$(AB)^2$ 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 Füchslin

zh aw

Inhalt

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

1

Einleitung / Vorgaben

Einleitung / Vorgaben

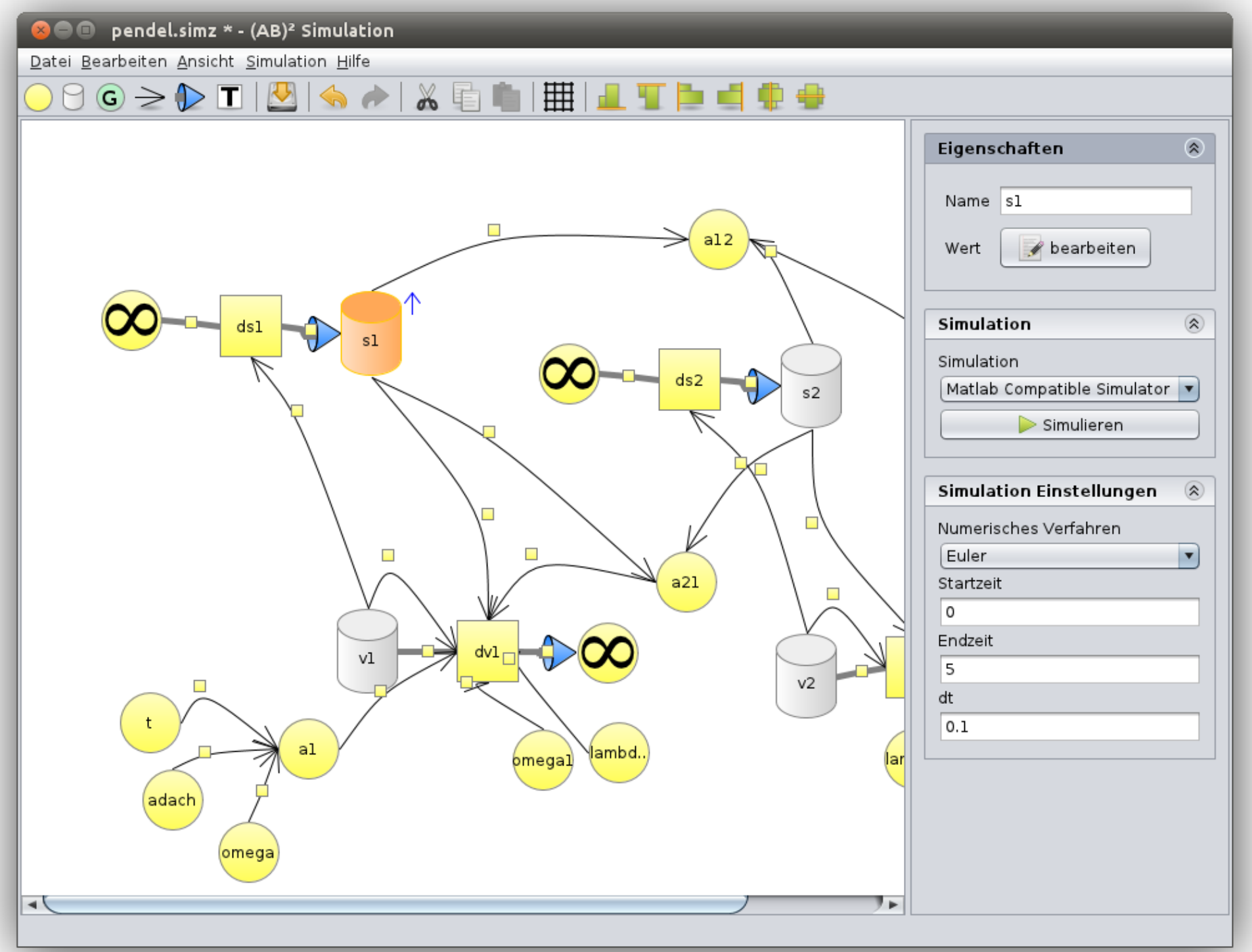
- Bereits existierende Tools
  - Berkeley Madonna
  - Simulink
- Benutzerführung
- Simulation im zweidimensionalen Raum
- Biologie / Physik, Zellen

2

Modellierungseditor

2.1

Flussdiagramm

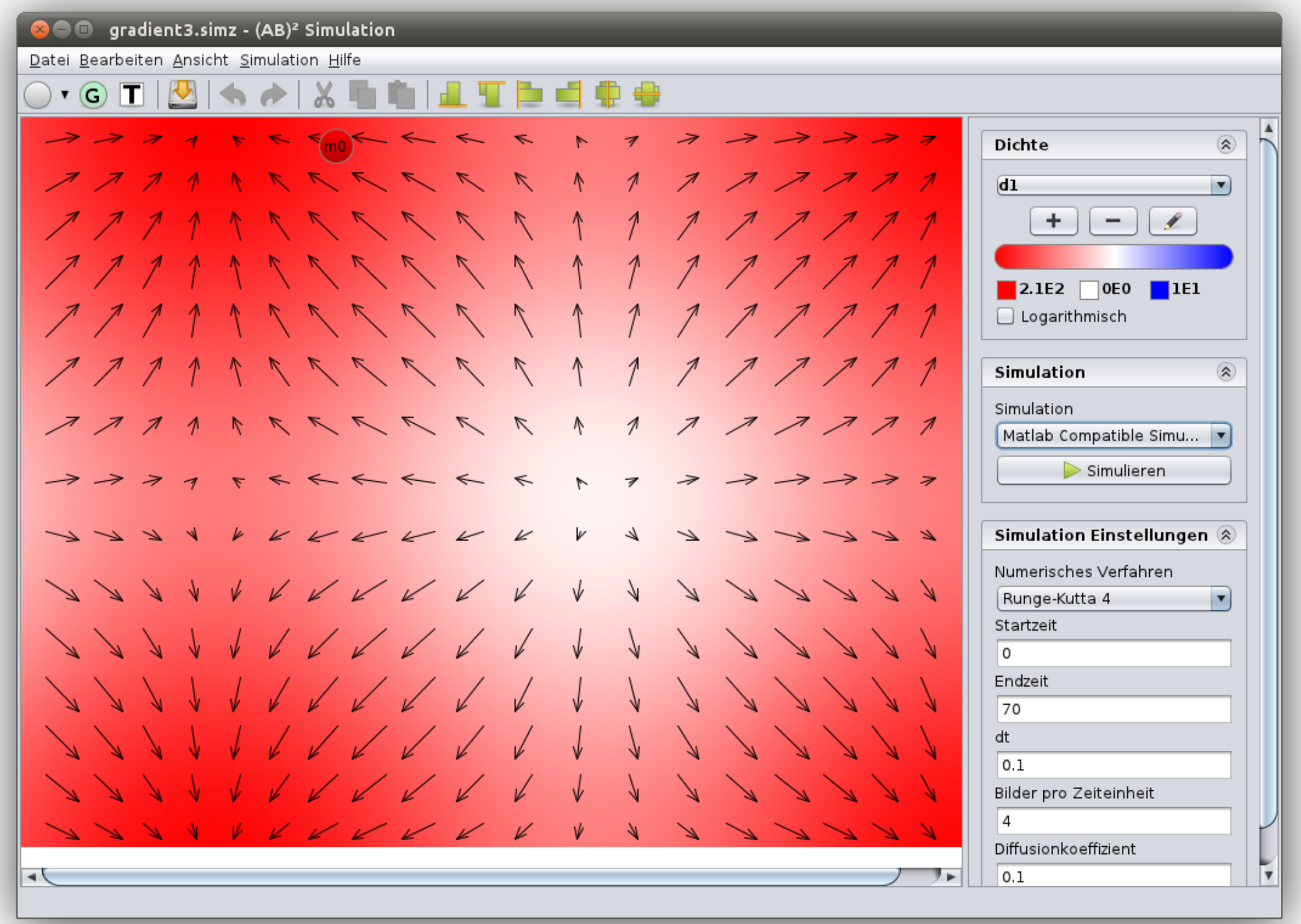

pendel.simz * - (AB)² Simulation
Datei Bearbeiten Ansicht Simulation Hilfe
ds1
s1
a12
ds2
s2
a21
v1
dv1
v2
t
a1
adach
omega
omega1
lambd..
Eigenschaften
Name
s1
Wert
bearbeiten
Simulation
Simulation
Matlab Compatible Simulator
Simulieren
Simulation Einstellungen
Numerisches Verfahren
Euler
Startzeit
0
Endzeit
5
dt
0.1

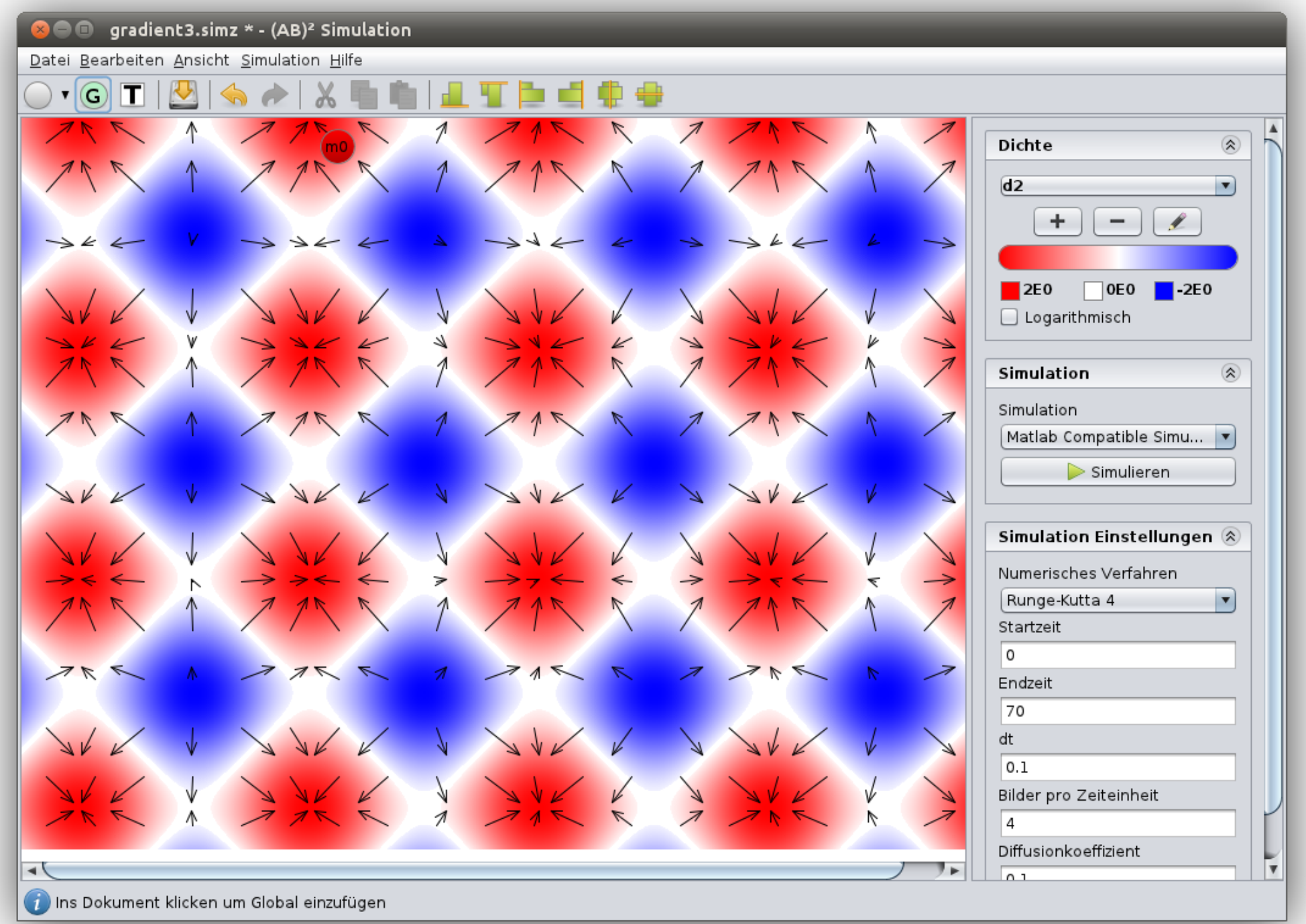
Features

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

2.2

XY-Diagramm


gradient3.simz - (AB)² Simulation
Datei Bearbeiten Ansicht Simulation Hilfe
m0
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
Diffusionkoeffizient
0.1

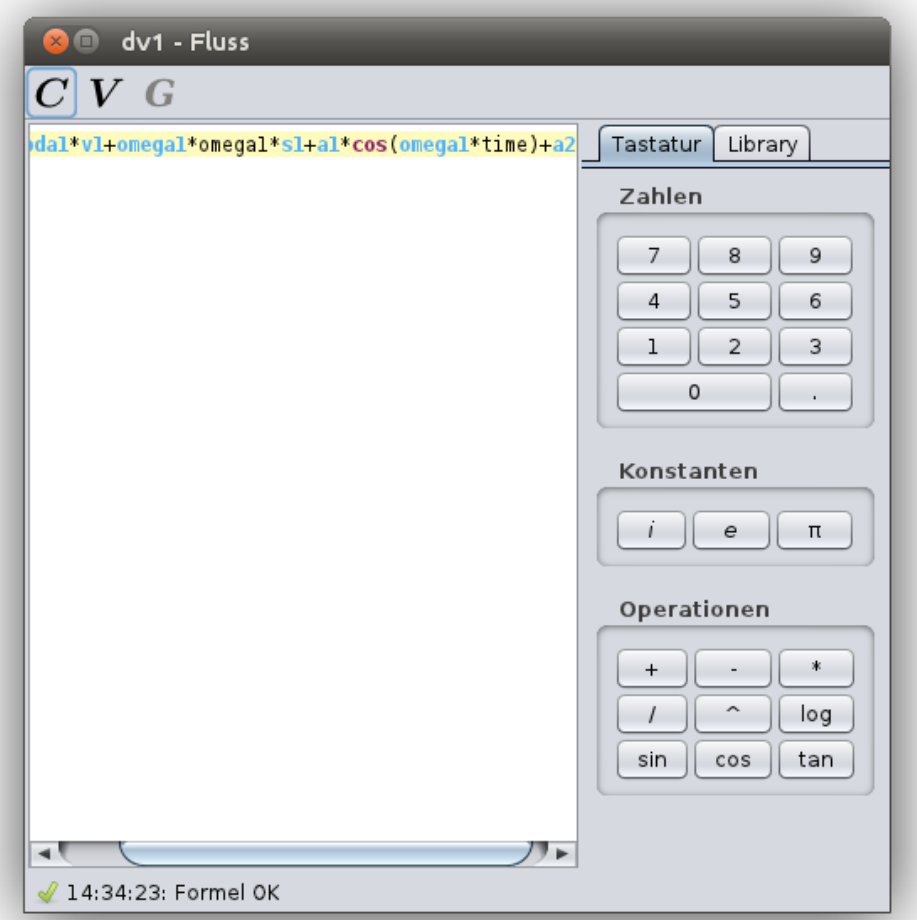

gradient3.simz * - (AB)² Simulation
Datei Bearbeiten Ansicht Simulation Hilfe
m0
Dichte
d2
2E0
0E0
-2E0
Logarithmisch
Simulation
Simulation
Matlab Compatible Simu...
Simulieren
Simulation Einstellungen
Numerisches Verfahren
Runge-Kutta 4
Startzeit
0
Endzeit
70
dt
0.1
Bilder pro Zeiteinheit
4
Diffusionkoeffizient
Ins Dokument klicken um Global einzufügen

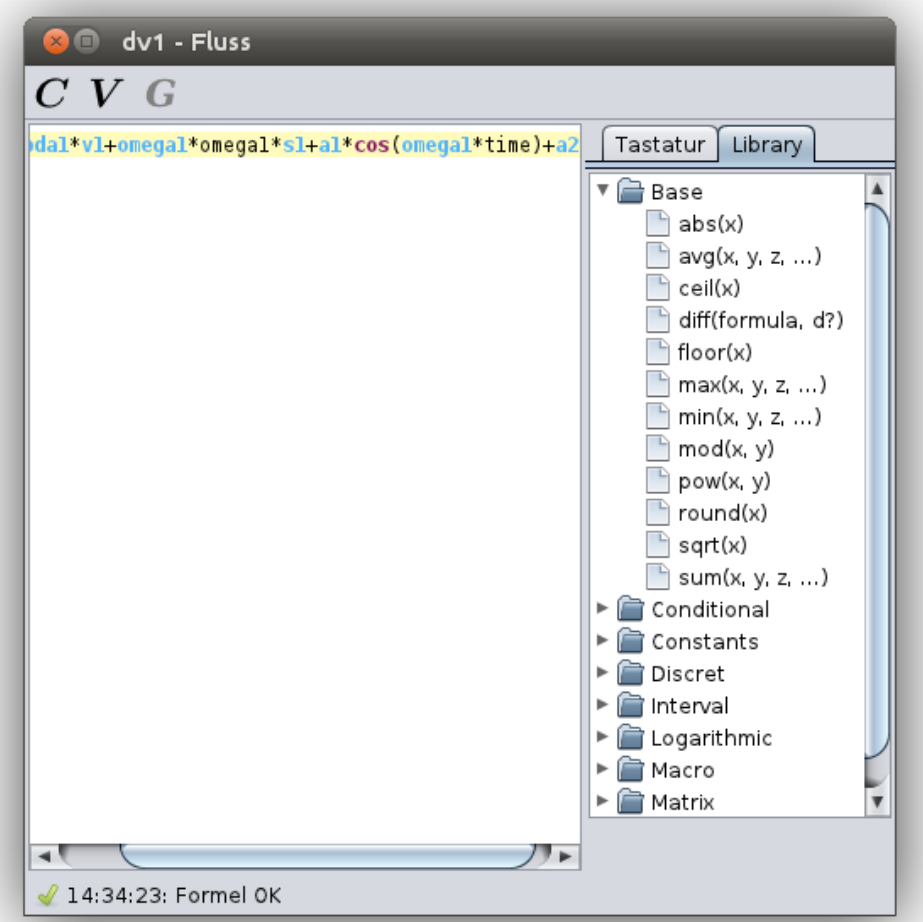
Features

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

2.3

Gernerelle Guielemente

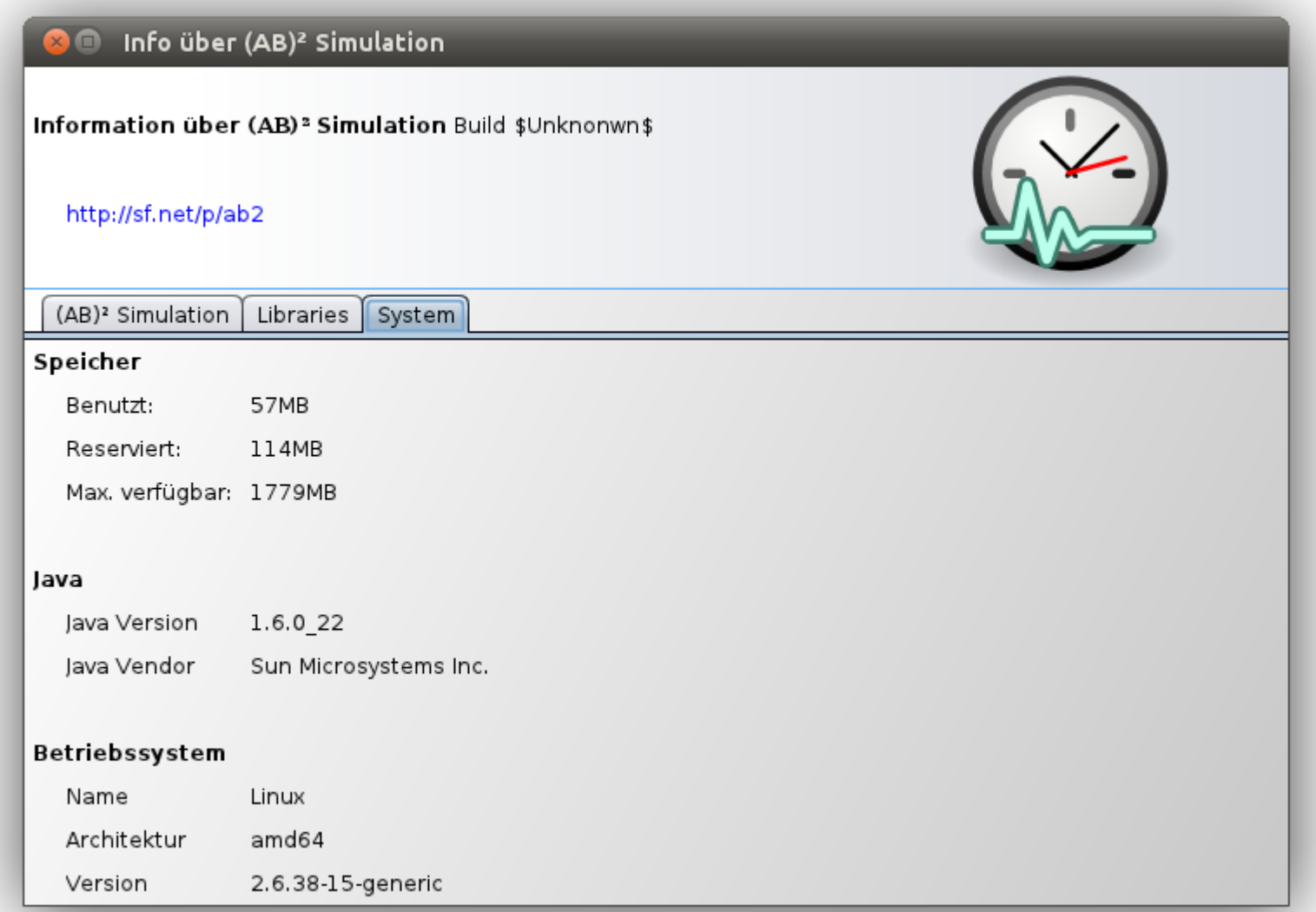

dv1 - Fluss
C V G
dal*vl+omegal*omegal*sl+al*cos(omegal*time)+a2
Tastatur
Library
Zahlen
7 8 9
4 5 6
1 2 3
0 .
Konstanten
i e π
Operationen
+ - *
/ ^ log
sin cos tan
14:34:23: Formel OK

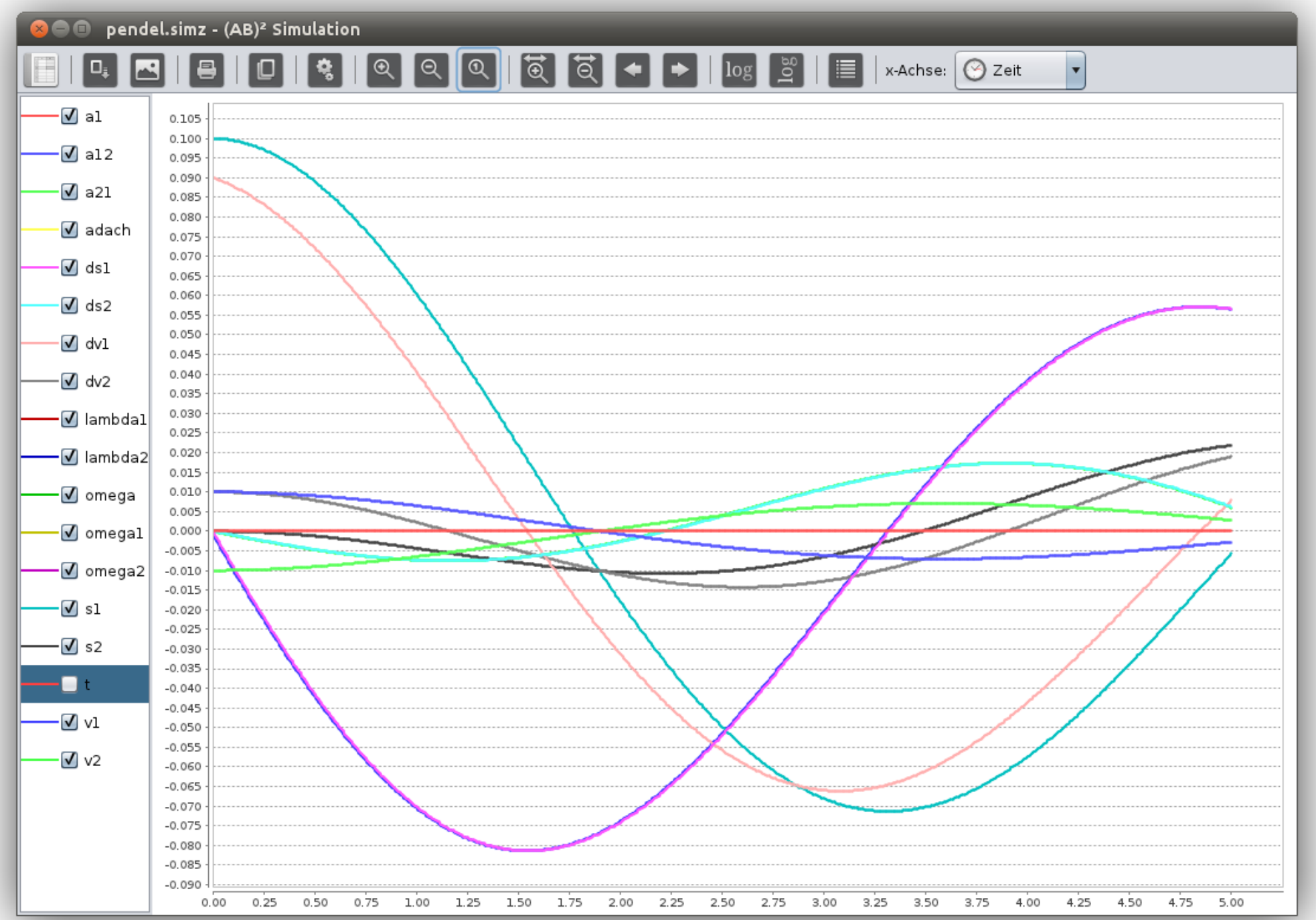

dv1 - Fluss
C V G
dal*vl+omegal*omegal*sl+al*cos(omegal*time)+a2
Tastatur
Library
Base
abs(x)
avg(x, y, z, ...)
ceil(x)
diff(formula, d?)
floor(x)
max(x, y, z, ...)
min(x, y, z, ...)
mod(x, y)
pow(x, y)
round(x)
sqrt(x)
sum(x, y, z, ...)
Conditional
Constants
Discret
Interval
Logarithmic
Macro
Matrix
14:34:23: Formel OK

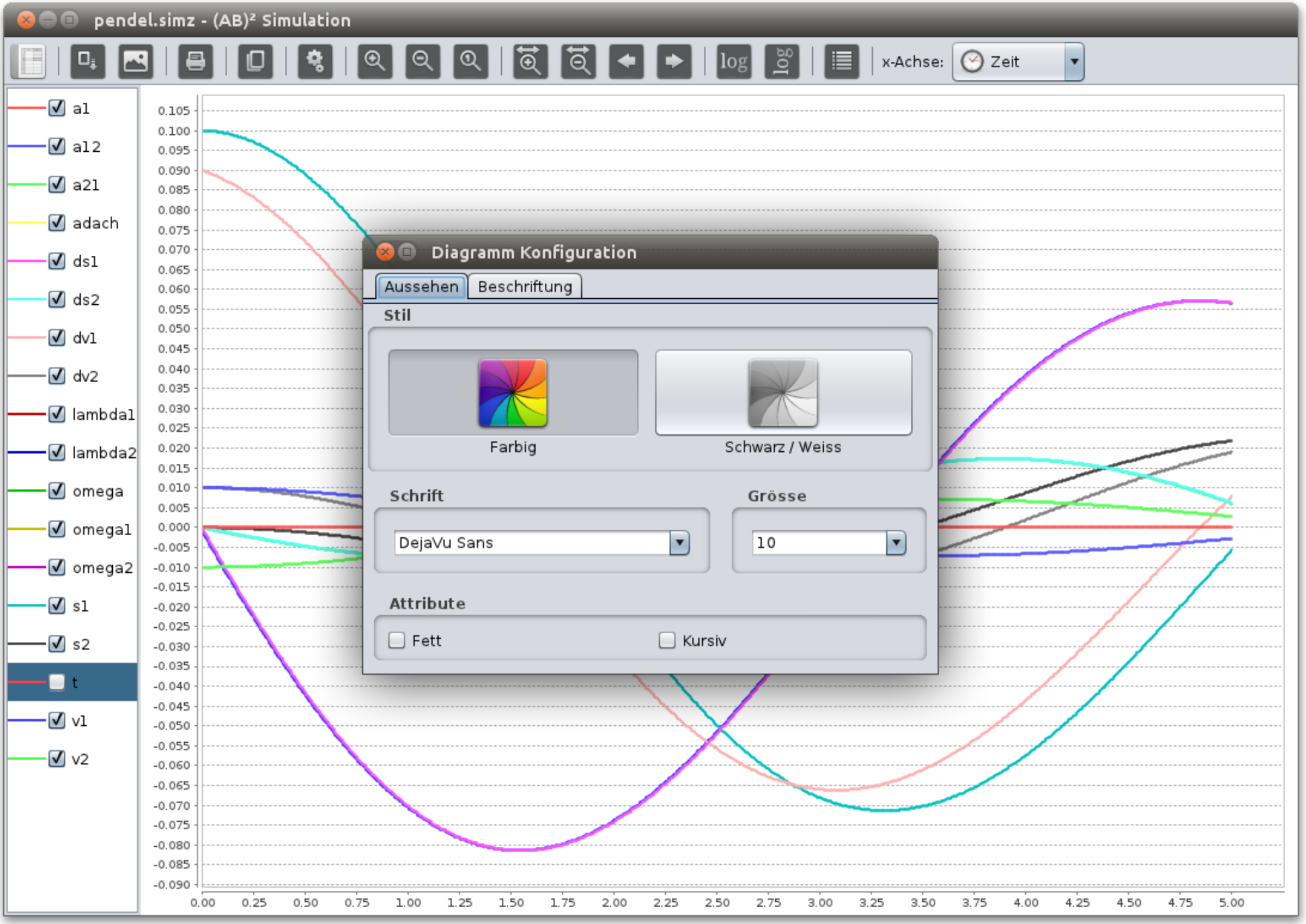
Formeleditor

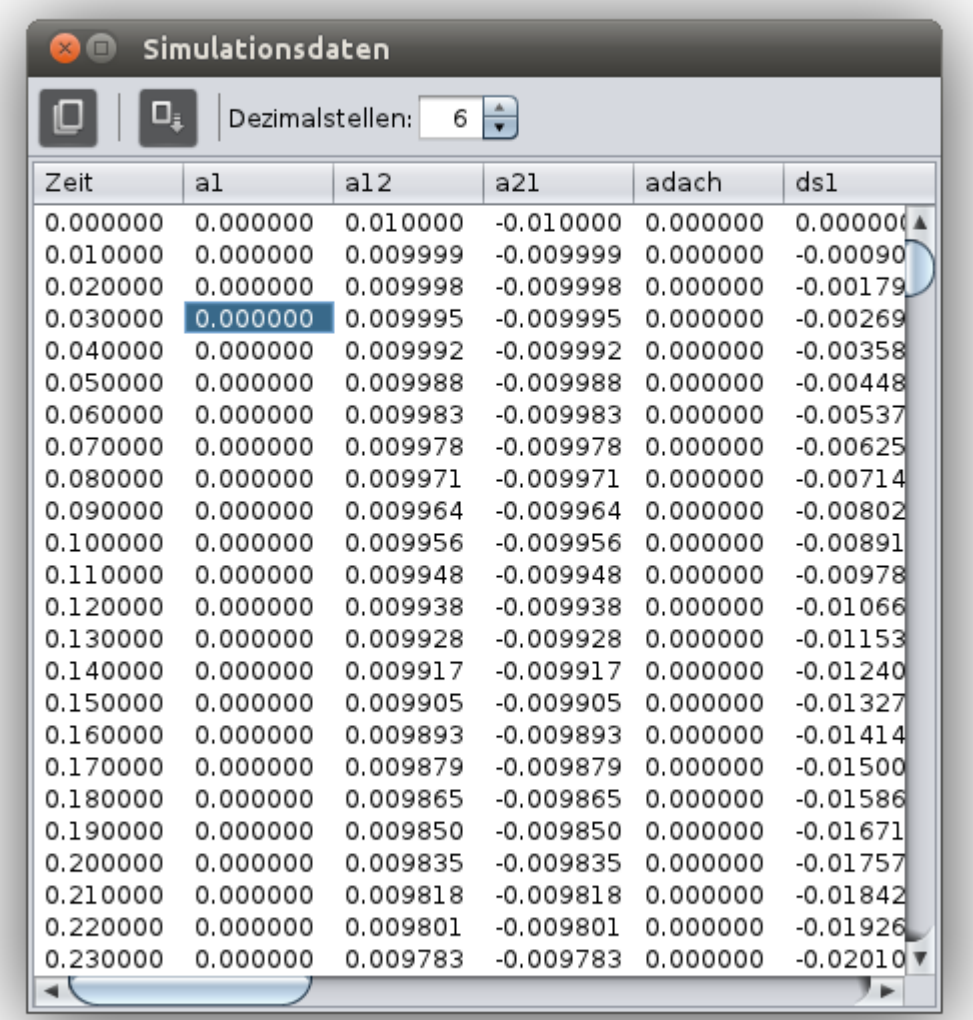
- Syntaxhilighting
- Autocomplete (für Lange Variablen)
- Menü mit Vorschlägen
- Library Browser
- Automatische Syntaxkontrolle


Als Bild speichern...
Name
pendel
Im Ordner speichern
Downloads
Nur Selektion
Format
PNG
SVG
PS
EMF
Abbrechen
In die Zwischenablage kopieren
Speichern


Info über (AB)² Simulation
Information über (AB)² Simulation Build $Unknonwn$
http://sf.net/p/ab2
(AB)² Simulation
Libraries
System
Speicher
Benutzt: 57MB
Reserviert: 114MB
Max. verfügbar: 1779MB
Java
Java Version 1.6.0_22
Java Vendor Sun Microsystems Inc.
Betriebssystem
Name Linux
Architektur amd64
Version 2.6.38-15-generic


pendel.simz - (AB)² Simulation
x-Achse:
Zeit
log
log
a1
a12
a21
adach
ds1
ds2
dv1
dv2
lambda1
lambda2
omega
omega1
omega2
s1
s2
t
v1
v2
0.105
0.100
0.095
0.090
0.085
0.080
0.075
0.070
0.065
0.060
0.055
0.050
0.045
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
-0.005
-0.010
-0.015
-0.020
-0.025
-0.030
-0.035
-0.040
-0.045
-0.050
-0.055
-0.060
-0.065
-0.070
-0.075
-0.080
-0.085
-0.090
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00
3.25
3.50
3.75
4.00
4.25
4.50
4.75
5.00


pendel.simz - (AB)² Simulation
x-Achse:
Zeit
a1
a12
a21
adach
ds1
ds2
dv1
dv2
lambda1
lambda2
omega
omega1
omega2
s1
s2
t
v1
v2
Diagramm Konfiguration
Aussehen
Beschriftung
Stil
Farbig
Schwarz / Weiss
Schrift
DejaVu Sans
Grösse
10
Attribute
Fett
Kursiv



Zeit	a1	a12	a21	adach	ds1
0.000000	0.000000	0.010000	-0.010000	0.000000	0.00000(
0.010000	0.000000	0.009999	-0.009999	0.000000	-0.00090
0.020000	0.000000	0.009998	-0.009998	0.000000	-0.00179
0.030000	0.000000	0.009995	-0.009995	0.000000	-0.00269
0.040000	0.000000	0.009992	-0.009992	0.000000	-0.00358
0.050000	0.000000	0.009988	-0.009988	0.000000	-0.00448
0.060000	0.000000	0.009983	-0.009983	0.000000	-0.00537
0.070000	0.000000	0.009978	-0.009978	0.000000	-0.00625
0.080000	0.000000	0.009971	-0.009971	0.000000	-0.00714
0.090000	0.000000	0.009964	-0.009964	0.000000	-0.00802
0.100000	0.000000	0.009956	-0.009956	0.000000	-0.00891
0.110000	0.000000	0.009948	-0.009948	0.000000	-0.00978
0.120000	0.000000	0.009938	-0.009938	0.000000	-0.01066
0.130000	0.000000	0.009928	-0.009928	0.000000	-0.01153
0.140000	0.000000	0.009917	-0.009917	0.000000	-0.01240
0.150000	0.000000	0.009905	-0.009905	0.000000	-0.01327
0.160000	0.000000	0.009893	-0.009893	0.000000	-0.01414
0.170000	0.000000	0.009879	-0.009879	0.000000	-0.01500
0.180000	0.000000	0.009865	-0.009865	0.000000	-0.01586
0.190000	0.000000	0.009850	-0.009850	0.000000	-0.01671
0.200000	0.000000	0.009835	-0.009835	0.000000	-0.01757
0.210000	0.000000	0.009818	-0.009818	0.000000	-0.01842
0.220000	0.000000	0.009801	-0.009801	0.000000	-0.01926
0.230000	0.000000	0.009783	-0.009783	0.000000	-0.02010

3

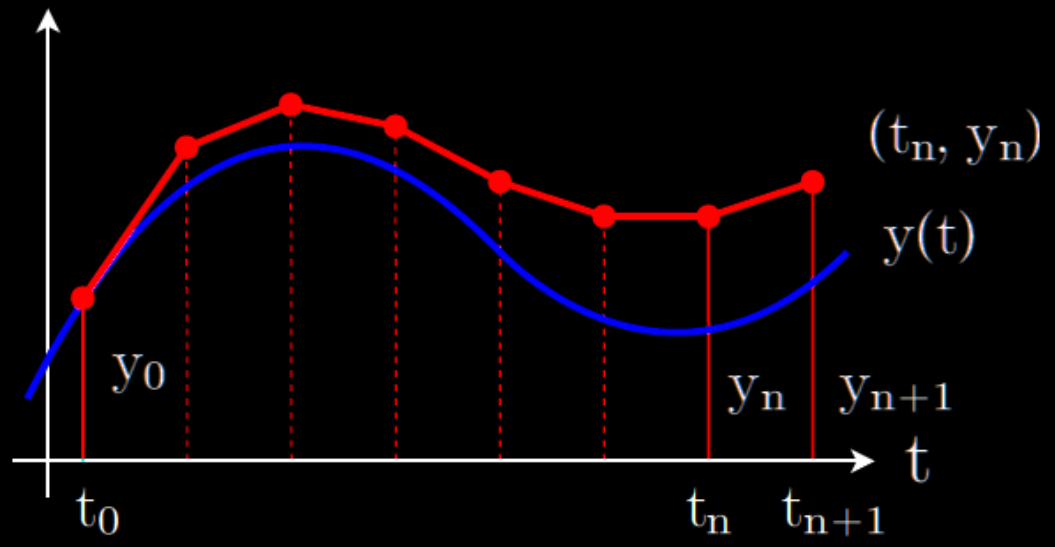
Simulation / Mathematik

Matlab (Markup Language)

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

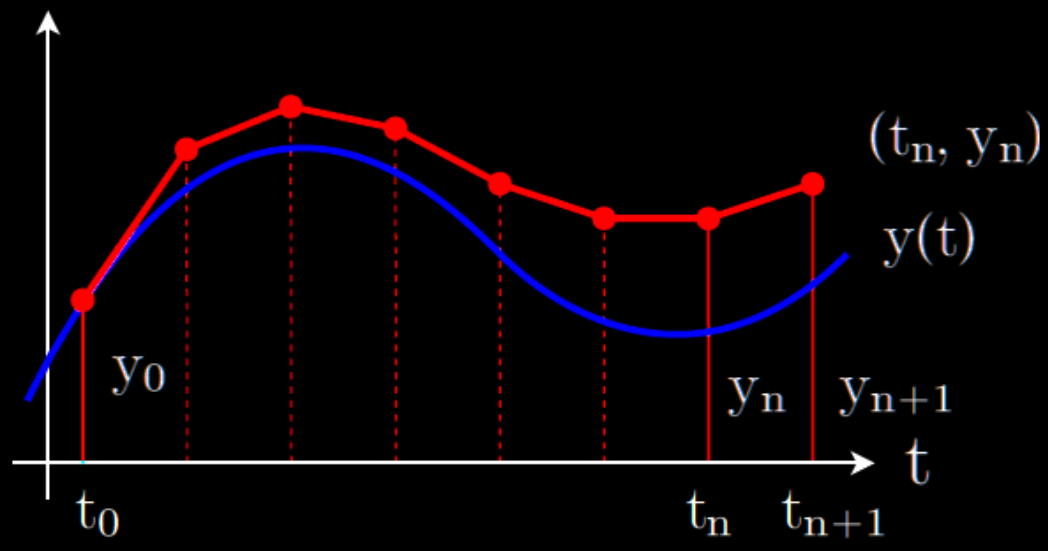
Simulations Verfahren (1)

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



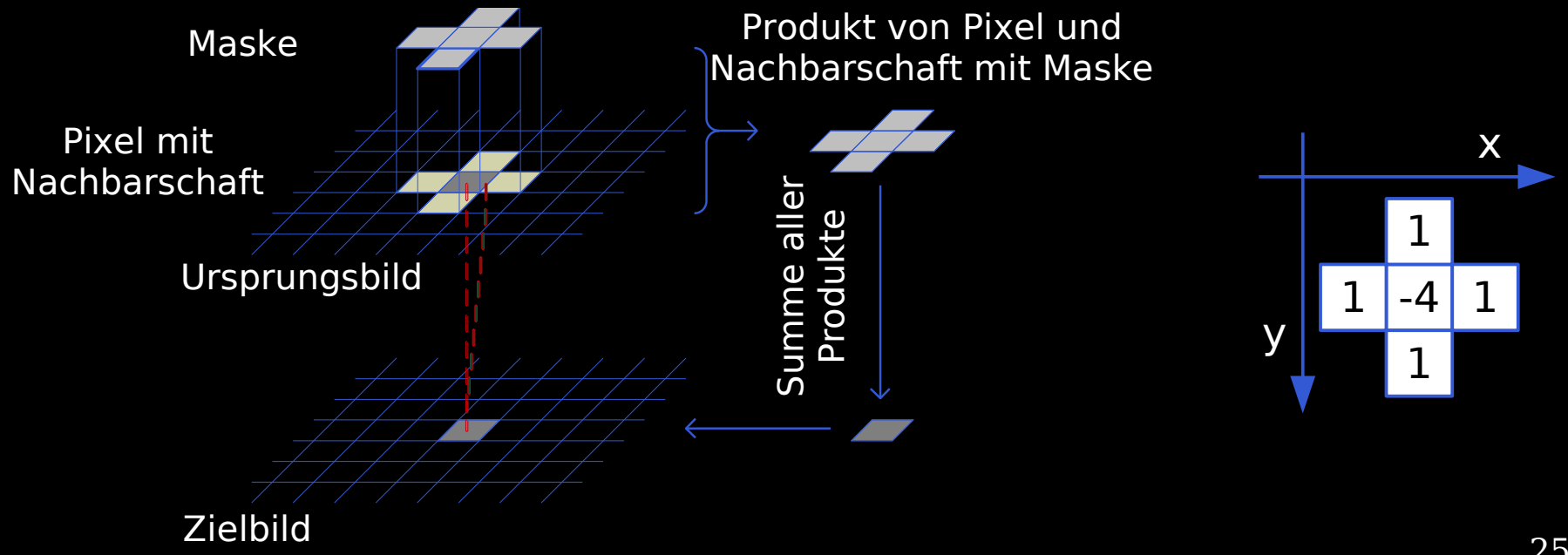
Simulations Verfahren (2)

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



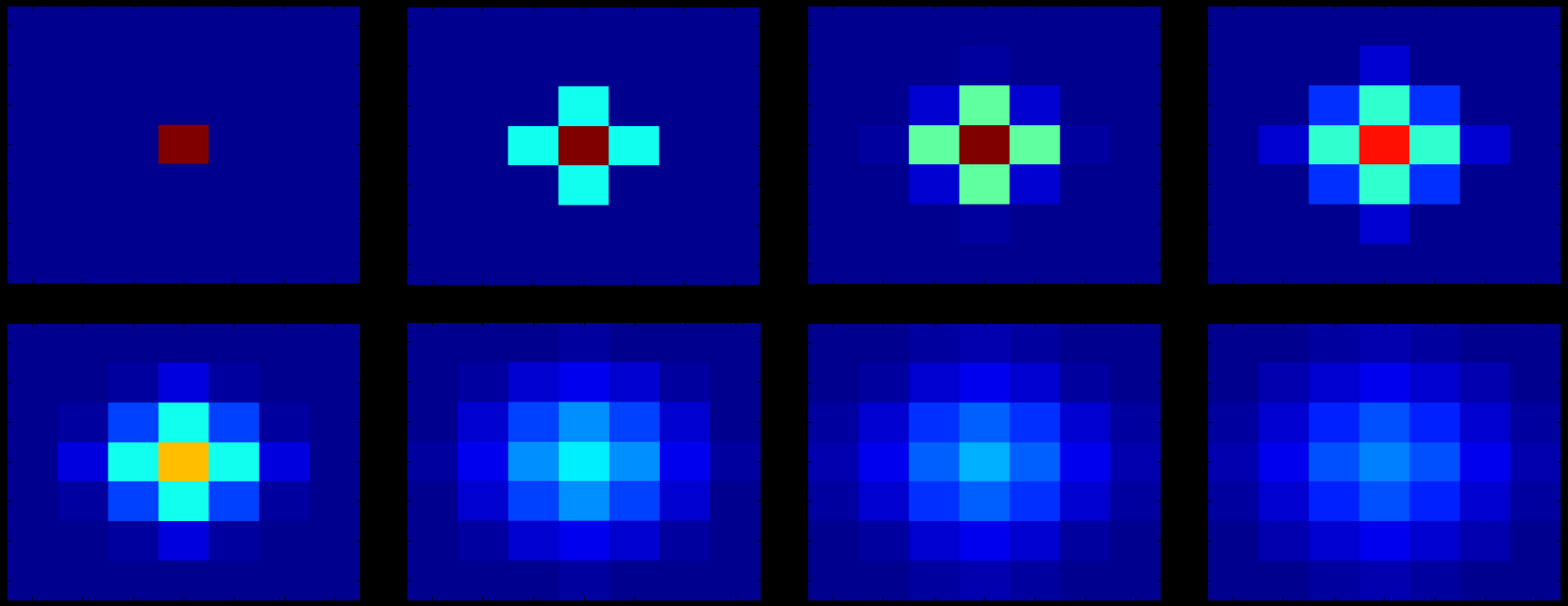
Diffusionsgleichung (1)

- Gleichmässige Verteilung/Ausbreitung im Raum
- Harte Kanten verhindern



Diffusionsgleichung (2)

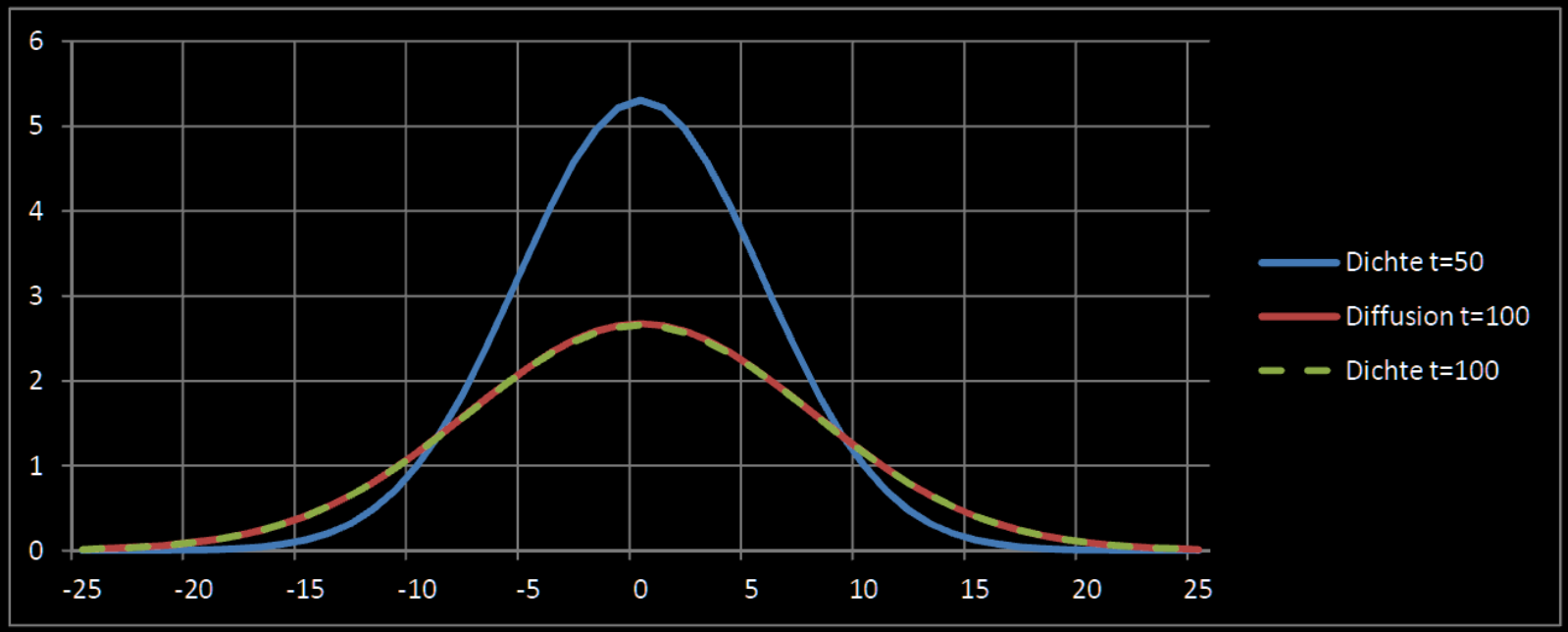
Resultat



Diffusionsgleichung (3)

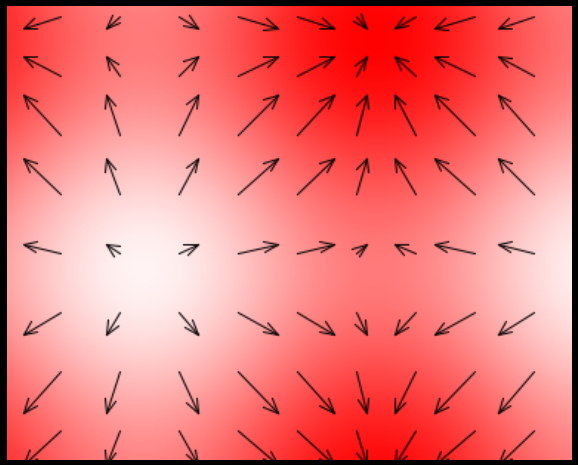
Verifikation: Dichtefunktion

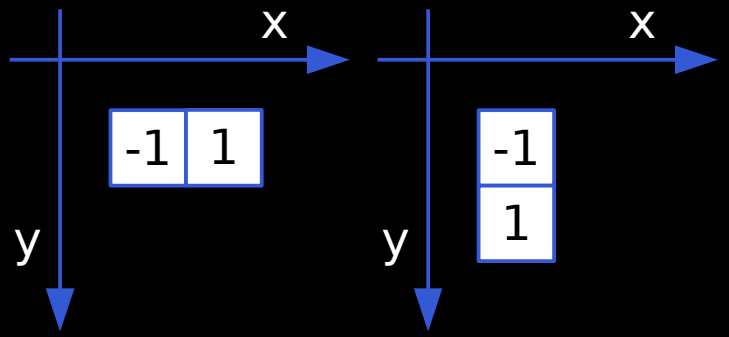
$$G(\vec{r},t)=\frac{1}{4\pi D t}\cdot e^{-\frac{x^2+y^2}{4Dt}}$$



Gradienten-Verfahren

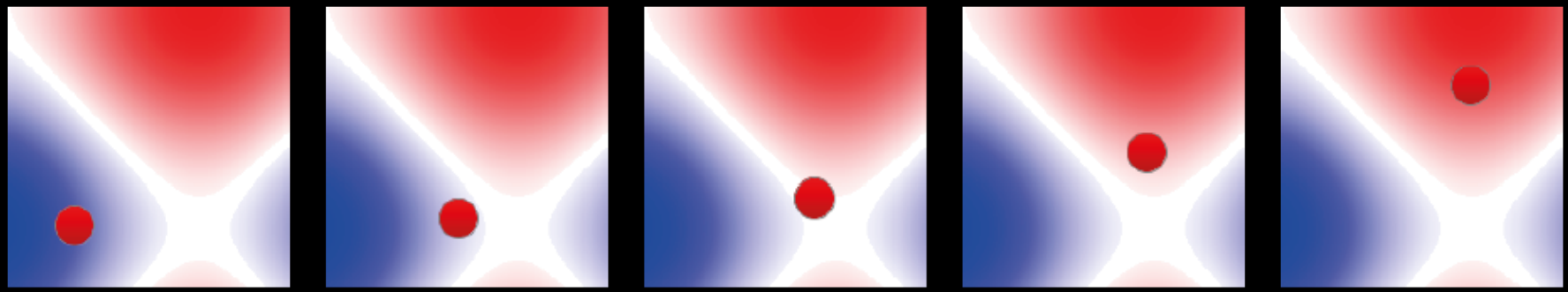
- Verfahren des steilsten Abstiegs
- Wandert schrittweise zum lokalen Minimum/ Maximum





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 Buildystem
- 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?