

E2 - Proteine



7. Tag: Simulation - Methoden und Werkzeuge

Ursula Kummer, Sven Sahle

Femke Mensonides, Irina Surovtsova, Jürgen Zobeley

Simulation allgemein



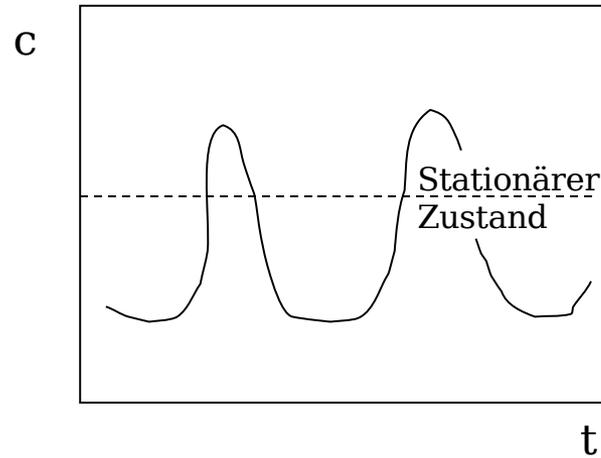
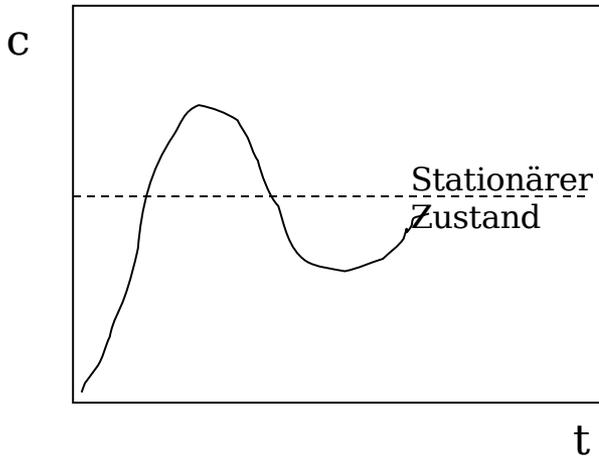
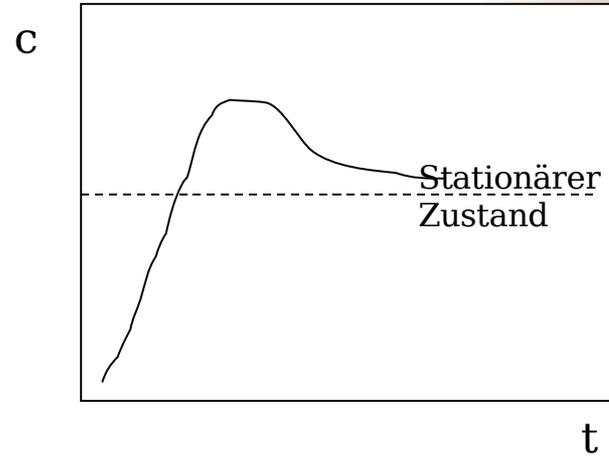
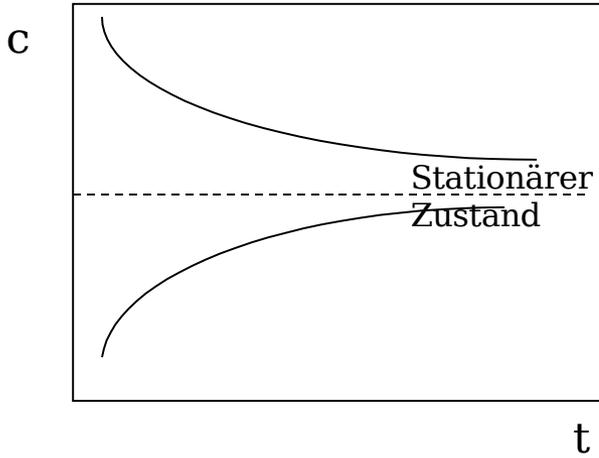
Berechnen des Zeitverlaufs der Systemgleichungen
Systemgleichungen der Form:

$$R' = -k_1 \cdot R_1 + k_2 \cdot R_2 \text{ etc.}$$

(wobei R1, R2 für Reaktionsterme steht)

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

offenen Systemen



Analytische Lösung

ur bei linearen Systemen und wenigen einfachen nichtlinea

Beispiel:

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

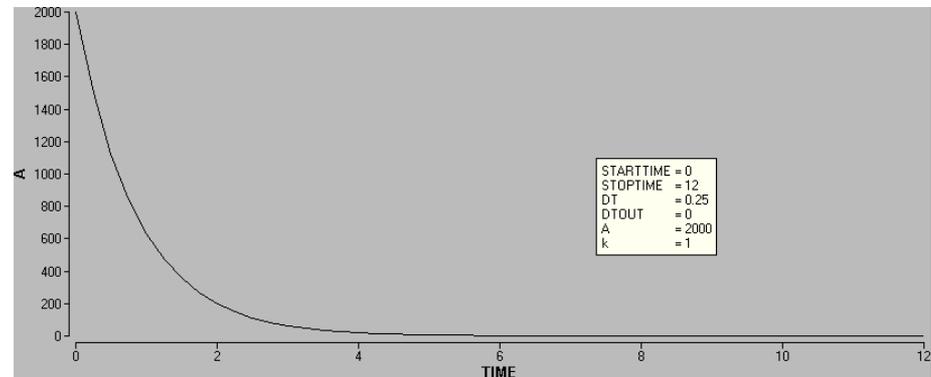
Variablentrennung:

$$\rightarrow \int_{A_0}^A \frac{dA}{A} = \int_0^t -k * dt$$

Integration:

$$\rightarrow \ln \frac{A}{A_0} = -k * t$$

$$A(t) = A_0 * e^{-k * t}$$



Euler-Verfahren:

Problem:

$$\dot{x} = f(x)$$

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

Näherungsansatz:

Diskretisierung der Zeit in Zeitintervalle Δt .

Randwertprobleme:

$$x = x_0$$

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

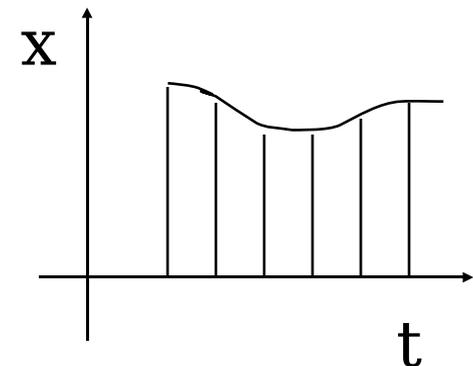
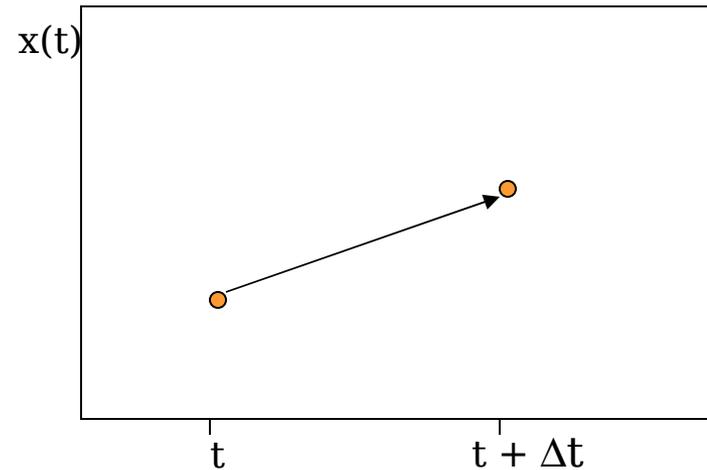
Reihe:

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

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

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

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



Runge-Kutta-Verfahren:

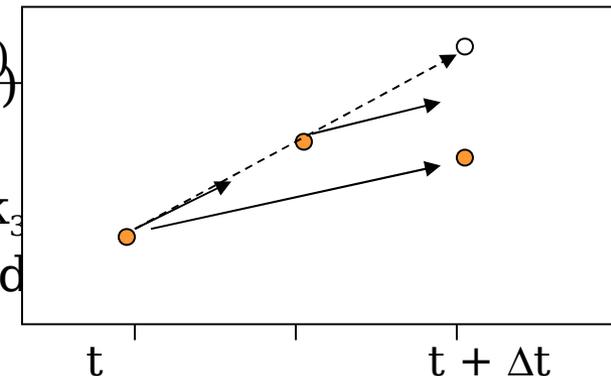
Idee:

Durch Einführung von Zwischenschritten kann man die Genauigkeit erhöhen.

Berechnungsvorschrift:

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

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



Andere Verfahren



er 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)

Detailliertes s. Numerical Recipes

LSODE/LSODA



- LSODE (Livermore Solver for Ordinary Differential Equations)
- Adams methods (predictor-corrector) für den weniger steifen Fall
- Backward Differentiation Formula (BDF - Gear) für den steifen Fall
- Sehr robust und zuverlässig

- Adams/Bashfort - Adams/Moulton
- Predictor-corrector-Methode
- Gehen von einigen Lösungspunkten aus und berechnen dann den besten Fit für ein Polynom n-ter Ordnung
- Keine Fehlerpropagation
- Predict (aufgrund des Polynoms) -> Differentiation
-> korrigierter Punkt
- Limitierung: Man muss die ersten Punkte anders gewinnen ->
man braucht „gute“ Differenzierung

- Backwards Differentiation Formula
- Interpolation von existierenden n Lösungspunkten mit dem unbekanntem Lösungspunkt $n+1$ (Anlegen der Tangente)
- Ableitung muß wieder mit der ODE verglichen werden
- Sehr gut bei steifen Systemen
- Limitierung ähnlich wie bei Adams, aber noch besser für steife Systeme

Wie entscheidet man sich für welche Numerik?



Biochemische Systeme sind fast immer steif -> auf keinen 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 verschiedener 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

Bookmarks 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

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

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 ...

2001-09-21 17:51

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

Monte Carlo Simulationen



stochastischer Ansatz

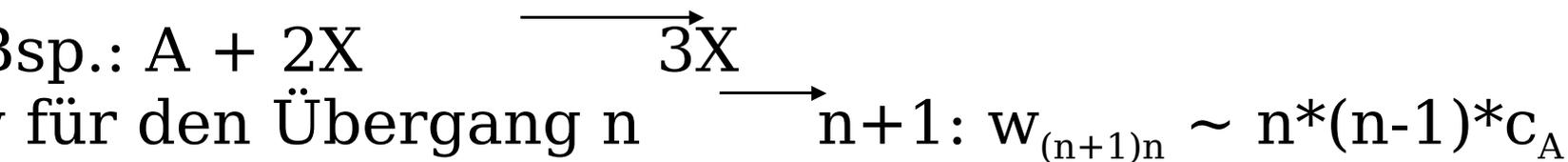
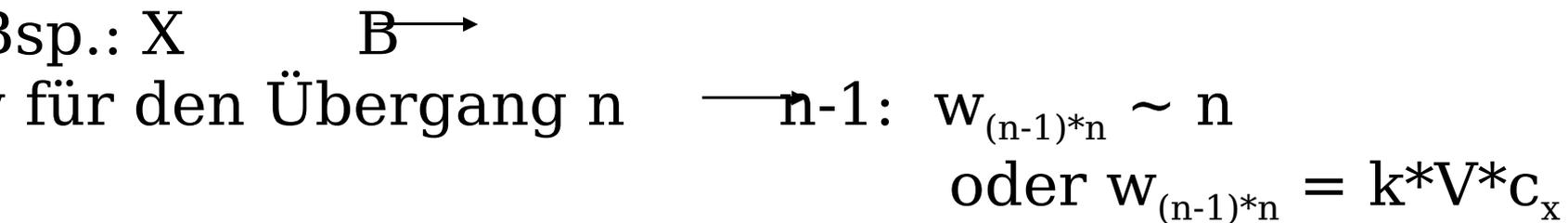
Es werden nicht die Konzentrationsänderungen eines Stoffes
sondern die Änderung der Teilchenzahl berechnet

Wann auf Teilchenebene in (bio)chemischen Systemen der stochastische
Einfluß groß ist (Brownsche Bewegungen etc.) wird bei
geringer Teilchenzahl der stochastische Ansatz gewählt

Darüber hinaus besteht die Gefahr von halben Teilchen etc.!

Die stochastische Reaktionswahrscheinlichkeit

$v_j \cdot dt$ beschreibt die durchschnittliche Wahrscheinlichkeit, dass eine einzelne Reaktion vom Typ j im nächsten Zeitintervall dt stattfindet



Die Mastergleichung



Die "grand probability function" $P(X_1, X_2, \dots, X_N; t)$ beschreibt die Wahrscheinlichkeit, daß in einem abgeschlossenen Bereich zu einem Zeitpunkt t X_1 Moleküle einer Spezies S_1 , X_2 Moleküle einer Spezies S_2 X_N Moleküle der Spezies S_N vorhanden sind

Die Mastergleichung beschreibt die zeitliche Entwicklung der Wahrscheinlichkeitsfunktion:

$$dP_n/dt = \sum [w_{nn'} * P_{n'}(t) - w_{n'n} * P_n(t)]$$

Simulation der Trajektorie mit festem Zeitschritt

gehend von einem Zustand n , berechnet man die Wahrscheinlichkeit, daß nach einem bestimmten Zeitintervall dt ein Sprung in einen anderen Zustand n' erfolgt.

... dt wird nachgeschaut, ob die Wahrscheinlichkeit ergibt, daß eine Sprung stattgefunden hat

... wenn ein Sprung statt, wird ermittelt, in welchen Zustand n' er erfolgt ist

WICHTIG!: Jede Simulation führt hier zu unterschiedlichen Trajektorien. Daraus folgt, daß man u.U.viele Durchläufe machen

mit stochastischem Zeitschritt

... nach jedem Zeitschritt abzufragen, ob ein Sprung stattfindet.
... eine Verweilzeitverteilung für den Zustand n bestimmt.

... dann wird eine Zufallszeit gemäß dieser Verteilung als Zeitschritt
... bestimmt.

... es hat insbesondere den Vorteil, daß man nicht ständig
... überprüfen muß, ob $u_0(n)dt < 1$ ist.

... nach hier gilt natürlich, daß jede Trajektorie anders aussieht

Gillespie-Algorithmus

(1) Zunächst wird die totale Reaktionswahrscheinlichkeit $u_0(n)$ berechnet. Diese ist die Summe aller Wahrscheinlichkeiten für die j möglichen, individuellen Reaktionen:

$$u_0(n) = \sum_j w_j$$

(2) Der stochastische Zeitschritt wird nach

$$\Delta t = -\frac{\ln(\xi_1)}{u_0(n)}$$

berechnet, wobei ξ_1 eine über $[0,1]$ gleichverteilte Zufallszahl ist.

(3) Nun muß berechnet werden, welche Reaktion α stattfindet. Dazu wird eine zweite über $[0,1]$ gleichverteilte Zufallszahl ξ_2 generiert und die Reaktion α nach folgendem Kriterium gewählt.

$$\sum_{j=1}^{\alpha-1} \frac{w_j}{u_0(n)} \leq \xi_2 \leq \sum_{j=1}^{\alpha} \frac{w_j}{u_0(n)}$$

Die entsprechende Reaktion wird dann auf Teilchenebene “durchgeführt”, d.h. die Anzahl der beteiligten Molekülspezies erniedrigt oder erhöht. Der gesamte Vorgang (1-3) wird dann so oft wiederholt, bis man die gewünschte Zeitspanne dargestellt hat.

Zufallszahlen



Generierung von Zufallszahlen:

- Achtung: Diese liegen nicht dicht, sondern nur gleich verteilt in Intervallen
- Sind vom Typ „real“
- Je nach Computer und Programmiersprache gibt es immer eine endliche Zahl von Nachkommastellen
- -> $\ln(x)$ daher nicht beliebig groß

Zeitreihen

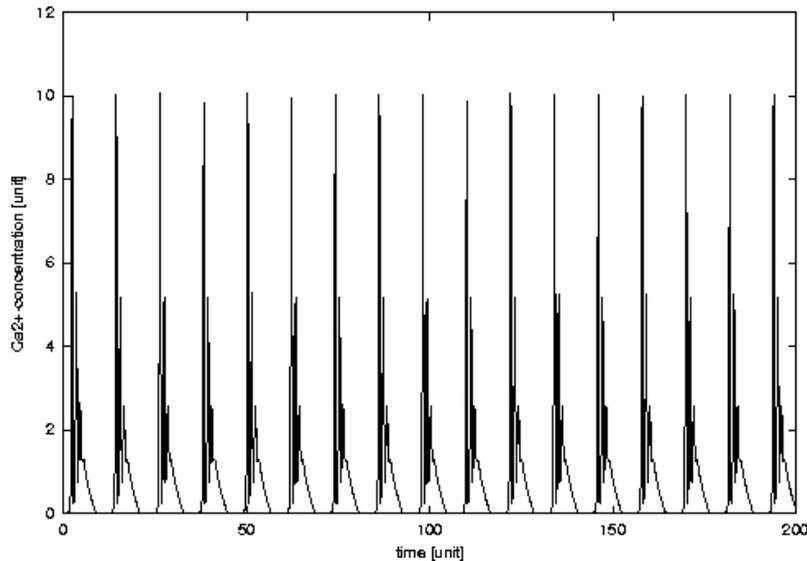


ft ist ja gewünscht, eben nicht den im Mittel gerechneten, deterministischen Verlauf zu ermitteln, sondern das Verhalten einzelner, stochastischer Trajektorien zu ermitteln, wie es die Natur auch macht

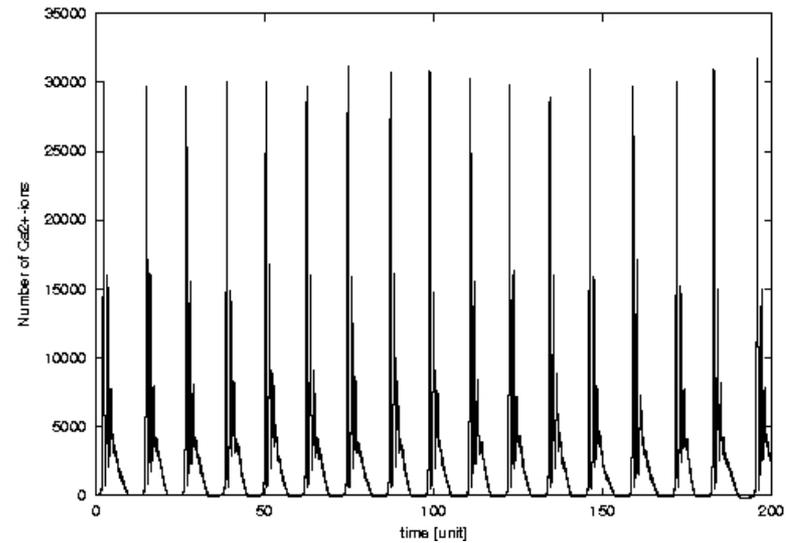
ehler mitteln sich natürlich auch nicht nur über eine Menge Zeitreihen, sondern auch mit der grossen Zahl von Molekülen und/oder grosser Zahl von Zeitschritten heraus

Calciumsignalen

deterministisch



stochastisch



gibt bisher keine festen Regeln, um zu bestimmen, wann der Übergang von stochastischem zu deterministischem Verhalten stattfindet!

Kann man auch komplexe Kinetik umsetzen?

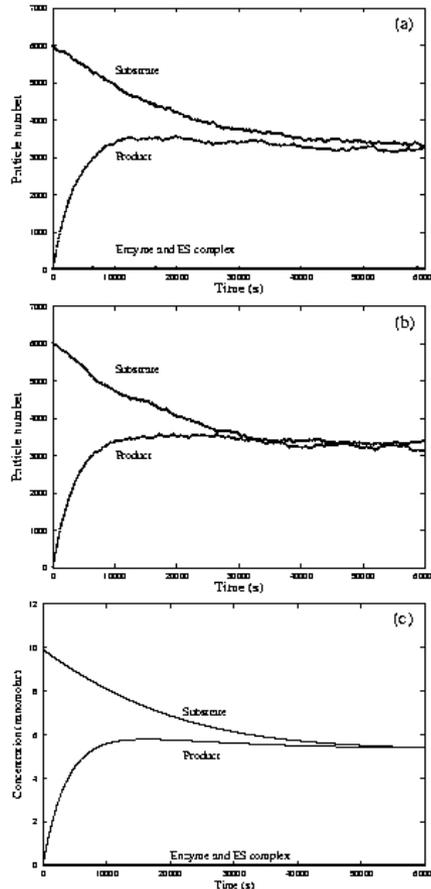


Figure 3: Simulations of a simple enzymatic reaction with high particle numbers. Parameters: $k_f=1$, $k_b=2$, $k=0.02$, $k_c=0.00145$, $k_d=0.00027$, $[S]_0=10$ nM, $[E]_0=0.1$ nM. a) stochastic simulation of the full open system corresponding to eqn. 10. b) stochastic simulation of the same system using the Michaelis-Menten approximation c) deterministic simulation of the same system.

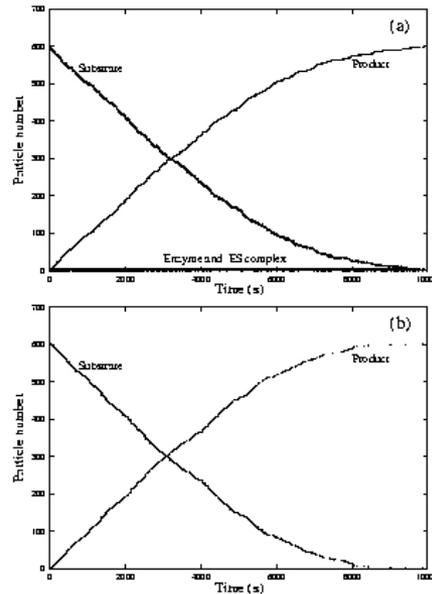


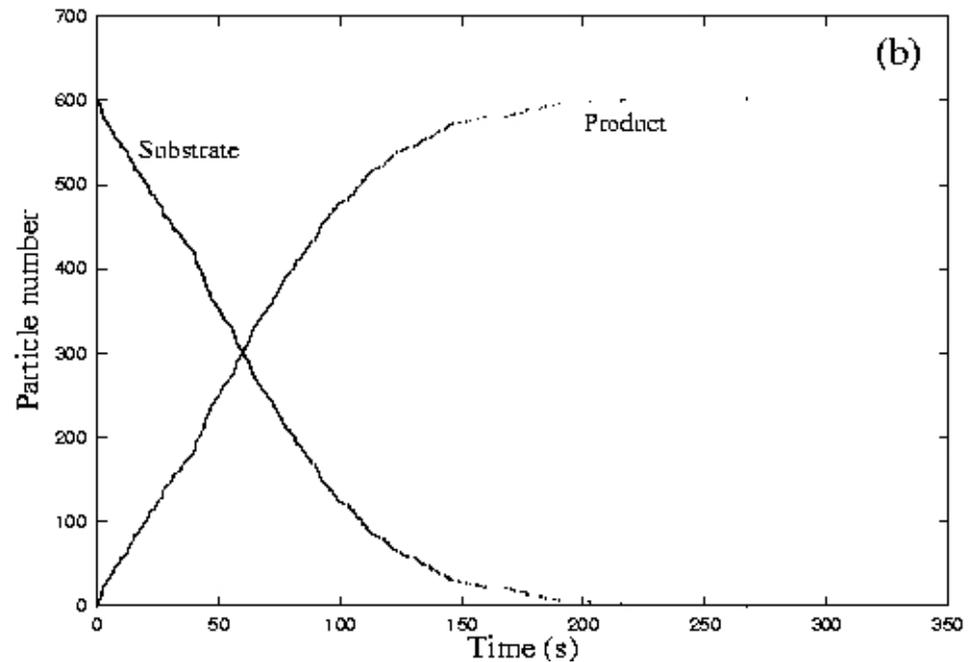
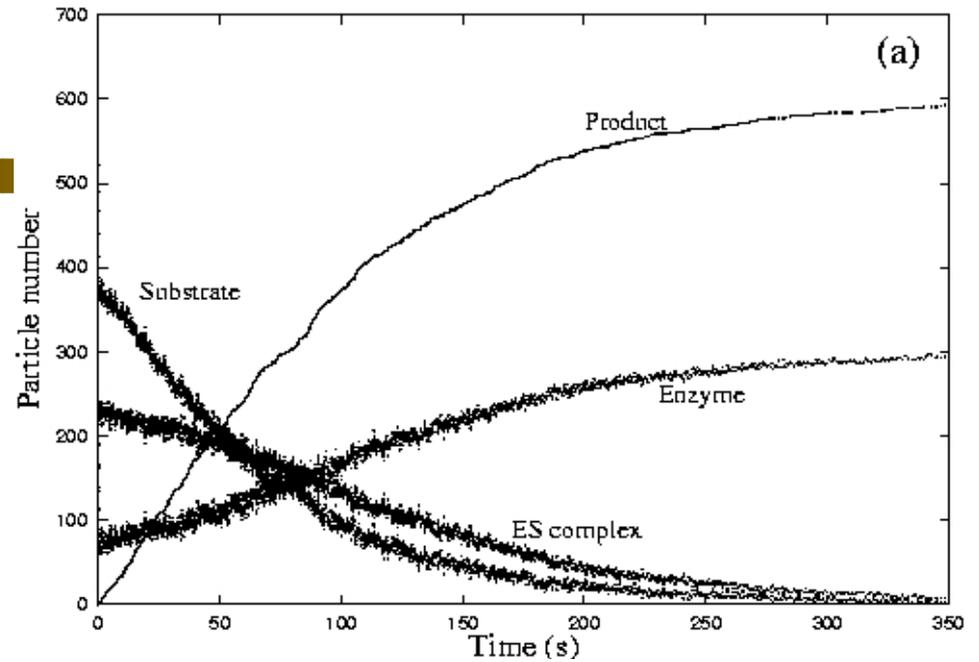
Figure 4: Simulations of a simple enzymatic reaction with low particle numbers. Parameters: $k_f=10$, $k_b=2$, $k=0.02$, $[S]_0=1$ nM, $[E]_0=0.01$ nM. a) stochastic simulation of the full system corresponding to eqn. 1. b) stochastic simulation of the same system using the Michaelis-Menten approximation.

- Im Prinzip ist der Formalismus NUR für Elementarreaktionen gültig!
- Man kann aber testen, ob eine komplexe Kinetik auch funktioniert:



Cont.

Wenn die Voraussetzungen für Michaelis-Menten nicht mehr gegeben sind, dann kann man natürlich diese auch in diesem Fall nicht einsetzen!



Verfahren



ibson und Bruck:

Prinzip wie Gillespie, aber es wird Zeit gespart, indem die Aktionen nach ihrem errechneten "tau" in eine priority-queue geschoben werden und dann abgearbeitet werden

tau-leap-Methode von Gillespie:

u. Versucht mehrere schnelle Schritte auf einmal auszuführen

Hybrid-Methode