

Modellierung und Simulation in der Biochemie

Ursula Kummer

Vorlesung 3

Simulation biochemischer Netzwerke

Übersicht

- Kurze Wdh.
- Simulation allgemein
- Numerische Integration, Programme + Beispiel

Kurze Wdh.

- Oft Modellierung auf Reaktionsebene
- Näherungen durch Michaelis-Menten falls möglich
- Systeme werden beschrieben, indem alle Terme, die direkt zur Produktion oder zum Verbrauch einer Variablen (z.B. Metabolit) führen, aufsummiert werden

Simulation allgemein

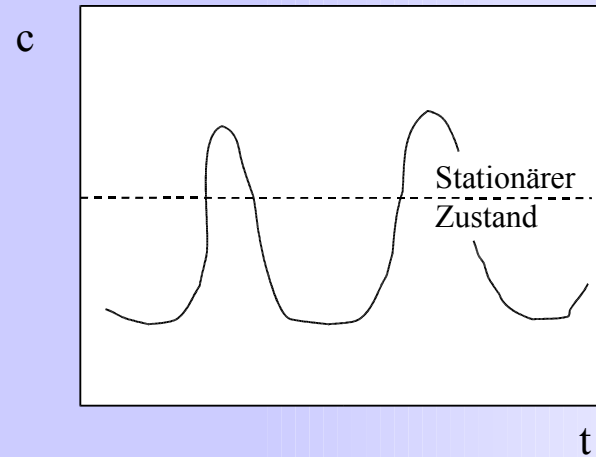
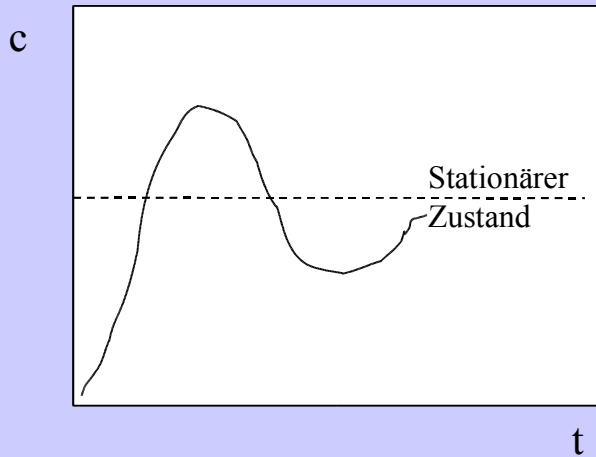
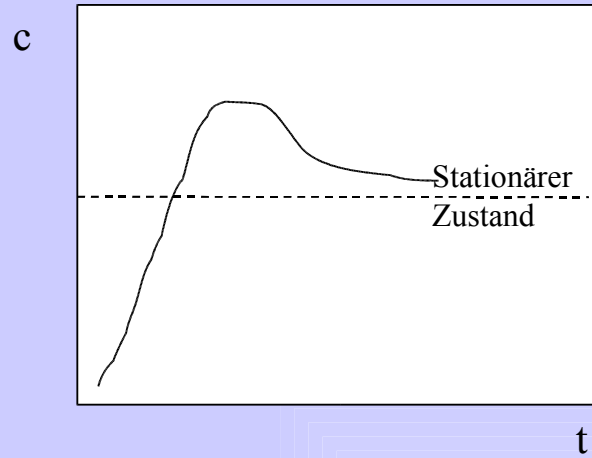
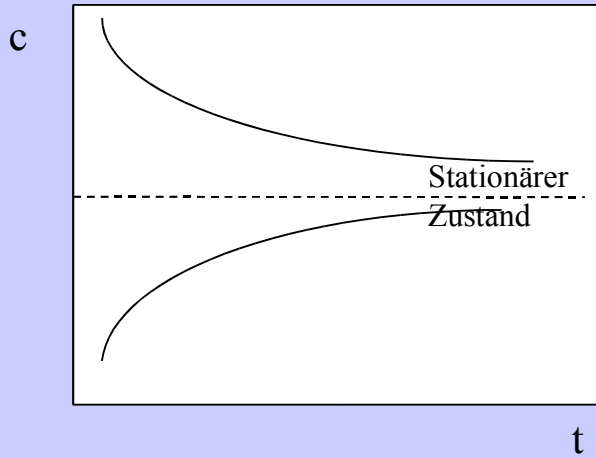
- Berechnen des Zeitverlaufs der Systemgleichungen
- Systemgleichungen der Form:

$$A' = -k_1 * R_1 + k_2 * R_2 \text{ etc.}$$

(wobei R1,R2 für Reaktionsterme steht)

- Da diese fast immer nichtlinear sind, muss man zum Lösen meist numerische Verfahren einsetzen

Konzentrations-Zeit-Diagramme in offenen Systemen



Analytische Lösung

- Nur bei linearen Systemen und wenigen einfachen nichtlinearen

Beispiel:

$$dA/dt = -k*A$$

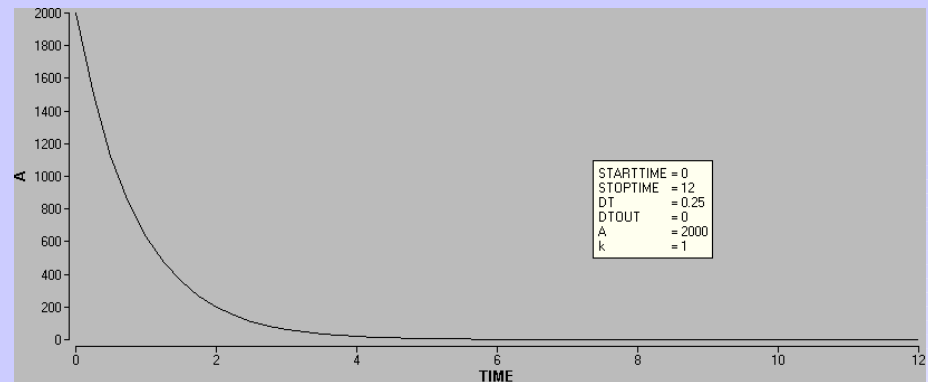
Variablentrennung:

$$\rightarrow \int_{A0}^A \frac{dA}{A} = \int_0^t -k*dt$$

Integration:

$$\rightarrow \ln \frac{A}{A0} = -k*t$$

$$A(t) = A0 * e^{-k*t}$$



Problem:

$$dx/dt = f(x)$$

und $x = (x_1, \dots, x_n)$

Euler-Verfahren:

Lösungsansatz:

Diskretisierung der Zeit in Zeitintervalle Δt .

Anfangswertprobleme:

$$x(t_0) = x_0$$

gesucht $x(t_0 + \Delta t) = ???$

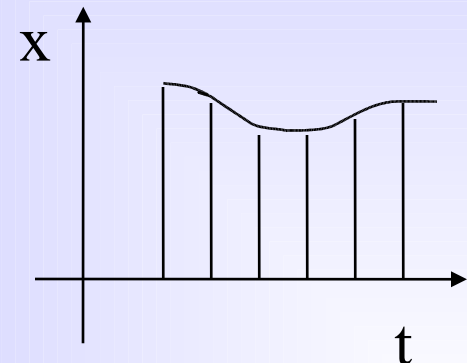
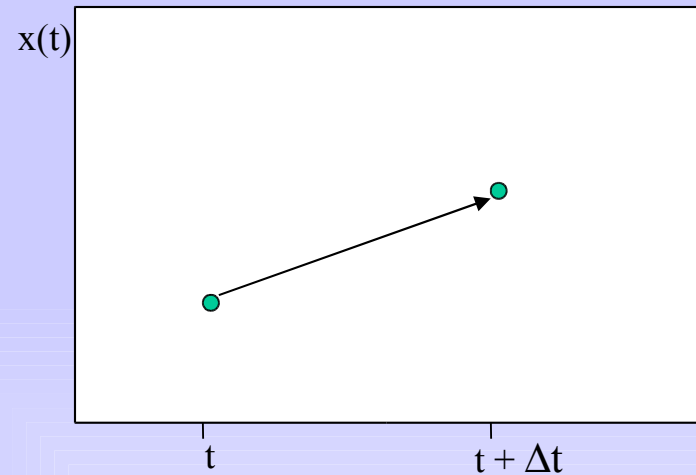
Taylorreihe:

$$x(t_0 + \Delta t) = x(t_0) + (dx/dt)_{t_0} * \Delta t + 1/2(d^2x/dt^2)_{t_0} * \Delta t^2 + \dots$$

für kleine Δt : $x(t_0 + \Delta t) = x(t_0) + (dx/dt)_{t_0} * \Delta t$

$$x(t_0 + \Delta t) = x(t_0) + f(x) * \Delta t$$

Der Fehler ist proportional (d^2x/dt^2)



Runge-Kutta-Verfahren:

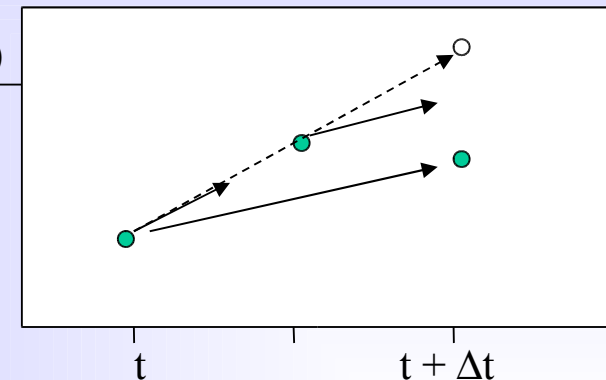
Idee:

Durch Einführung von Zwischenschritten kann man die Genauigkeit erhöhen.

Rechenvorschrift:

t	x	$k_i = \Delta t * f(x_i, t_i)$
t_0	x_0	$k_1 = \Delta t * f(x_0, t_0)$
$t_0 + 1/2 * \Delta t$	$x_0 + 1/2 k_1$	$k_2 = \Delta t * f(x_0 + 1/2 k_1, t_0 + 1/2 * \Delta t)$
$t_0 + 1/2 * \Delta t$	$x_0 + 1/2 k_2$	$k_3 = \Delta t * f(x_0 + 1/2 k_2, t_0 + 1/2 * \Delta t)$
$t_0 + \Delta t$	$x_0 + k_3$	$k_4 = \Delta t * f(x_0 + k_3, t_0 + \Delta t)$
$t = t_0 + \Delta t$	$x = x_0 + k$	$k = 1/6(k_1 + 2k_2 + 2k_3 + k_4)$

Verfahren 4.Ordnung, d.h. Fehler ist proportional d^5x/dt^5



Andere Verfahren

- Euler und Runge-Kutta nicht geeignet für steife ODEs
- Entweder fortentwickelte Runge-Kutta-Verfahren mit variablem Zeitschritt und z.B. zusätzlicher “Midpoint”-Methodik (Beispiel: Rosenbrock)
- Predictor-Corrector-Methoden:
“Kontrollieren” die Integration, indem sie rückwärts differenzieren.
(Beispiele: Gear, Adams, LSODE (Hindmarsh et al.), LSODA)

Details s. Numerical Recipes

Wie entscheidet man sich für welche Numerik?

- Biochemische Systeme sind fast immer steif -> auf keinen Fall Euler oder Runge-Kutta!
- Rosenbrock, Gear oder LSODE/LSODA sind gute Solver
- Man kann noch immer numerische Artefakte haben, daher Vorsicht!
- Am besten man vergleicht die Ergebnisse zweier verschieden ausgelegter Solver
- Leider bieten die meisten Programme dies nicht an

Programme - Madonna 1

Netscape: Berkeley Madonna - Modeling and Analysis of Dynamic Systems

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Location: <http://www.berkeleymadonna.com/> What's Related

Members WebMail Connections BizJournal SmartUpdate Mktplace

BERKELEY MADONNA

Modeling and Analysis of Dynamic Systems

QUICK TOUR

- Features
- Examples
- Download
- Pricing
- Purchase

$$\frac{dx_i}{dt} = f_i(x_1, \dots, x_n, t), \quad i = 1, \dots, n$$

Solve Differential Equations in Milliseconds!

Berkeley Madonna is arguably the fastest, most convenient, general purpose differential equation solver available today. It is relatively inexpensive and runs on both Windows and Mac OS. Developed on the Berkeley campus under the sponsorship of NSF, it is currently used by academic and commercial institutions for constructing mathematical models for research and teaching.

Latest Version: 8.0.1

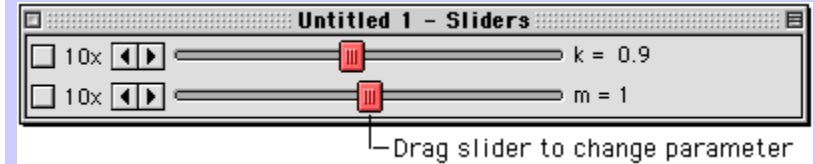
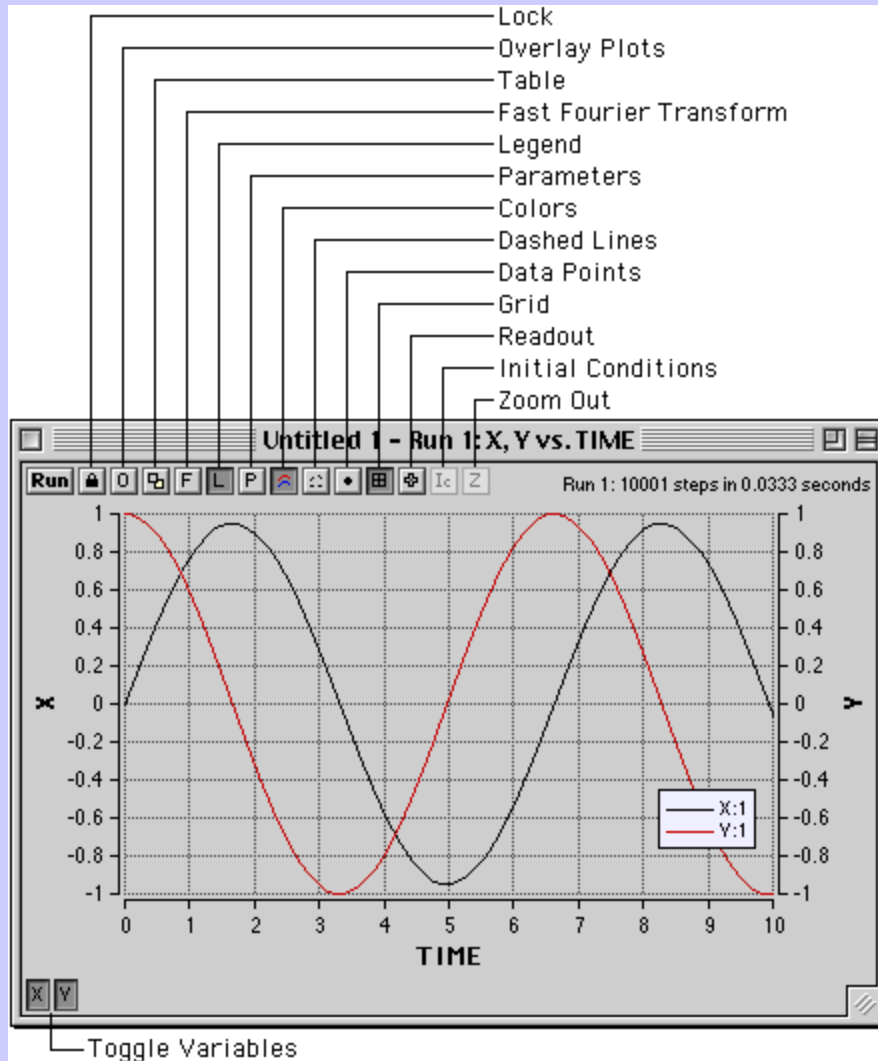
Berkeley Madonna? was developed by Robert Macey and [George Oster](#) of the [University of California at Berkeley](#).
Programmed by Tim Zahnley. Email: madonna@kaci.com

Copyright ©1993-2001 Robert I. Macey & George F. Oster
Site Design: [Hinshaw Design Group](#)

Konsole - ... VMware W... Netscape: ...
Terminal - ... Netscape ...

17:51
2001-09-21

Programme - Madonna 2



Programme - Copasi 1


COPASI : HomePage - Mozilla

File Edit View Go Bookmarks Tools Window Help

http://www.copasi.org/tiki-index.php

Home Bookmarks The Mozilla Or... Latest Builds

backlinks...



Complex Pathway Simulator

[Copasi FAQ](#) | [Screenshots](#) | [Download](#) | [People](#)
Latest version: **Test Build 10** released Nov 4, 2004

What's new in Build 10

Copasi is a software application for simulation and analysis of biochemical networks. The test version of Copasi is free for **Download**, full releases will continue to be free for academic users. Commercial use of Copasi only possible through licensing agreements. **The current version of Copasi is still a test version; new releases appear frequently, come back to www.copasi.org!**



Current Features:

- stochastic and deterministic time course simulation
- steady state analysis (including stability)
- elementary mode analysis
- mass conservation analysis
- imports and exports **SBML** level 2
- loads **Gepasi** files
- versions for Windows, Linux, and OS X
- visit this page often, new releases will contain many more features!

Currently known problems:

- importing some SBML files crashes copasi.

By the **Mendes group** at VBI and **Kummer group** at EML Research.



Created by: system last modification: Sunday 07 of November, 2004 [21:03:40 UTC] by mendes

[history](#) [similar](#) [export](#)

Login

user:

pass:

Remember me

[I forgot my pass](#)

Programme - Copasi 2

COPASI (4.0 Build 10 test version)

File Tools Help

Copasi

- Model
 - Biochemical
 - Compartments
 - Metabolites
 - Moiety
 - Reactions
 - ADH
 - AK
 - ALD
 - ATPase
 - ENO
 - G3PDH
 - GAPDH
 - Glycogen Branch
 - HK
 - HXT
 - PDC
 - PFK
 - PGI
 - PGK
 - PGM
 - PYK
 - Succinate Branch
 - TPI
 - Trehalose Branch
 - Mathematical
- Tasks
 - Steady-State
 - Stoichiometry
 - Time Course
 - Result
- Output
 - Plots
 - ConcentrationPlot
 - Reports
- Functions

Task Name: Trajectory Task Task Executable

Start Time: 0 End Time: 2

Interval size: 0.02 Intervals: 100

store time series in memory

Method: Deterministic (LSODA)

Parameter value	Value
LSODA.RelativeTolerance	1e-12
LSODA.AbsoluteTolerance	1e-06
LSODA.AdamsMaxOrder	12
LSODA.BDFMaxOrder	5
LSODA.MaxStepsInternal	10000

Copasi Plot: ConcentrationPlot <2>

Zoom Print

ConcentrationPlot

Legend:

- GLC1
- ATP
- G6P
- ADP
- F6P
- F16bP
- AMP
- DHAP
- GAP
- NAD
- BPG
- NADH
- P3G
- P2G
- PEP
- PYR
- AcAld
- Succinate
- Trehalose
- Glycogen
- Glycerol
- EtOH
- CO2
- F26bP
- GLCo

Run Revert ReportDefinition

Weitere Programme

- Numerische Integration kann natürlich von praktisch allen Mathematik-Programmen durchgeführt werden, also MAPLE, Mathematica, Octave etc.
- Unter Linux: gnuode
- Mit Sicht auf Chemie oder Biochemie:
Copasi, Jarnac, E-cell