

(AB)²

Simulation



Entwicklung eines graphischen Editors
zur Modellierung von Systemen mit
dynamischer Modellstruktur

Diplomanden: **Andreas Bachmann, Andreas Butti**

Dozierende: **Prof. Dr. Stephan Scheidegger,
Dr. Rudolf Marcel Fuchsli**

Inhalt

1. Einleitung / Vorgaben
2. Modellierungseeditor
3. Simulation / Mathematik
4. Technische Hintergründe
5. Demonstration

1

Einleitung / Vorgaben

Einleitung / Vorgaben

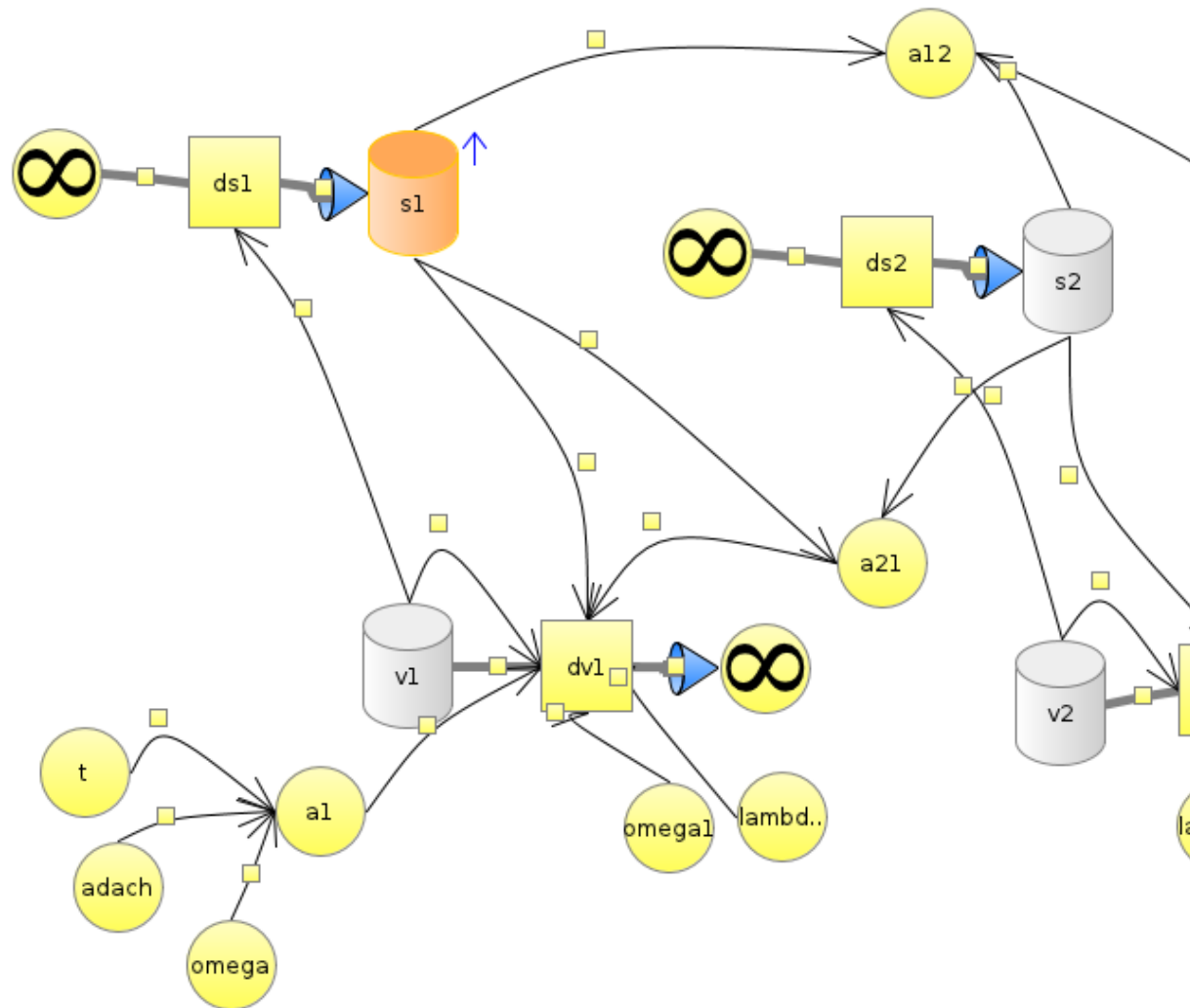
- Bereits existierende Tools
 - Berkeley Madonna
 - Simulink
- Benutzerführung
- Simulation im zweidimensionalen Raum
- Biologie / Physik, Zellen

2

Modellierungseeditor

2.1

Flussdiagramm



Eigenschaften

Name

Wert

Simulation

Simulation

Simulation Einstellungen

Numerisches Verfahren

Startzeit

Endzeit

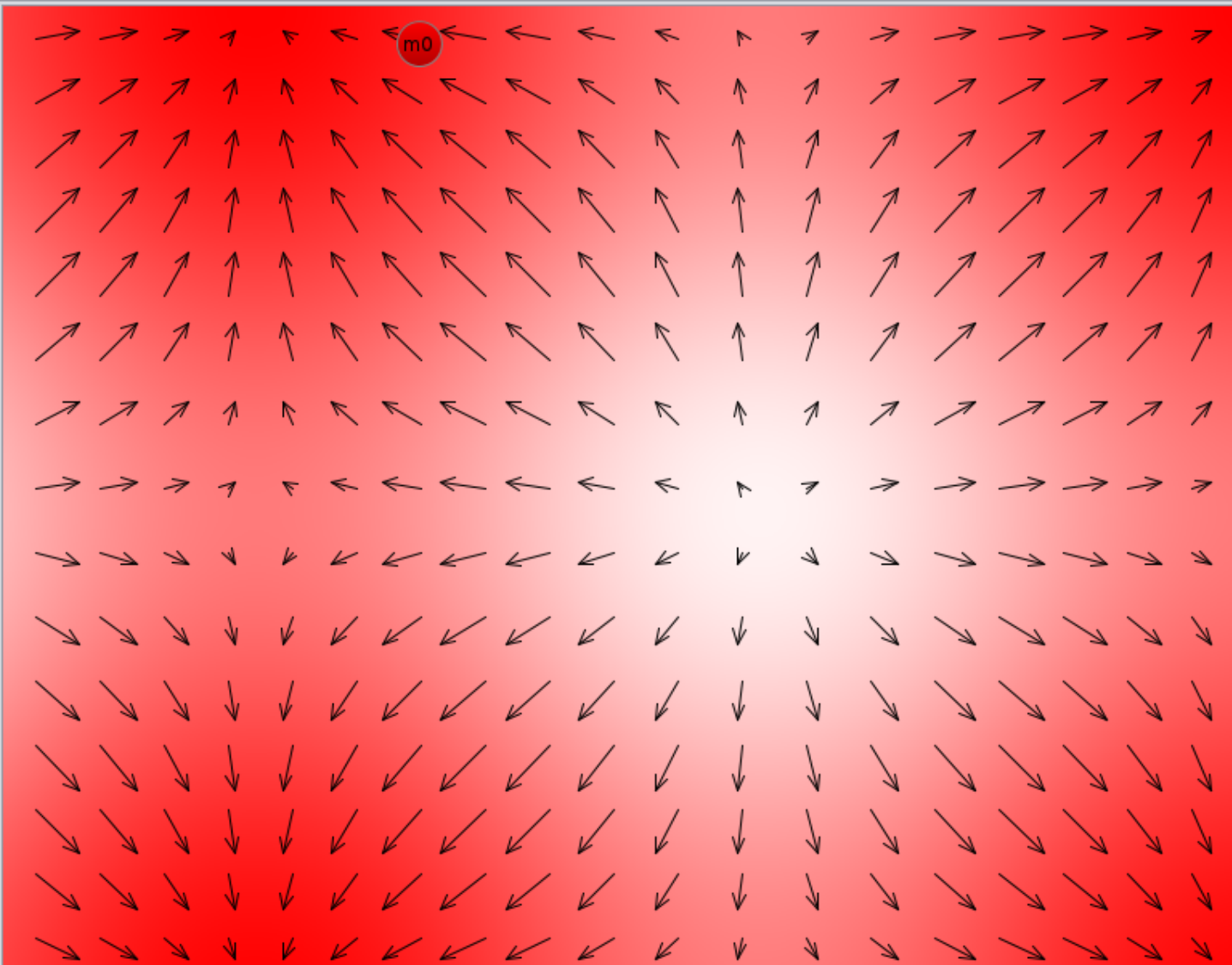
dt

Features

- Copy & Paste
- Exportieren als Vektorgrafik
- Undo / Redo
- Importieren bestehender Modelle
 - Berkeley Madonna
 - Dynasys

2.2

XY-Diagramm



Dichte

d1

+

-



2.1E2 0E0 1E1

☐ Logarithmisch

Simulation

Simulation

Matlab Compatible Simu...



Simulieren

Simulation Einstellungen

Numerisches Verfahren

Runge-Kutta 4

Startzeit

0

Endzeit

70

dt

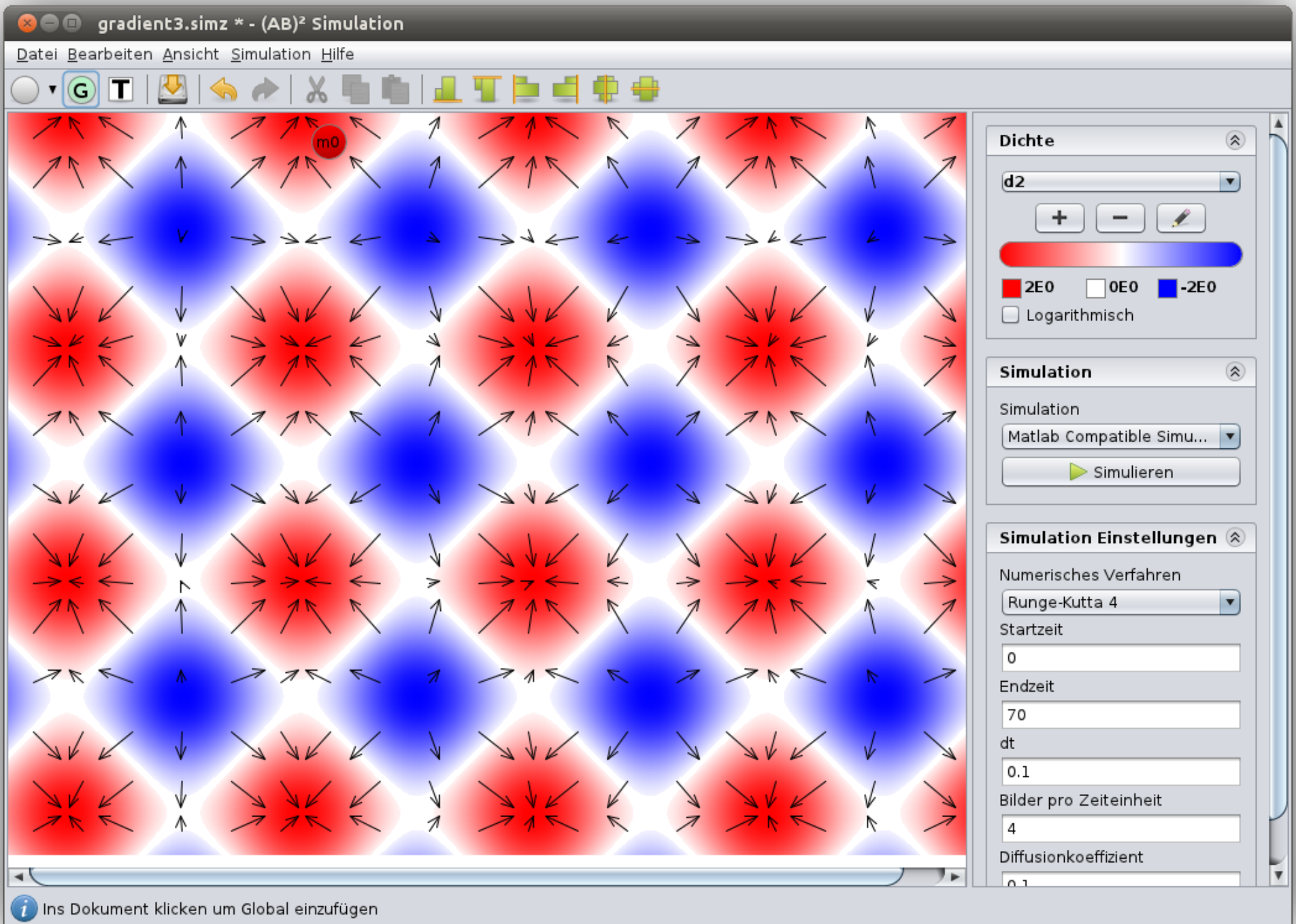
0.1

Bilder pro Zeiteinheit

4

Diffusionskoeffizient

0.1

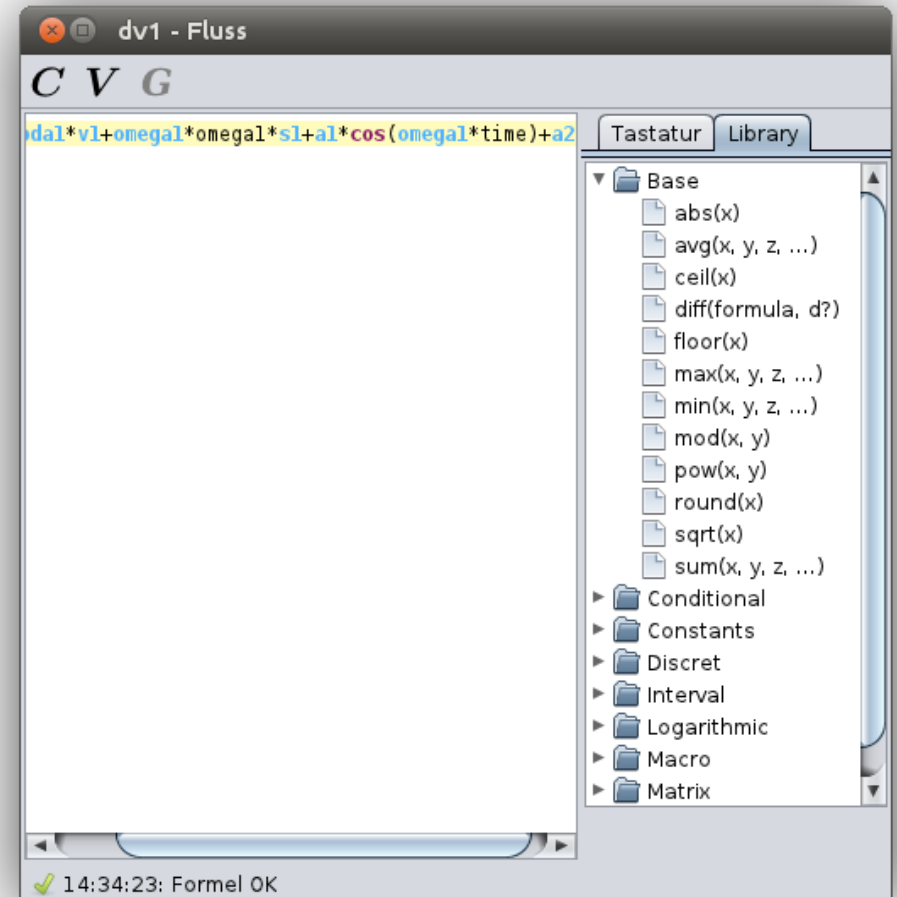
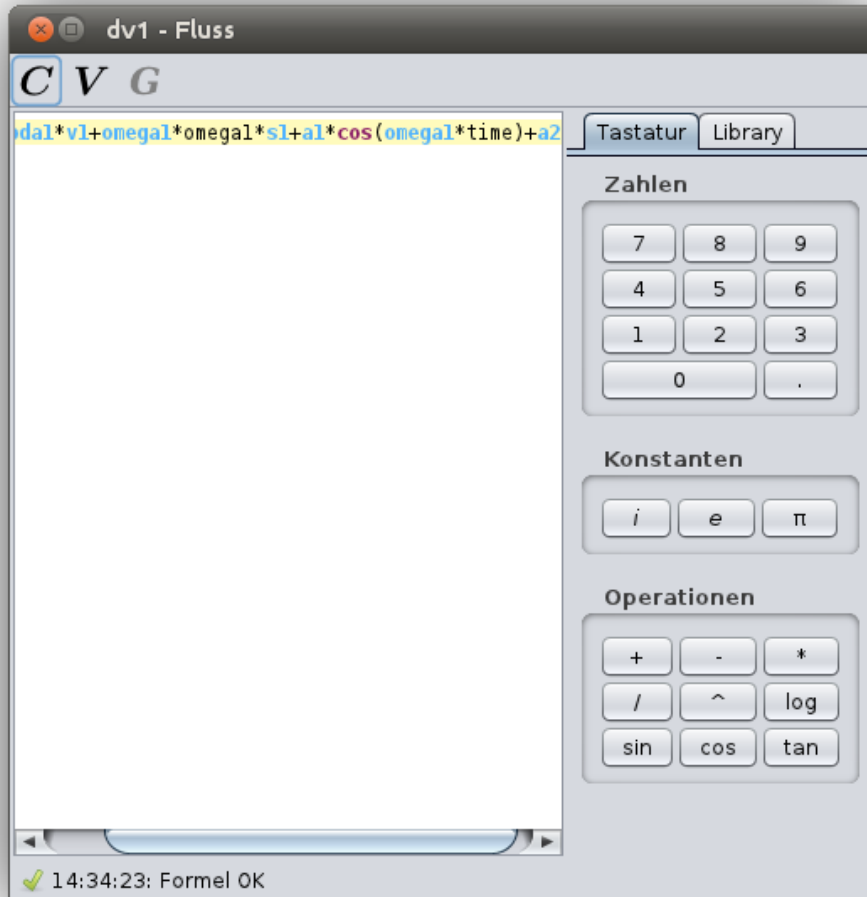


Features

- Dichten
 - Darstellung mit Gradientenpfeile
- Meso Kompartimente
 - Können gleiches Model beinhalten

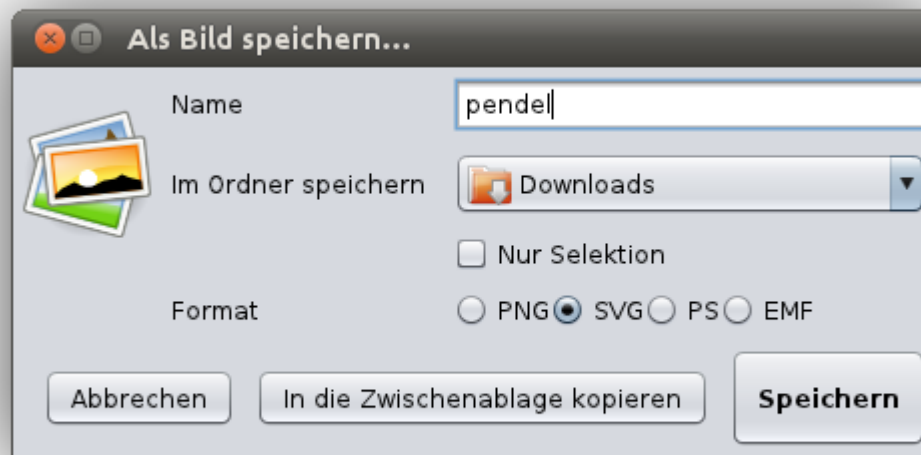
2.3

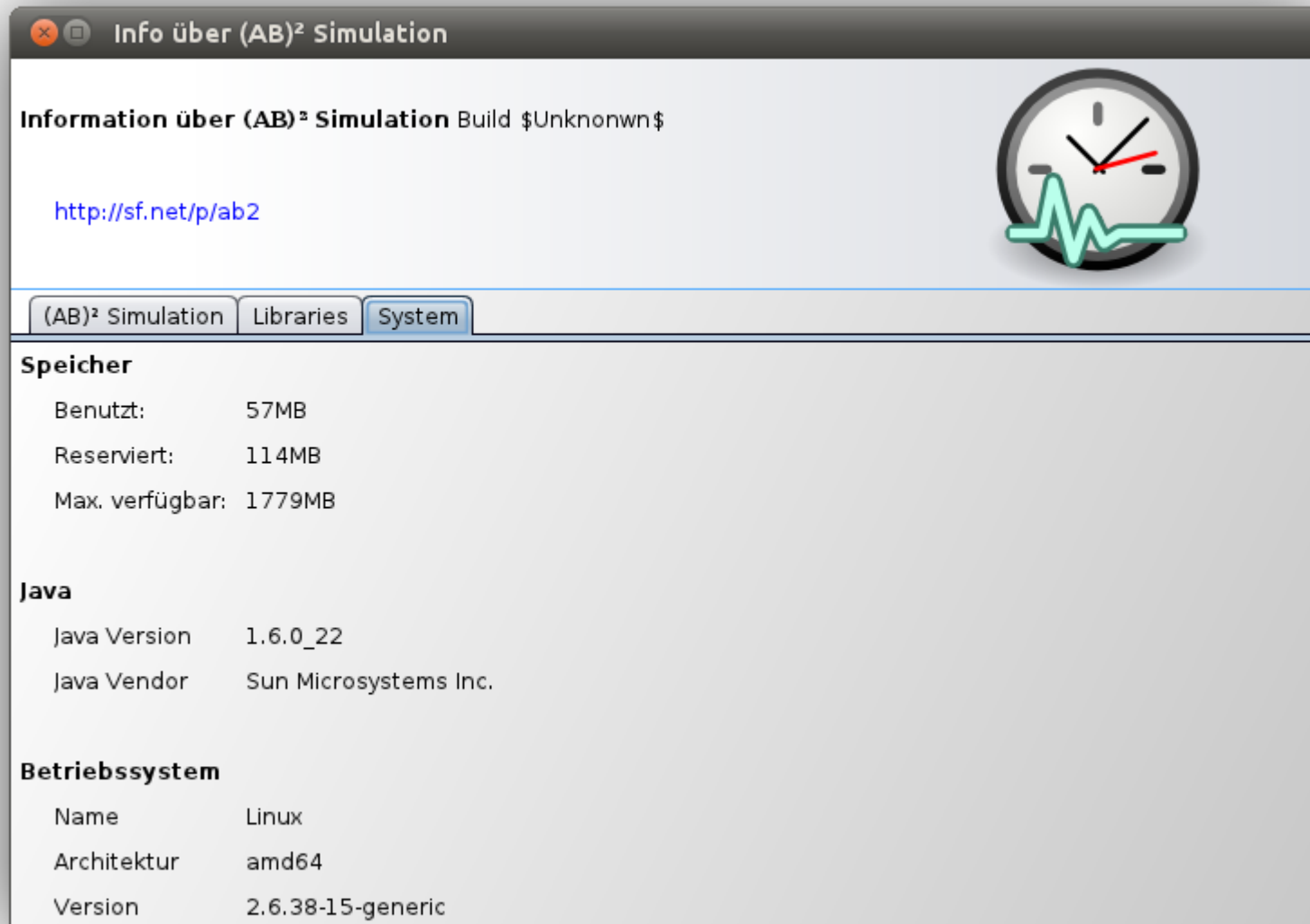
Gernerelle Guielemente

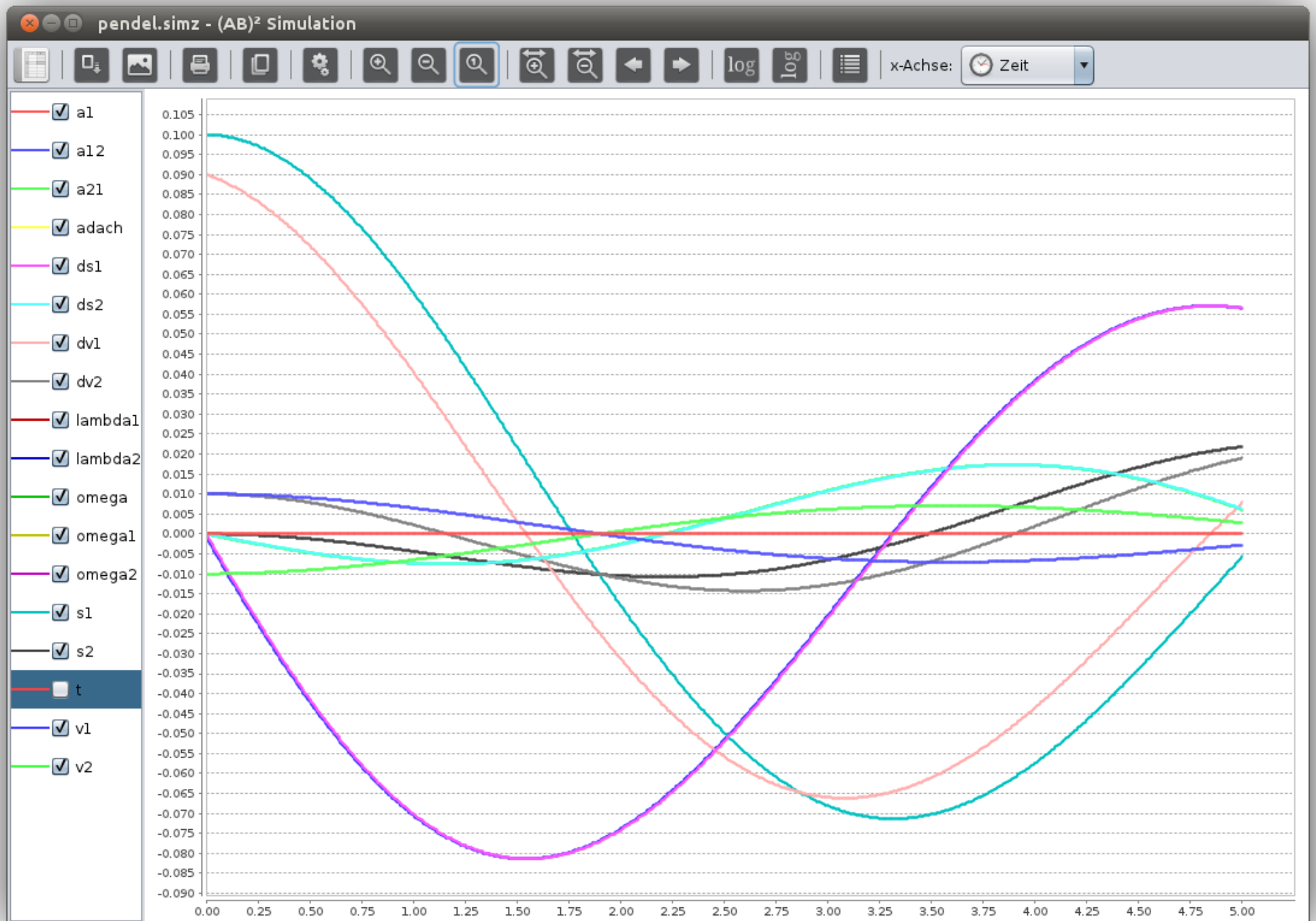


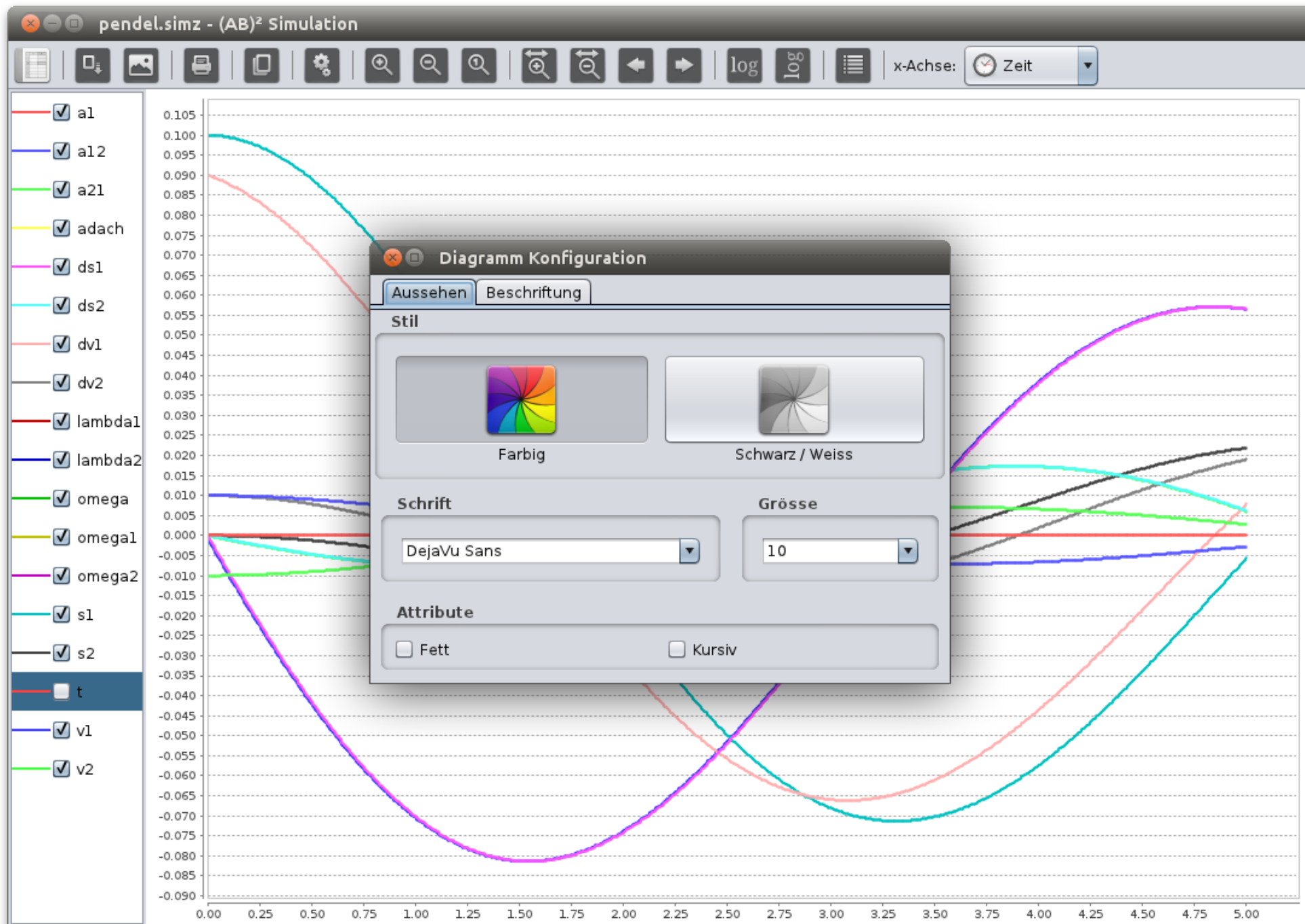
Formeeditor

- Syntaxhighlighting
- Autocomplete (für Lange Variablen)
- Menü mit Vorschlägen
- Library Browser
- Automatische Syntaxkontrolle









Simulationsdaten

Dezimalstellen: 6

| Zeit | a1 | a12 | a21 | adach | ds1 |
|----------|----------|----------|-----------|----------|-----------|
| 0.000000 | 0.000000 | 0.010000 | -0.010000 | 0.000000 | 0.000000 |
| 0.010000 | 0.000000 | 0.009999 | -0.009999 | 0.000000 | -0.000900 |
| 0.020000 | 0.000000 | 0.009998 | -0.009998 | 0.000000 | -0.001790 |
| 0.030000 | 0.000000 | 0.009995 | -0.009995 | 0.000000 | -0.002690 |
| 0.040000 | 0.000000 | 0.009992 | -0.009992 | 0.000000 | -0.003580 |
| 0.050000 | 0.000000 | 0.009988 | -0.009988 | 0.000000 | -0.004480 |
| 0.060000 | 0.000000 | 0.009983 | -0.009983 | 0.000000 | -0.005370 |
| 0.070000 | 0.000000 | 0.009978 | -0.009978 | 0.000000 | -0.006250 |
| 0.080000 | 0.000000 | 0.009971 | -0.009971 | 0.000000 | -0.007140 |
| 0.090000 | 0.000000 | 0.009964 | -0.009964 | 0.000000 | -0.008020 |
| 0.100000 | 0.000000 | 0.009956 | -0.009956 | 0.000000 | -0.008910 |
| 0.110000 | 0.000000 | 0.009948 | -0.009948 | 0.000000 | -0.009780 |
| 0.120000 | 0.000000 | 0.009938 | -0.009938 | 0.000000 | -0.010660 |
| 0.130000 | 0.000000 | 0.009928 | -0.009928 | 0.000000 | -0.011530 |
| 0.140000 | 0.000000 | 0.009917 | -0.009917 | 0.000000 | -0.012400 |
| 0.150000 | 0.000000 | 0.009905 | -0.009905 | 0.000000 | -0.013270 |
| 0.160000 | 0.000000 | 0.009893 | -0.009893 | 0.000000 | -0.014140 |
| 0.170000 | 0.000000 | 0.009879 | -0.009879 | 0.000000 | -0.015000 |
| 0.180000 | 0.000000 | 0.009865 | -0.009865 | 0.000000 | -0.015860 |
| 0.190000 | 0.000000 | 0.009850 | -0.009850 | 0.000000 | -0.016710 |
| 0.200000 | 0.000000 | 0.009835 | -0.009835 | 0.000000 | -0.017570 |
| 0.210000 | 0.000000 | 0.009818 | -0.009818 | 0.000000 | -0.018420 |
| 0.220000 | 0.000000 | 0.009801 | -0.009801 | 0.000000 | -0.019260 |
| 0.230000 | 0.000000 | 0.009783 | -0.009783 | 0.000000 | -0.020100 |

3

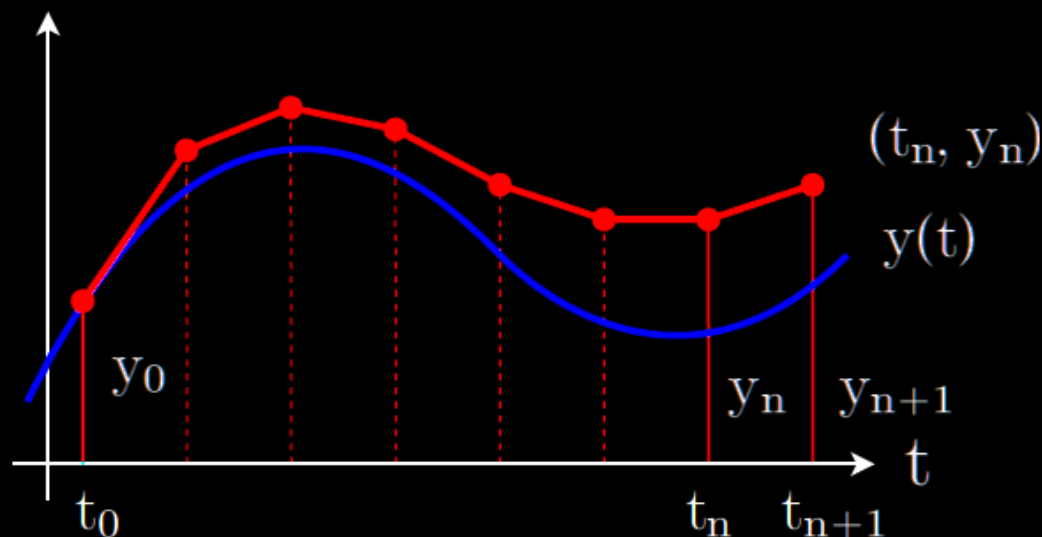
Simulation / Mathematik

Matlab (Markup Language)

- Vorteile:
 - Klare Schnittstelle nach aussen
 -
- Nachteile:
 - Eigener Prozess
 - Nicht beeinflussbar
 - (Fehler-)Ausgabe schwer zu erhalten

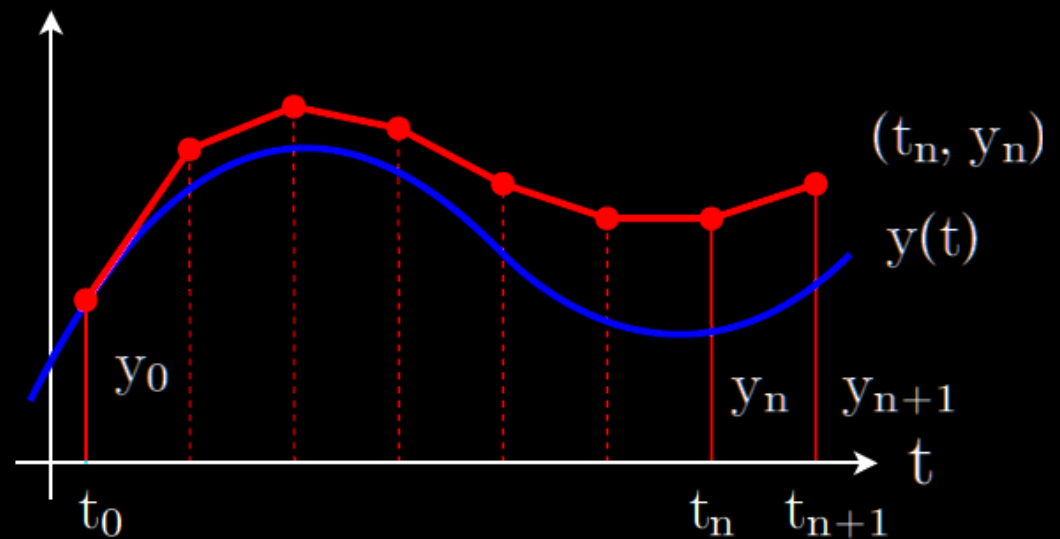
Simulations Verfahren (1)

- Gewöhnliche Differentialgleichungen (Ordinary Differential Equation ODE)
- Differential Equation Solver
- Numerisch lösen



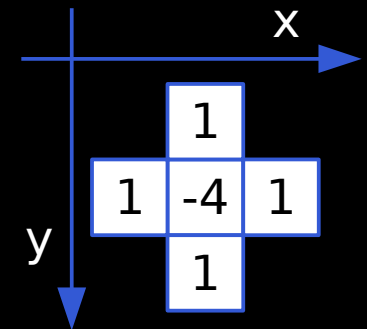
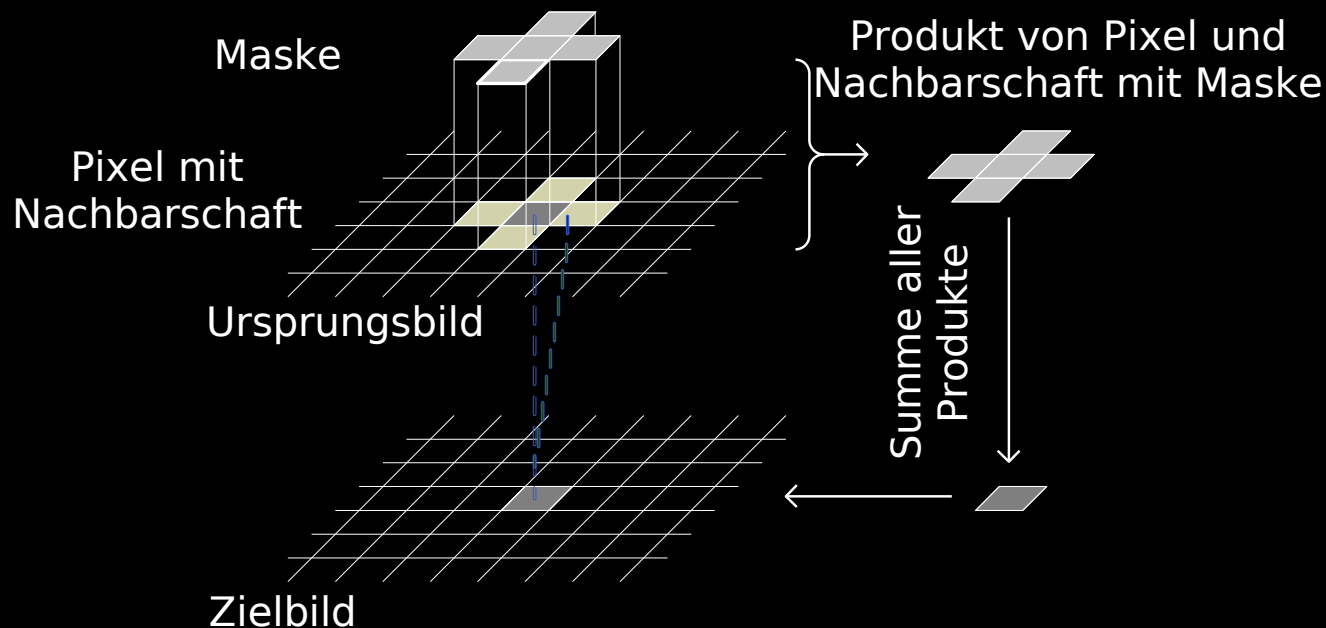
Simulations Verfahren (2)

- Matlab-Eigene Funktionen (ode45 u.ä.)
- Selbst Entwickelte Funktionen
 - Euler
 - Runge-Kutta
 - Klassisch
 - Dormand & Prince



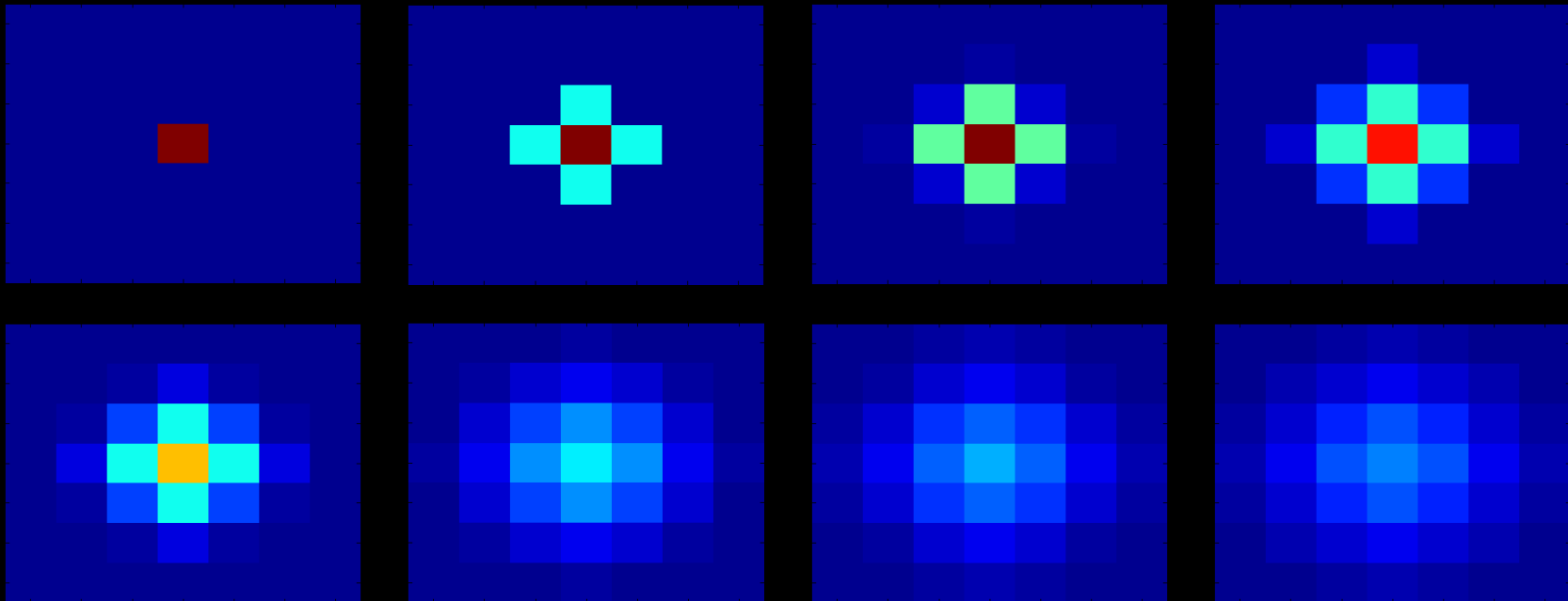
Diffusionsgleichung (1)

- Gleichmässige Verteilung/Ausbreitung im Raum
- Harte Kanten verhindern



Diffusionsgleichung (2)

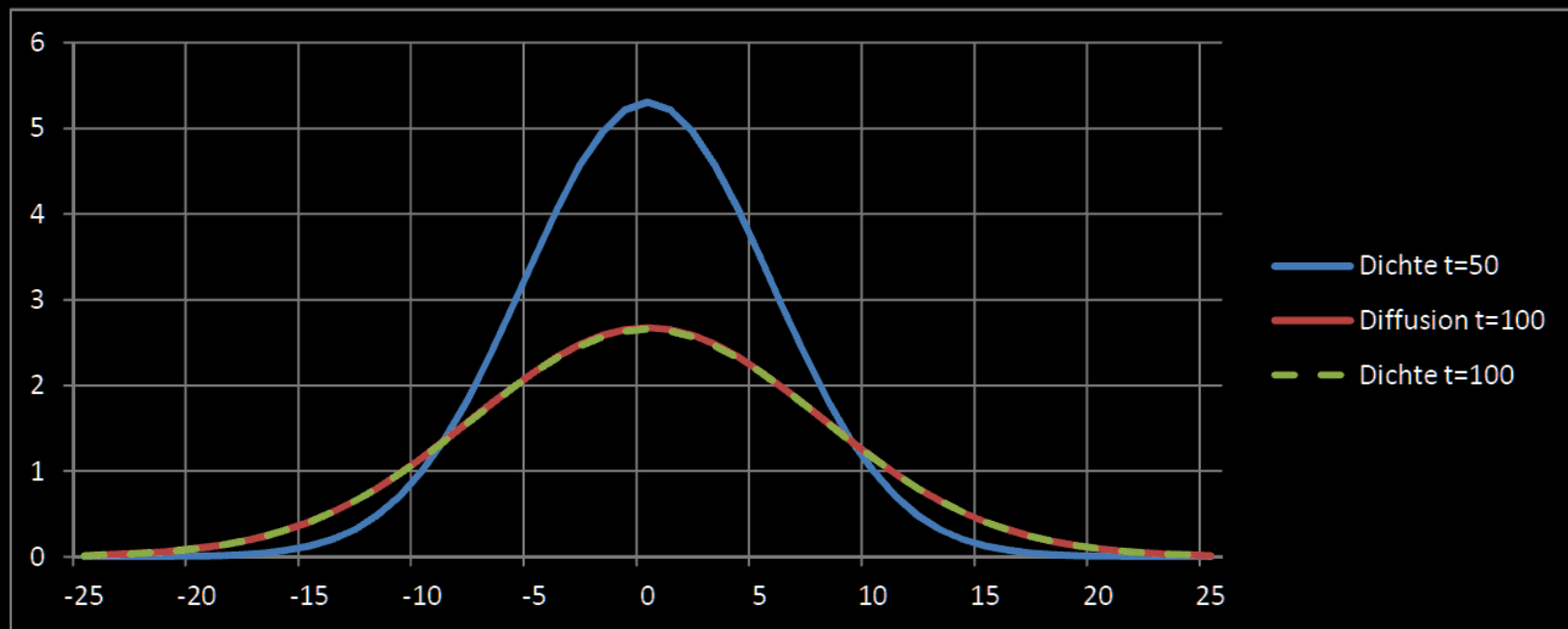
Resultat



Diffusionsgleichung (3)

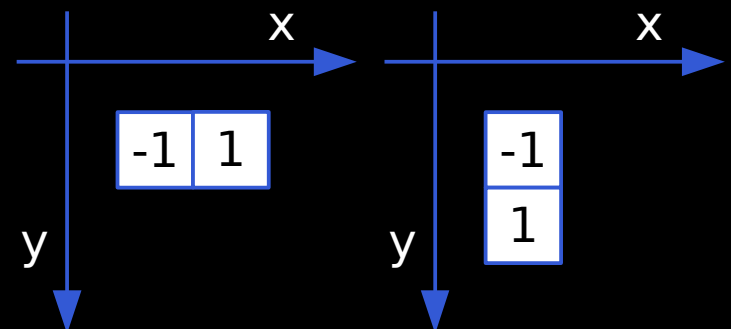
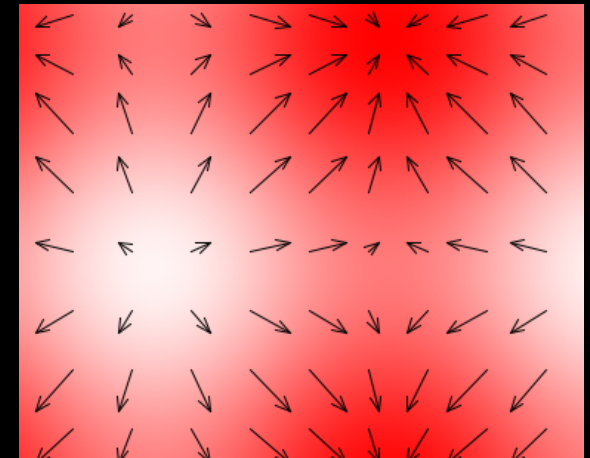
Verifikation: Dichtefunktion

$$G(\vec{r}, t) = \frac{1}{4\pi Dt} \cdot e^{-\frac{x^2+y^2}{4Dt}}$$



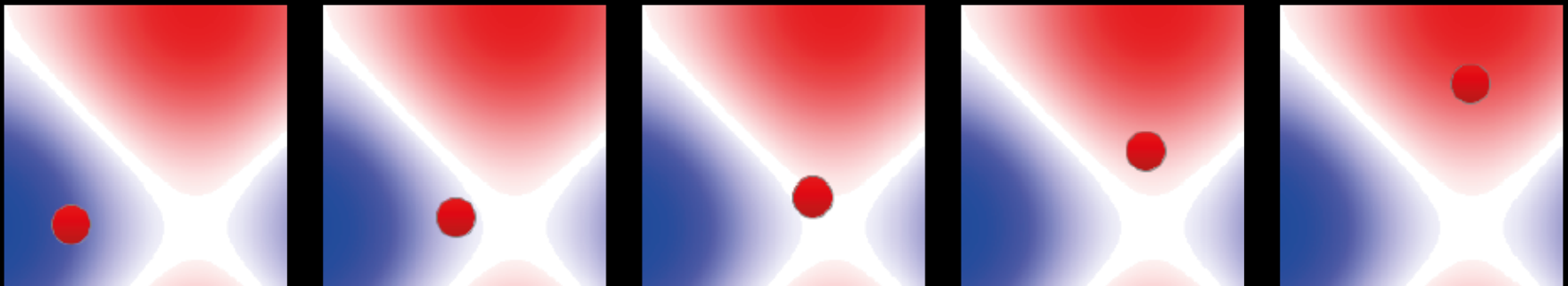
Gradienten-Verfahren

- Verfahren des steilsten Abstiegs
- Wandert schrittweise zum lokalen Minimum/Maximum



Gradienten-Verfahren

Resultat



4

Technische Hintergründe

Plattform unabhängig

- Java
- Setup für
 - Linux
 - Mac
 - Windows
- Portable Version



Modularität

- ~20 Projekte
- Erweiterbarkeit
- Keine Redundanzen
- Klare Grenzen / Schnittstellen

Plugins / Erweiterbarkeit

- Simpel (200 Zeilen Code)
- Importieren von Fremdformaten
- Simulation
- Weitere Schnittstellen denkbar

Fehlerhandling

- Alle erwarteten Fehler gehandelt
 - Meldung an Benutzer, ggf. mit Abhilfe
 - Datei nicht Schreibbar
 - Eingabe falsch etc.
- Alle unerwarteten Fehler / Eventloop Exceptions
 - Fehler Benutzer anzeigen / loggen
 - Programmierfehler
 - Manipulierte Daten etc.

Erweiterungen nach Abgabe

- Aussehen für Mac OS X angepasst
- Neue Legende für Dichte
- Dichte exportieren bei „Bild exportieren“
- Exportieren als Film
- Gradientenpfeile
- Raster für Flussdiagramm
- Nativer Filechooser für Mac, Linux, Win
- „Nur Selektion“ für Bildexport
- Berkeley Madonna XML Files importieren
- Funktionen Library

Fehlerkorrekturen nach Abgabe

- Copy & Paste bei Meso wird nun Formel für X und Y korrekt kopiert
- XYModell Autorparser funktioniert nun
- DensityContainer löschen funktioniert
- Div. Korrekturen am Buildsystem
- Repaintproblem bei Flussverbindungen
- Aboutdialog mit Tabs / kleine Bildschirme
- Prüfung Modell verbessert
- Formeleditor zeigt „Initialw.“ im Titel etc.
- CSV Export mit NaN funktioniert
- Parser erweitert z.B. PULSE und STEP

5

Demonstration

Ende

Sind noch Fragen?