothek
 - Programmschnittstelle (C und Fortran)

- **Erweiterungspakete** (z.B. Simulink)

- Änderbar, erweiterbar (Sammlung von .M Dateien)
- Werden nach Bedarf geladen
- Über 50 Pakete vorhanden
- `ver` — Befehl für verwendete Version inkl. verfügbare Pakete

Einführung 2 – MATLAB Übersicht

- Interaktiv
- Programmieren
 - Skripte
 - Funktionen
 - Programme
- Visualisierung
 - Direkte Befehle
 - Plot-Sammlung

Demo

Differentialgleichung

- Nur numerische Lösung möglich (ohne Toolbox)
- Keine Integrationskonstante
 - Intervall muss vorgegeben werden
- Keine Formvariable
 - $3x^2 + \mathbf{a}x + 4 \rightarrow$ nicht möglich
- ode45 Standard (Runge-Kutta)
 - Viele weitere Lösungsverfahren
- Nur von Ordnung 1 möglich
 - Umformen zu einem System 1. Ordnung

Differentialgleichungen

Demo

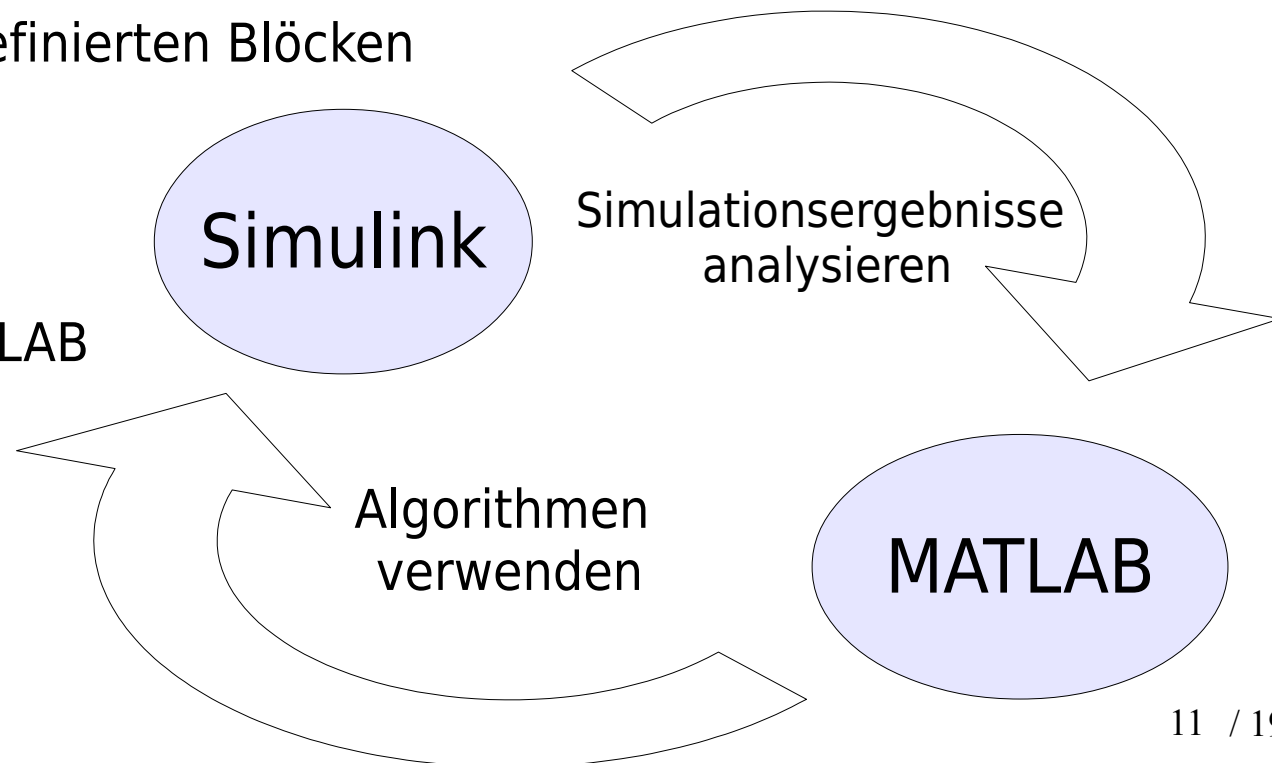
Programmieren

- Skripte — keine Parameter
 - Kommandofolge
 - Aufruf über Dateiname (ohne .M) oder *Save and Run*
- Funktionen — Ein- und Ausgabeparameter
 - `Function ergebnisparameter_liste = name_der_funktion (eingabeparameter_liste)`
 - Kommandofolge
 - Diverse regeln
- Programme mit grafischer Anwenderoberfläche (`guide`)

```
function dy = blub (x,y)  %...
dy = 2*y;                %...
```

- Schleifen, Kontrollstrukturen, Fehler abfangen
- Vordefinierte Funktionen
- Funktionsbibliotheken

- Blockdiagrammumgebung für Entwurf und Simulation
- Unterstützt kontinuierliches Testen und Verifizieren von eingebetteten Systemen
- Zeitgesteuerte Simulation (ereignisgesteuert — *Stateflow Toolbox*)
- Umfasst Grafikeditor
- Blockbibliotheken mit vordefinierten Blöcken
- Eigene Blöcke erstellen
- Interaktion mit MATLAB
- Aufruf `>>simulink` in MATLAB



Demo

Visualisierung



Demo

Möglichkeiten

- Codegenerierung aus z.B. Simulink Blöcken (*Simulink Coder Toolbox* — nicht in der Uni-Lizenz)
- Parallelisierung mit *Computing Toolbox* (vorhanden)
- Sehr hohe Kompatibilität
 - Programmschnittstelle (C, Fortran, Java, .NET)
- Import von Datenbanken, Bildern, Tabellen, Geräten direkt
- Objektorientierte Programmierung

Vorteile

- Leicht zu lernende umfangreiche Programmierumgebung
- Matrizenorientiertes Konzept
- Vielzahl an implementierten Funktionen
- Zuverlässiges, lange gepflegtes System
- Sehr hohe Kompatibilität
- Nicht «*coden*» (keine Variablen deklarieren, keinen Speicher allokkieren, keine Datatypen angeben)

Nachteile

- Lizenz
- Effizienz

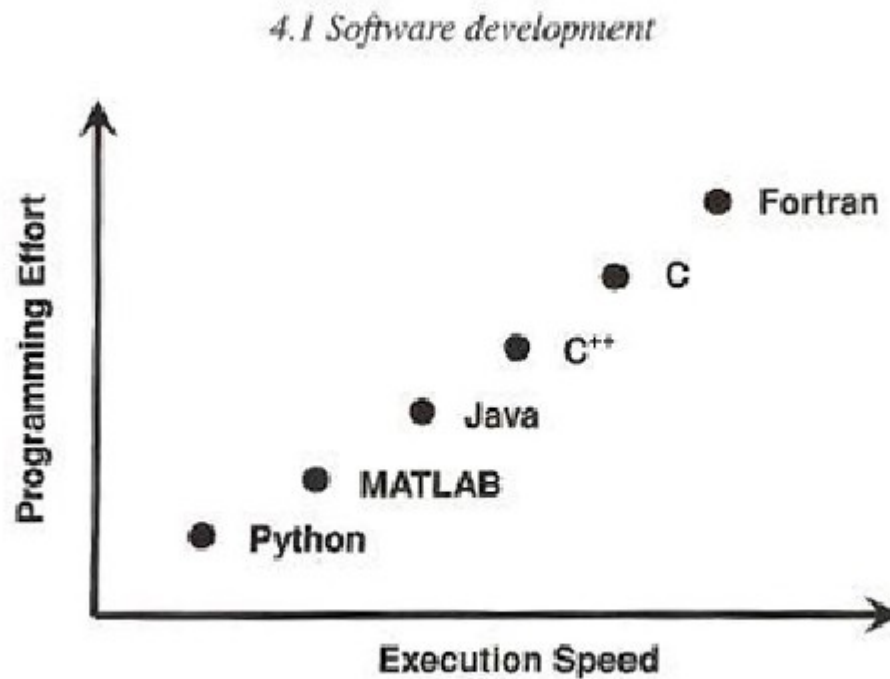


Figure 4.1 *Qualitative* example of programming effort versus execution speed (adapted from Wilson, 2009).

Alternativen

- Symbolische

- MAPLE

- maple(diff...)
 - War Teil von MatLab

- MATHEMATICA

- MuPAD

- Ist Teil von MatLAB

- Numerische

- MATHCAD

- Freeware

- GNU Octave
 - FreeMat

- MATLAB:

- Schwächer bei symbolischen Berechnungen
 - Überlegen bei numerischen
 - Verliert gegen MATHCAD in Ingenieurmathematik
 - Aber unzählige Erweiterungen (Toolbox)

Fazit

- Motivation des Herstellers:
 - <http://www.mathworks.de/videos/simulations-made-easy-with-simulink-68945.html?type=large>
- Erfunden für «Programmierlaien»
 - Optimal für Ingenieure
 - Schwaches Argument für Informatiker
- Bequemlichkeit auf Kosten der Performance
 - Schnelle Lösung oder langfristige Nutzung
- Sehr gute Visualisierungsmöglichkeiten
- Sehr großes Angebot an Literatur

Quellen

Literatur

- Benker H. *Ingenieurmathematik kompakt. Problemlösungen mit MATLAB* (Springer, 2010)(273s)
- Werner M. *Digitale Signalverarbeitung mit MATLAB* (Vieweg Teubner, 4th ed., 2008)(294s)
- Hoffmann J. *MATLAB und Simulink in Signalverarbeitung und Kommunikationstechnik* (492s)
- Benker H. *Differentialgleichungen mit Mathcad und Matlab* (Springer, 2005)(297s)
- Hoffmann J. *MATLAB und Simulink. Beispielorientierte einföhrung in die Simulation dynamischer Systeme* (Addison Wesley, 1998)(505s)
- Quarteroni A., Saleri F. *Wissenschaftliches Rechnen mit MATLAB* (Springer, 2006)(269s)

Abbildungen:

- Folie 13 : <http://www.mathworks.com/matlabcentral/fileexchange/35250-matlab-plot-gallery-wind/content/html/Wind.html>
- Folie 16 : Oberkampf W.L., Roy C.J. *Verification and Validation in Scientific Computing* (Cambridge University Press, 2010)