

# Internet Protokolle II

---

*Messungen & Simulationen*


*Thomas Fuhrmann*



Network Architectures  
Computer Science Department  
Technical University Munich

---

# *Messungen & Messfehler*

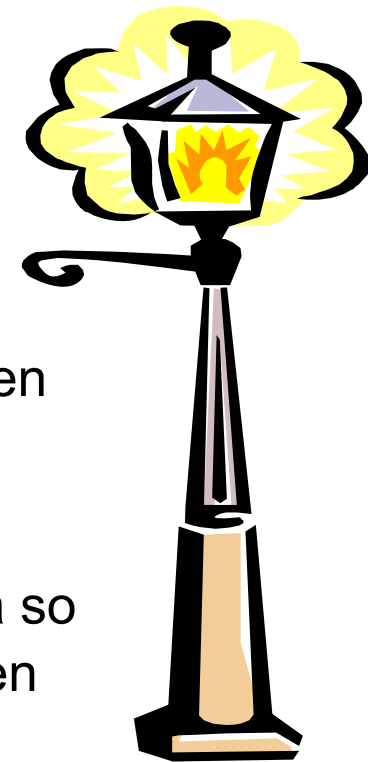


# Modelle und Modellbildung

- Oft kann man ein Protokoll nicht vollständig oder nicht in einem realistischen Anwendungsumfeld testen.
- Dann hilft ein Modell des Protokolls bzw. seines Anwendungsumfelds.
- Manchmal kann die Implementierung des Protokolls selbst als Modell dienen (→ Simulation).
- Meist abstrahiert das Modell aber von der Wirklichkeit, d.h. erfasst nicht alle Details (Kostensparnis, Erkenntnisgewinn).
- Oft wird diese Abstraktion mittels Mathematik formuliert (Funktion, Differentialgleichung). Sie kann aber auch algorithmisch bzw. durch ein vereinfachtes Protokoll simuliert werden.
- Gute Modelle vereinfachen soweit wie möglich (Ockham), aber eben nicht weiter (Einstein). Menschen neigen dazu, komplizierten Modellen mehr zu glauben (Watzlawick).
- Messungen helfen ein Modell mit der Wirklichkeit abzugleichen.

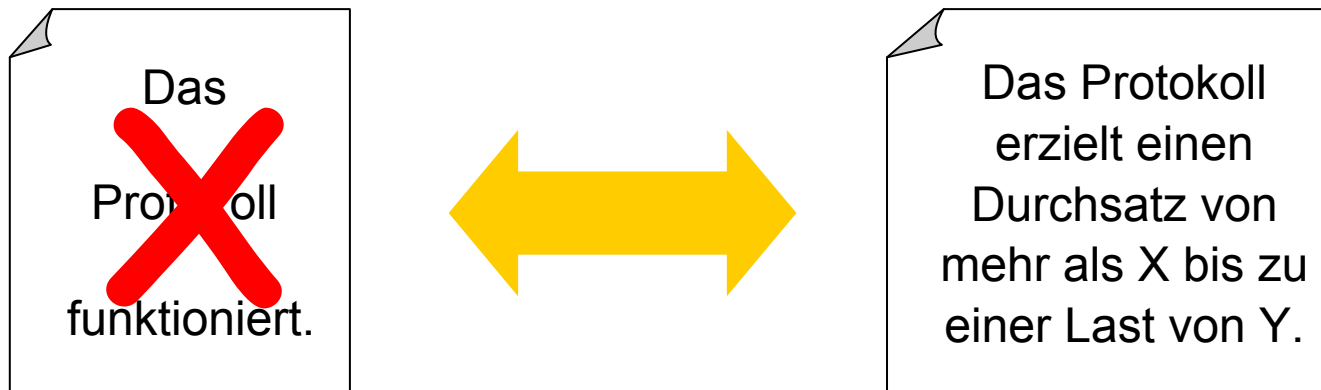
# Modellbildung und Messung

- Der wiederholte Durchlauf des Zyklus ...
  - Formulierung eines Modells bzw. Protokollentwurf
  - Analyse des Modells und Ableiten von Vorhersagen
  - Überprüfung der Vorhersagen durch Messungen... erfordert Messungen sowohl in der Realität als auch (z.B. mittels Simulation) am Modell selbst.
- Ziel der Messungen ist es, Abweichungen zwischen antizipiertem Verhalten und der Wirklichkeit aufzudecken bzw. die Grenzen des Protokolls bzw. Modells auszuloten.
- Achtung: Nicht unter der Laterne suchen nur weil es da so schön hell ist, sondern da suchen, wo man vermutet den verlorenen Schlüssel wieder zu finden.



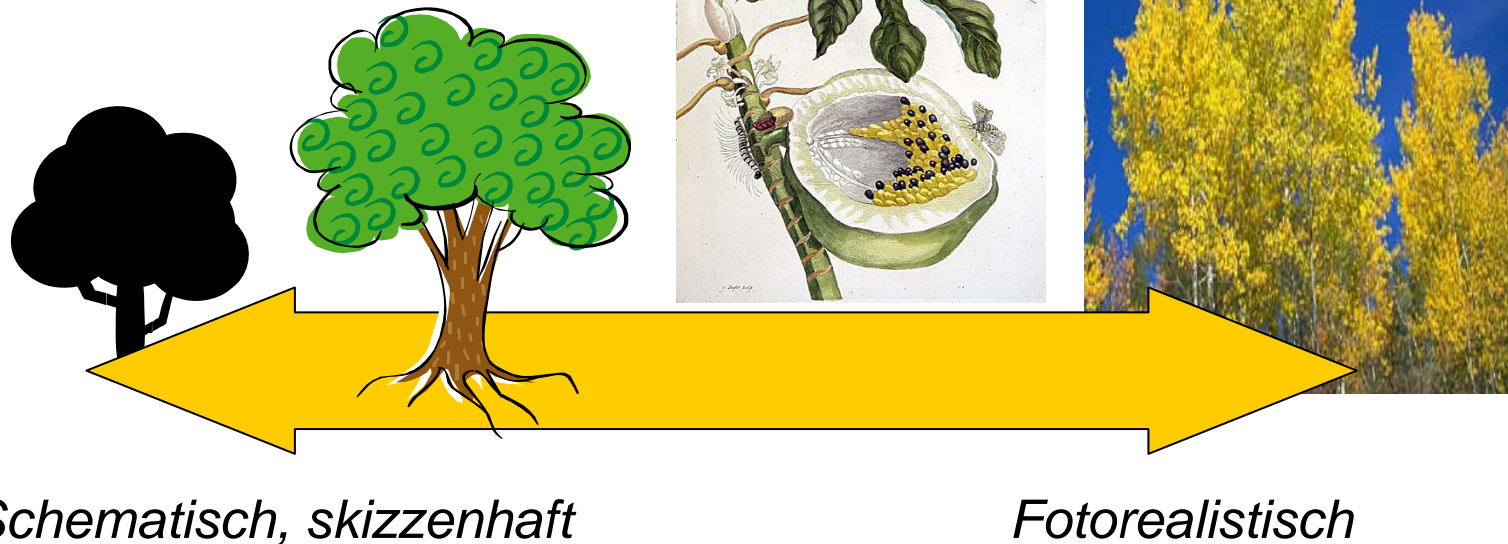
# Messung ist nicht Funktionstest

- Grenzen ausloten bedeutet:
  - Im Maschinenbau: Messe bis zum Zerbrechen der Probe!
  - Hier: Messe unter immer größeren Anforderungen, solange bis das Protokoll seinen Zweck nicht mehr erfüllt.
- Untersuche zunächst die typischen Anwendungsfälle (um rasch Abweichungen zur Erwartung aufzudecken).
- Untersuche dann auch extreme Fälle (um die Grenzen auszuloten):
  - Extreme Anforderungen (viele Anfragen, große Datenmengen, etc.)
  - Extreme Randbedingungen (viel Querverkehr, Node Churn, etc.)



# Feinheit der Messung

- Messungen und Simulation ähneln sich beim Protokollentwurf.
- Entscheidend ist in beiden Fällen aber der Grad des Realitätsbezugs:
  - Wie viele Details erfasse ich in der Simulation?
  - Wie realistisch ist das Umfeld der Messung?



# Messung und Simulation

	<i>Messung</i>	<i>Simulation</i>
<i>Skizzenhaft</i>	<p><u>Beispiel:</u> Protokoll wird implementiert und auf fünf Testgeräten vermessen.</p> <p><u>Problem:</u> Unrealistisches Messumfeld; Ergebnis ggf. nicht auf spätere Anwendung übertragbar.</p>	<p><u>Beispiel:</u> Entwurfsprinzip wird in Form eines einfachen Simulationsmodells untersucht.</p> <p><u>Problem:</u> Reine Modellaussage, Relevanz bleibt unklar.</p>
<i>Detailgenau</i>	<p><u>Beispiel:</u> Weit verbreitetes Protokoll wird im Internet vermessen.</p> <p><u>Problem:</u> Hoher Aufwand, oft zu spät für Protokollentwicklung</p>	<p><u>Beispiel:</u> Protokoll wird implementiert und im Simulator getestet.</p> <p><u>Problem:</u> Simulation benötigt immer noch ein Modell, z.B. für das Nutzerverhalten.</p>



= Aufwand nimmt von rechts oben nach links unten zu, Modellanteil nimmt ab.

# Bemerkungen zur Nützlichkeit

---

- Eine skizzenhafte Simulation wird oft abgelehnt weil die Aussagekraft der Ergebnisse von der Qualität der Skizze (=Modell) abhängt.
- Ihr Hauptvorteil ist, schnell Ergebnisse zu liefern, die direkt zur Verbesserung des Protokollentwurfs beitragen können.
- Skizzenhafte Messungen und detailgetreue Simulationen haben ein höheres Ansehen.
- Oft werden die Gefahren eines unrealistischen Messumfelds bzw. falscher Modellannahmen übersehen.
- Das Vermessen von realen Systemen (z.B. Internet) liefert realistische Ergebnisse. Zum Protokollentwurf tragen sie aber nur indirekt bei, z.B. durch Formulierung von neuen Modellen.



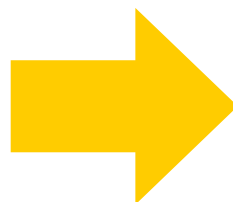
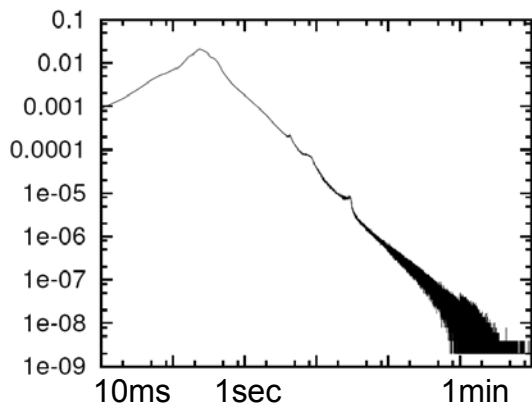
## Beispiel: Skizzenhafte Simulation

- Aufgabe: Alle Rechner eines verteilten Systems speichern Musikdateien. Die Dateien sind unterschiedlich groß und werden unterschiedlich oft nachgefragt. Jede Datei kann auf mehreren Rechnern vorhanden sein.
- Protokollidee: Ein zentraler Server speichert eine Liste aller Dateien zusammen mit Verweisen auf die Rechner die diese Dateien besitzen. Benötigt ein Rechner eine Datei wählt er einen Rechner aus der Liste und lädt die Datei. Danach trägt er sich selbst in die Liste ein.
- Varianten:
  - Der Rechner wird zufällig aus der Liste gewählt.
  - Es wird immer der zuletzt eingetragene Rechner gewählt.
  - Es wird der Rechner gewählt, der die letzte Anfrage am schnellsten beantwortet hat. (Nach dem Laden wird die dafür benötigte Zeit in der Liste eingetragen.)
- Frage: Welche Variante ist die beste?

Eine einfache Simulation (vgl. nächste Vorlesung) kann hier nützliche Einsichten liefern, schon bevor man ein Protokoll entworfen hat.

# Beispiel: RTT eines P2P Dienstes

Messung eines P2P Filesharing-Dienstes im Internet



Modell der Round-Trip-Time

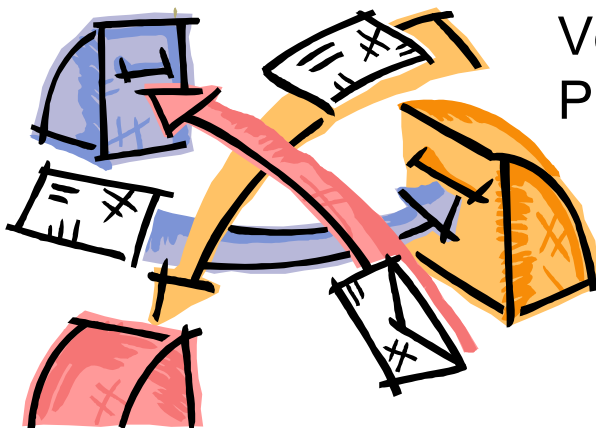
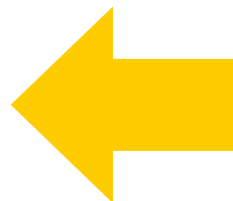
- Würfle  $x \in [0;1)$
- Falls  $x < \dots$ 
  - setze  $RTT = \dots$
  - sonst ...



Simulation



Verbessertes Protokoll

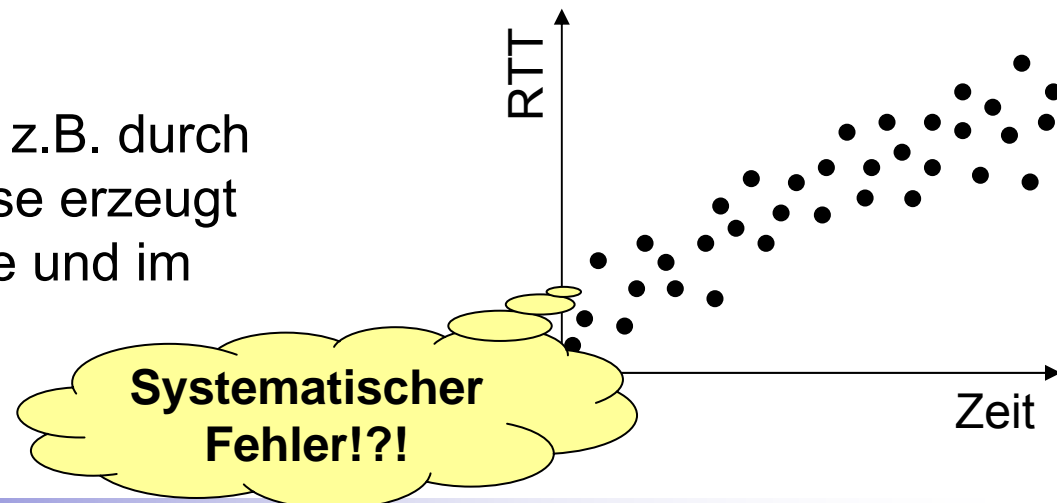
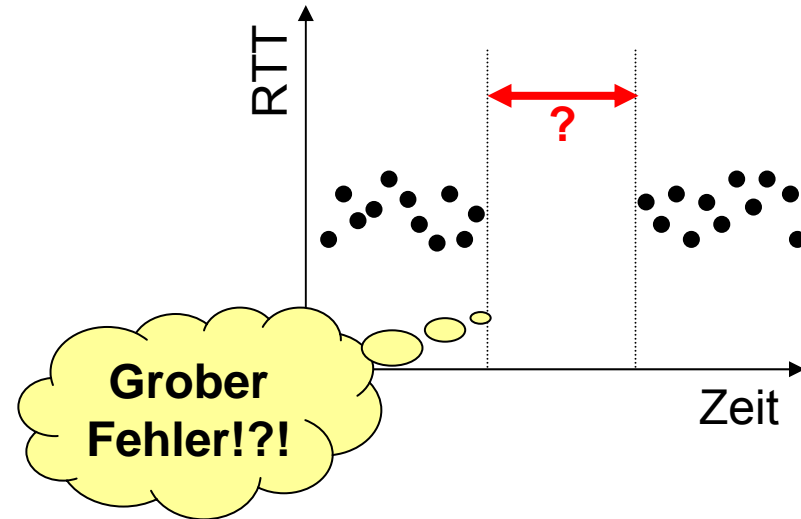


- Fehler kommen bei jeder Messung und bei den meisten Simulationen\* vor. Man unterscheidet:
  - Grobe Fehler – (Hoffentlich) leicht zu finden durch kritisches Hinterfragen
  - Systematische Fehler – Schwierig zu finden; erfordert genaue Analyse der Messung, z.B. durch Verfeinerung des Modells
  - Statistische Fehler – Unvermeidlich, aber quantitativ bestimmbar durch Fehlerrechnung

\* Der Sprachgebrauch bezeichnet auch das Gewinnen von (Mess-)werten bei einer Simulation als „Messung“. Beispiel: Diese Simulation misst die durchschnittliche Laufzeit der Nachrichten.

# Fehler – Beispiele

- Grober Fehler
  - Alte Messungen ausgewertet
  - Falsche Netzwerkschnittstelle gemessen
  - Daten des Messprogramms falsch weiterverarbeitet
  
- Systematische Fehler
  - Uhren laufen nie wirklich synchron
  - Messung selbst stört, z.B. durch Loggen der Ergebnisse erzeugt Last auf der Maschine und im Netz



# Konfidenzintervall (1)

- Statistische, also zufällige, Fehler heben sich im Mittel weg.
- Sind die statistischen Fehler **normalverteilt**, liegt der tatsächliche Wert mit 68,3% im Intervall

$$\left[ \bar{x} - \frac{\sigma}{\sqrt{n}}; \bar{x} + \frac{\sigma}{\sqrt{n}} \right]$$

Dabei ist  $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$  der Mittelwert und  $\sigma$  die Standardabweichung.

Es gilt:

$$\begin{aligned} \sigma &= \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \\ &= \sqrt{\left( \frac{1}{n} \sum_{i=1}^n x_i^2 \right) - \left( \frac{1}{n} \sum_{i=1}^n x_i \right)^2} \\ &= \sqrt{\overline{x^2} - \bar{x}^2} \end{aligned}$$

## Konfidenzintervall (2)

- Allgemein spricht man bei einem Intervall, in dem der unbekannte, zu messende Wert liegt, von einem Konfidenzintervall.
- Bei einer **Normalverteilung** der statistischen Fehler gilt:
  - Man spricht beim Intervall  $[x-\sigma, x+\sigma]$  vom 1-Sigma-Intervall.
  - Für das 2-Sigma-Intervall  $[x-2\sigma, x+2\sigma]$  beträgt die Wahrscheinlichkeit 95,4%.
  - Für das 3-Sigma-Intervall  $[x-3\sigma, x+3\sigma]$  beträgt die Wahrscheinlichkeit 99,7%.
- Statt der unübersichtlichen Intervallschreibweise schreibt man häufig auch  $x \pm \sigma$ , beispielsweise  $3,45 \pm 0,28$  Sekunden und meint damit das 1-Sigma-Intervall.
- Achtung: Mit mehr als 30% Wahrscheinlichkeit liegt der tatsächliche Wert außerhalb des 1-Sigma-Intervalls!

## Konfidenzintervall (3)

- Die Standardabweichung ist eine Eigenschaft des Zufallsprozesses der sich mit der Messung verbindet.
  - Sie kann aus vielen unabhängigen Messungen empirisch bestimmt werden (siehe oben).
  - Manchmal kann die Standardabweichung auch aus dem verwendeten Messprozess heraus bekannt sein.
- Durch häufige Wiederholung der Messung kann das Konfidenzintervall verkleinert werden, ohne dass sich die Standardabweichung des Zufallsprozesses ändern würde.
  - Alternativ: Betrachte ganzen Satz von Messungen und ordne diesem eine neue, kleinere Standardabweichung zu.
- Achtung: Eine Halbierung des Konfidenzintervalls erfordert eine Vervierfachung der Zahl der Messungen!

## Aufgabe (1)

- In jedem siebten Überraschungsei soll sich eine Spielfigur finden. Die Mitarbeiter des Telematik-Instituts finden im Selbstversuch in 288 Eier 63 Figuren.
- Stimmt die Aussage „In (mindestens) jedem siebten Ei ist eine Figur.“?
- Modellierung: Die Zufallsvariable nimmt die Werte 0 und 1 an.
- Mittelwert ist  $63/288 \approx 0.219$
- Standardabweichung ist  $\sigma = \sqrt{\frac{63}{288} - \frac{63^2}{288^2}} \approx 0.413$
- Bei 288 Messungen liegt also der wahre Wert mit 68.3% Wahrscheinlichkeit im Intervall [0.195; 0.243].
- Entsprechend mit 95.4% im Intervall [0.174; 0.267] und mit 99.7% im Intervall [0.150; 0.291].
- Die Behauptung  $1/7 \approx 0.143$  Chance ist somit mit sehr hoher Wahrscheinlichkeit erfüllt.
- Mit hoher Wahrscheinlichkeit ist sogar mind. in jedem 6. Ei eine Figur.



## Aufgabe (2)

- Angenommen weitere „Messungen“ würden weder Mittelwert noch Standardabweichung ändern.
- Wieviele Überraschungseier müssten insgesamt gegessen worden sein, damit man mit 1-Sigma-Sicherheit sagen kann, dass in mehr als jedem fünften Ei eine Figur ist?
- Erster Lösungsschritt:  
Mehr Messungen verkleinern das Konfidenzintervall.  
Wie groß muss das Konfidenzintervall sein, damit es bis zur gefragten Grenze  $1/5$  reicht?
- Teilantwort:  
Gesucht ist das Intervall  $[0.200; 0.238]$ . Dieses ist  $0.038$  breit.
- Gesamtantwort:  
Dieses Intervall ist  $0.049/0.038 = 1,3$  mal kleiner als das Konfidenzintervall bei 288 Messungen. Also braucht man  $1,3^2 * 288 = 486$  Messungen.

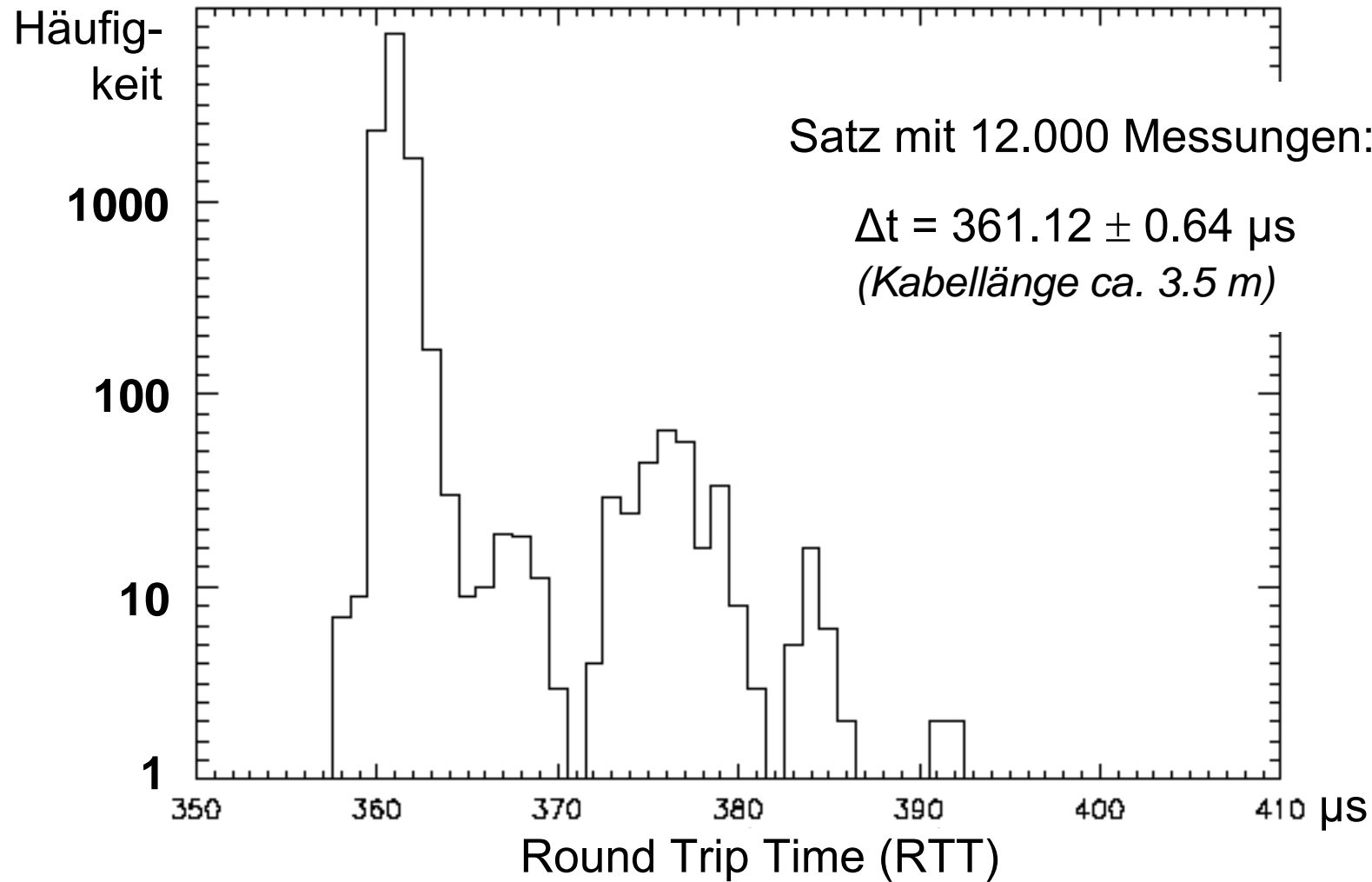
# Vorgriff: Wahrscheinlichkeitsverteilungen

- Mittelwert und Standardabweichung sind die Parameter der Normalverteilung.
- Das Beispiel mit dem „Ziehen von Ü-Eiern“ (statt Bällen aus einer Urne, wie im Lehrbuch) folgt einer Normalverteilung.
- Andere Verteilungen haben andere Parameter.
- Wenn die Messwerte bzw. die Messfehler nicht normalverteilt sind, machen alle diese Rechnungen hier keinen Sinn!
- Bei vielen Messungen sind *die Fehler* normalverteilt, bzw. sind nahe genug an einer Normalverteilung, dass man diese vereinfachte Rechnung machen darf!
- Viele Messwerte folgen aber anderen Verteilungen!
  
- Beispiel: Ein Satz Schrauben enthält je 100 Stück der Länge 3, 5, 7 und 9 Millimeter. Berechnen Sie Mittelwert und Standardabweichung. Wie viel Prozent der Schrauben haben eine Länge zwischen 5 und 7 Millimetern? – Eine Rechnung wie oben wäre hier Unsinn!

# Messung Übertragungszeit (1)

- Die mathematische Handhabung statistischer Fehler ist ein mächtiges Werkzeug, wenn genügend Messungen zur Verfügung stehen.
- Behauptung: Ein einfaches „Ping“ auf Anwendungsebene reicht aus, die Übertragungszeit (bzw. umgekehrt Lichtgeschwindigkeit) auf einem Ethernetkabel auf 7% genau zu messen.
- Probleme:
  - Übertragungszeit nur halbe Mikrosekunde bei 100 Meter Kabel
  - Ping-Anwendung rundet auf ganze Mikrosekunden
  - Betriebssystem-Overhead: mehrere Hundert Mikrosekunden
  - Schwankungen durch Betriebssystem-Scheduler
- Idee:
  - Sätze mit sehr vielen Messungen: Verkleinerung des Konfidenzintervalls
  - Unterschiedliche Kabellänge isoliert Betriebssystem-Overhead von der Übertragungszeit

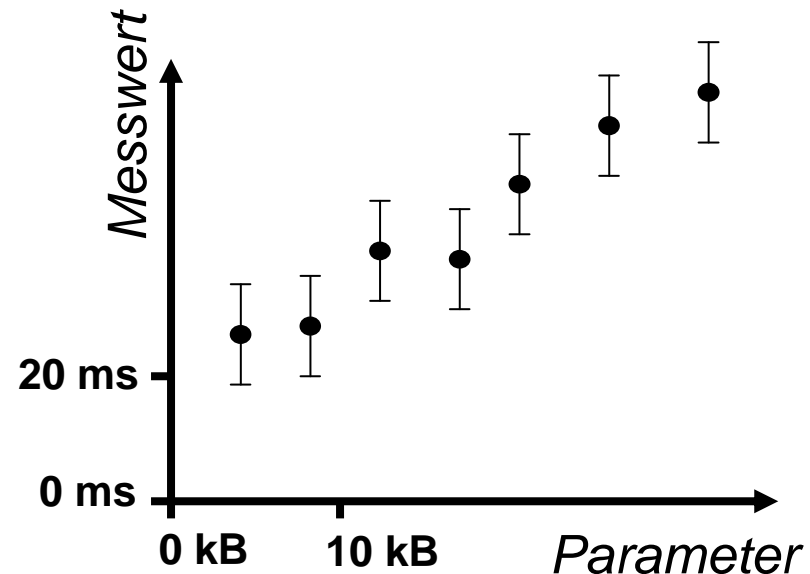
# Messung Übertragungszeit (2)



*Nach Joel Lepak und M. Crescimanno*

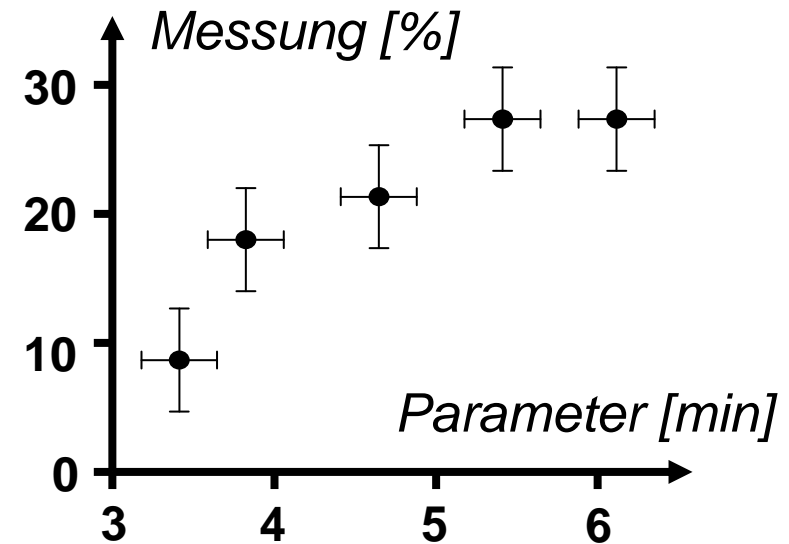
# Graphische Darstellung (1)

- Graphische Darstellungen von Messergebnissen sind meist besser „lesbar“ als Tabellen mit Zahlen
- Jede graph. Darstellung (=„Plot“) muss klar machen, was wie aufgetragen ist:
  - Achsenbeschriftung mit prägnanter Bezeichnung was aufgetragen wurde.
  - Teilstriche und Maßzahlen an den Achsen
  - Maßeinheiten, entweder direkt hinter die Zahlenwerte oder einmal pro Achse
- Falls Konfidenzintervalle bekannt und relevant sind werden diese als Fehlerbalken eingezeichnet.



## Graphische Darstellung (2)

- Falls zwei fehlerbehaftete Messgrößen gegen einander aufgetragen werden, müssen für jede Messung zwei Fehlerbalken eingezeichnet werden.
- Die Achsenbeschriftung sollte überdies das Lesen leicht machen:
  - Ist nur eine Zahl angegeben, wird man annehmen, Nullpunkt und Achsenschnittpunkt fielen zusammen. Diese Minimaldarstellung sollte vermieden werden.
  - Zwei Zahlen bezeichnen Skala und Achsenabschnitt zwar eindeutig. Die Darstellung ist aber immer noch schwer lesbar.
  - Auch die Unterscheidung in lineare und logarithmische Achsen sollte durch eine hinreichend große Anzahl von Teilstrichen und Maßzahlen klar illustriert werden.



- Setzt sich eine Messgröße aus mehreren einzelnen Messgrößen zusammen, so muss man der Fehler in der Rechnung berücksichtigen.
- Beispiele:
  - Satz von Messungen mit bekannter Standardabweichung
  - Fehler beim Messverfahren angegeben
- Für eine Größe, die sich nach der Formel

$$y = f(x_1, \dots, x_k)$$

berechnet, ergibt sich der zugehörige Fehler als

$$\Delta y = \sum_{i=1}^k \left| \frac{\partial f}{\partial x_i} \right| \Delta x_i$$

## Fehlerrechnung – Beispiel

- Während  $14.2 \pm 0.3$  Sekunden werden  $97.3 \pm 0.7$  Kilobyte Daten übertragen. Welche Datenrate wurde erzielt?
- Lösung:  
Mittelwert berechnet sich nach  $r = s / t = 6.85$  kB/s. Fehler wird durch Fehlerrechnung ermittelt:

$$\begin{aligned}\Delta r &= \frac{1}{t} \Delta s + \frac{s}{t^2} \Delta t \\ &= \frac{0.7 \text{ kB}}{14.2 \text{ sec}} + \frac{97.3 \text{ kB} \cdot 0.3 \text{ sec}}{(14.2 \text{ sec})^2} \\ &= 0.19 \text{ kB/sec}\end{aligned}$$

- Antwort:  $6.85 \pm 0.19$  kB/s



# Beispiel Übertragungszeit

- Per „Ping“ auf Anwendungsschicht wurde Round Trip Time für verschiedene Kabellängen bestimmt.
- Die Signalgeschwindigkeit ergibt sich als

$$c = d / (t_{\text{mess}} - t_{\text{overhead}})$$

- Die Messung ergab für  $d = 3.53 \pm 0.01$  Meter die Round Trip Time

$$t = 361.12 \pm 0.64 \text{ Sekunden.}$$

- Wie bestimmt man die Signalgeschwindigkeit?
- Lösung: Mehrere Messungen und Eliminieren der Overhead-Zeit.

# Anpassung an Messwerte

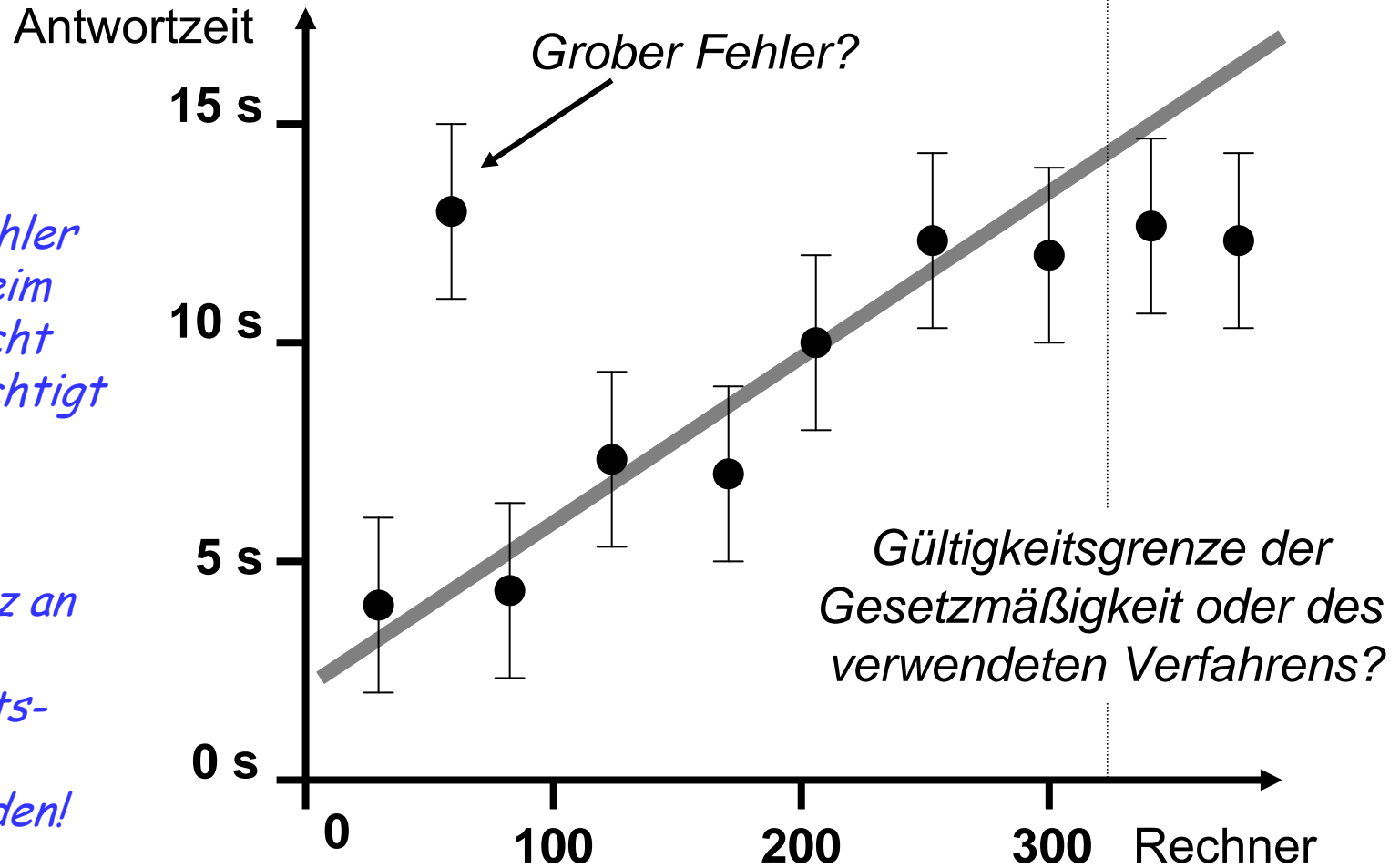
---

- Die Fehlerrechnung bestimmt eine Größe wenn alle zu ihrer Berechnung nötigen Größen bzw. Parameter bekannt sind.
- Oft sind nicht alle Parameter bekannt, sondern müssen aus den Messwerten selbst bestimmt werden.
- Lösung:  
Anpassen (=„Fitten“) einer parametrisierten Kurve an die Messwerte
  - graphisch mit Augenmaß
  - rechnerisch nach der Methode der kleinsten Quadrate
- Achtung:
  - Das reine graphische Auftragen von Messwerten sollte eine neutrale Wiedergabe von Fakten sein.
  - Das Fitten an einen vermuteten Zusammenhang ist eine Argumentation für diesen Zusammenhang.

# Graphisches Fitten (1)

*Grobe Fehler sollten beim Fitten nicht berücksichtigt werden.*

*Ebenso: Datensatz an den Gültigkeitsgrenzen abschneiden!*

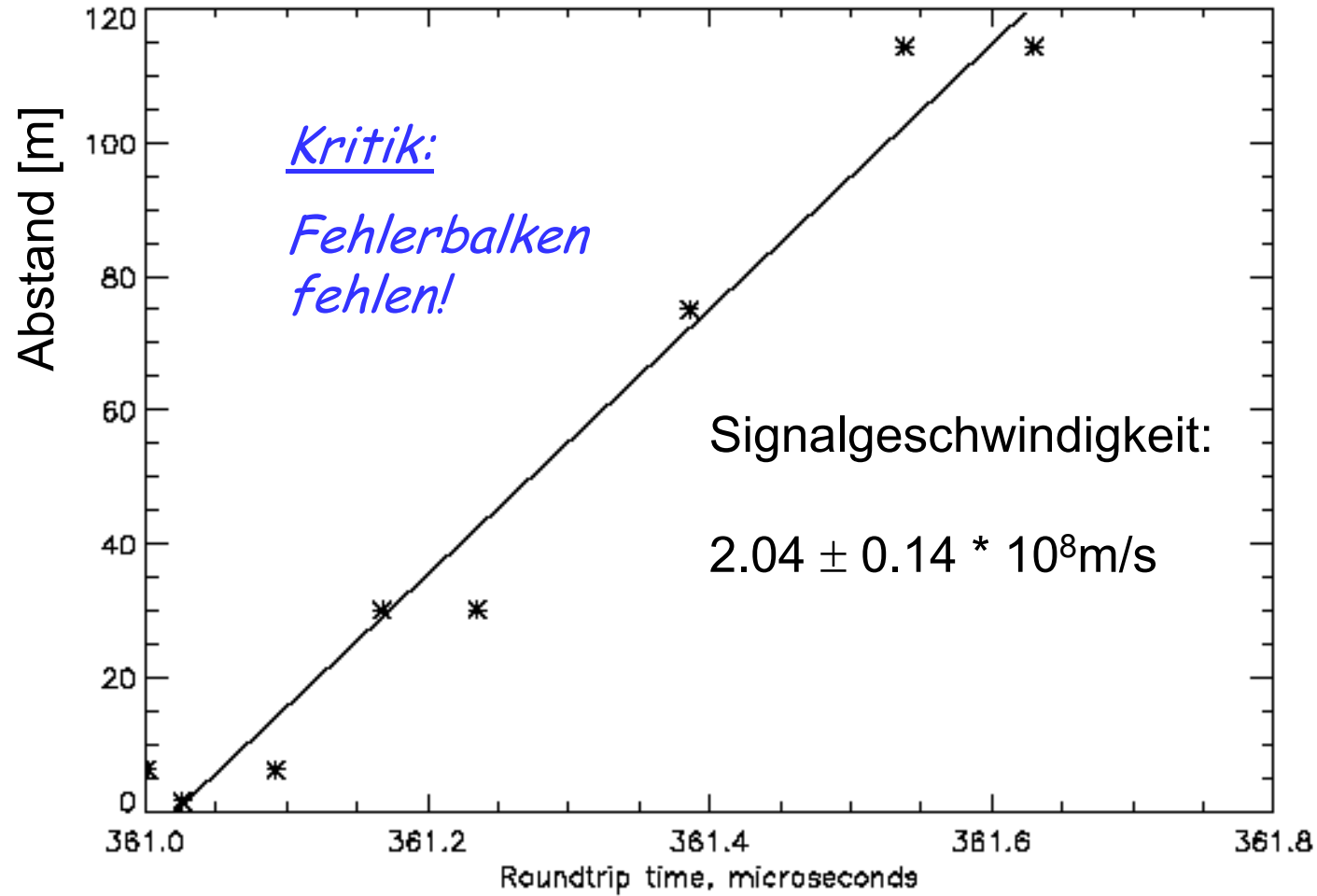


## Graphisches Fitten (2)

---

- Das menschliche Auge ist gut geeignet, um
  - Zusammenhänge aus einem Plot abzulesen
  - Geraden an Messpunkte anzufitten
- Achtung: Das Auge erkennt oft mehr Struktur, als in der Messung enthalten ist.
  - Beispiel: Sternbilder! – Eigentlich gibt es dort keine Strukturen, aber das Auge bildet sie sich trotzdem ein.
- Die Beurteilung, ob der vermutete Zusammenhang mit den Messdaten zusammenpasst, muss mit Hilfe der Fehlerbalken geschehen:
  - Faustregel: Schneiden ca. 70% der Fehlerbalken die Gerade?
  - Häufig gibt es eine deutlich bessere Übereinstimmung. Dies deutet auf eine zu konservative Abschätzung der Fehler hin!

# Messung Übertragungszeit (3)



*Nach Joel Lepak und M. Crescimanno*

# Methode der Kleinsten Quadrate

- Nach *Carl Friedrich Gauss*
- Wenn Fehler normalverteilt sind, geht der Abstand des Messergebnisses vom vermuteten tatsächlichen Wert wie folgt in die Wahrscheinlichkeit ein:

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(\xi - \bar{x})^2}{2\sigma^2}}$$

- Sind die Messungen unabhängig, multiplizieren sich die Wahrscheinlichkeiten.
- Damit die Wahrscheinlichkeit maximal wird, muss das Argument der Exponentialfunktion minimal werden:

$$\sum (x_i - \bar{x})^2 \stackrel{!}{=} \min$$

# Lineare Regression

- Spezialfall der Methode der kleinsten Quadrate
- Anfitzen an eine lineare Gesetzmäßigkeit:  $y = a x + b$
- Daraus folgt:

$$\sum (y_i - ax_i - b)^2 \stackrel{!}{=} \min$$

- Partielles Ableiten nach a und b ergibt:

$$a = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2}$$
$$b = \frac{\bar{y} \sum (x_i - \bar{x}) - \bar{x} \sum (y_i - \bar{y})}{\sum (x_i - \bar{x})}$$

# Einschub: Noch mal kleinste Quadrate

- Falls  $x$  und  $b$  die gleiche Dimension haben ist die Vektorgleichung  $Ax = b$  durch Multiplikation mit der inversen Matrix im allgemeinen eindeutig lösbar.

- Falls  $\dim x < \dim b$  ist die Gleichung überbestimmt und man verwendet folgende Verallgemeinerung:

$$A\vec{x} = \vec{b}$$

$$A^T A\vec{x} = A^T \vec{b}$$

$$\vec{x} = (A^T A)^{-1} A^T \vec{b}$$

- Dies entspricht einem Fitten nach der Methode der kleinsten Quadrate.
- Beweis: Nachrechnen indem man  $\text{grad } |Ax - b|^2 = 0$  nach  $x$  auflöst!

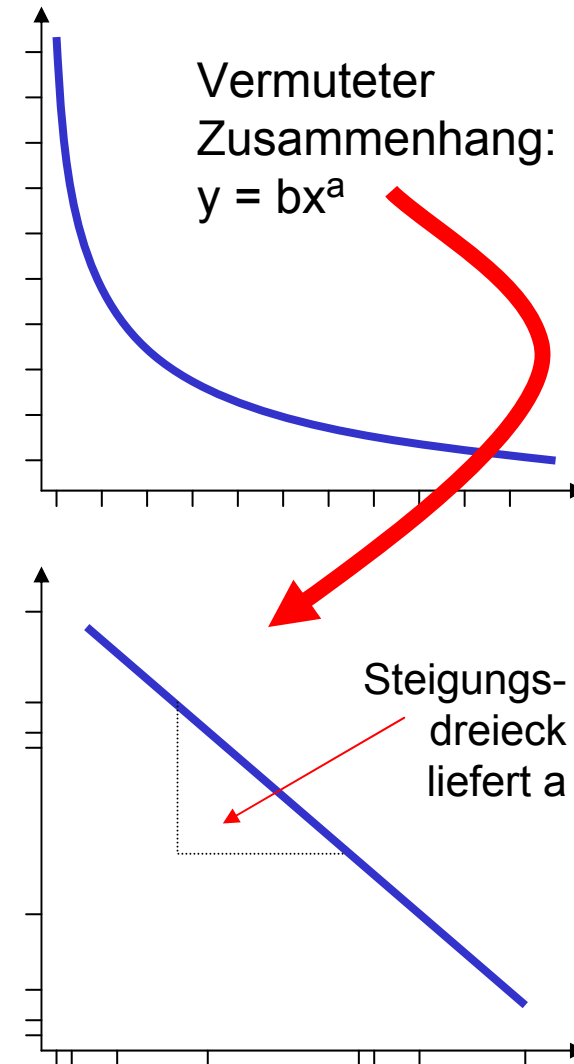
*Wiederholung  
aus der Numerik*



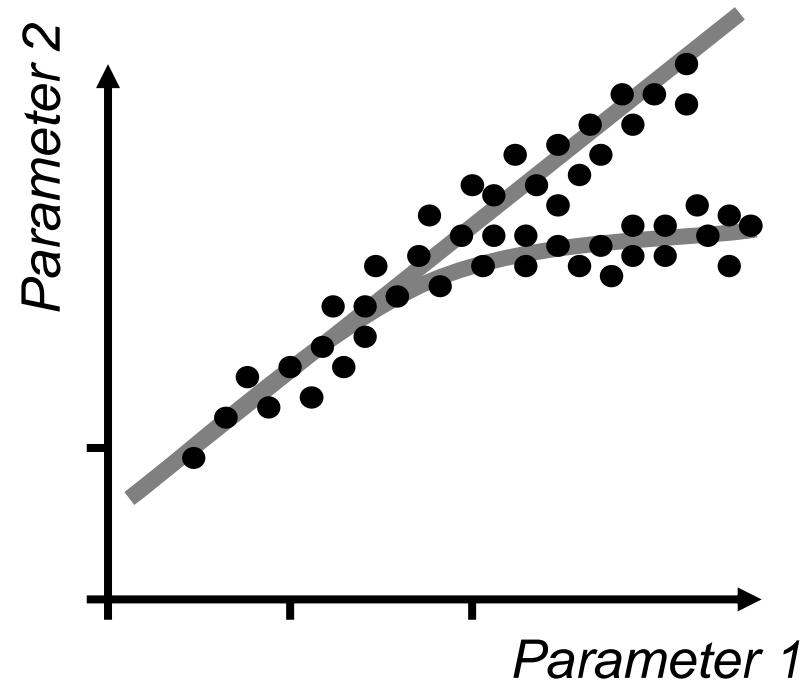
- In der Praxis verwendet man bei nicht linearen Gesetzmäßigkeiten manchmal eine Transformation auf einen linearen Zusammenhang.
- Dabei müsste aber auch die Verteilung der Fehler transformiert werden. Die Eigenschaft der Normalverteilung ginge verloren!
- Oft weiß man aber sowieso nichts über die Verteilung der Fehler. Man rechnet nur der Einfachheit halber mit einer Normalverteilung.
- Was tun?
  - Untersuchen, in welcher Form statistische Fehler in die Messung hineinkommen, d.h. in welcher Transformation die Fehler vermutlich einer Normalverteilung unterliegen.
  - Beispiel: Fehler sind prozentual zum Messwert. Dann ist eine Transformation auf eine logarithmische Achse sinnvoll. (Vgl. auch A-Law oder  $\mu$ -Law in der Multimediatechnik.)
  - Alternativ: Graphisch Fitten per Augenmaß

# Darstellungsformen

- Die drei hauptsächlichsten Darstellungsformen (linear, logarithmisch, doppel-logarithmisch) zeigen jeweils typische Zusammenhänge als Gerade:
  - Lineare Achsen zeigen  $y = ax+b$  als Gerade.
  - Logarithmische Darstellung zeigt  $y = be^{ax}$  als Gerade.
  - Doppel-logarithmische Darstellung zeigt  $y = bx^a$  als Gerade.
- Am besten wählt man die Darstellungsform, die zu einer Geraden führt.

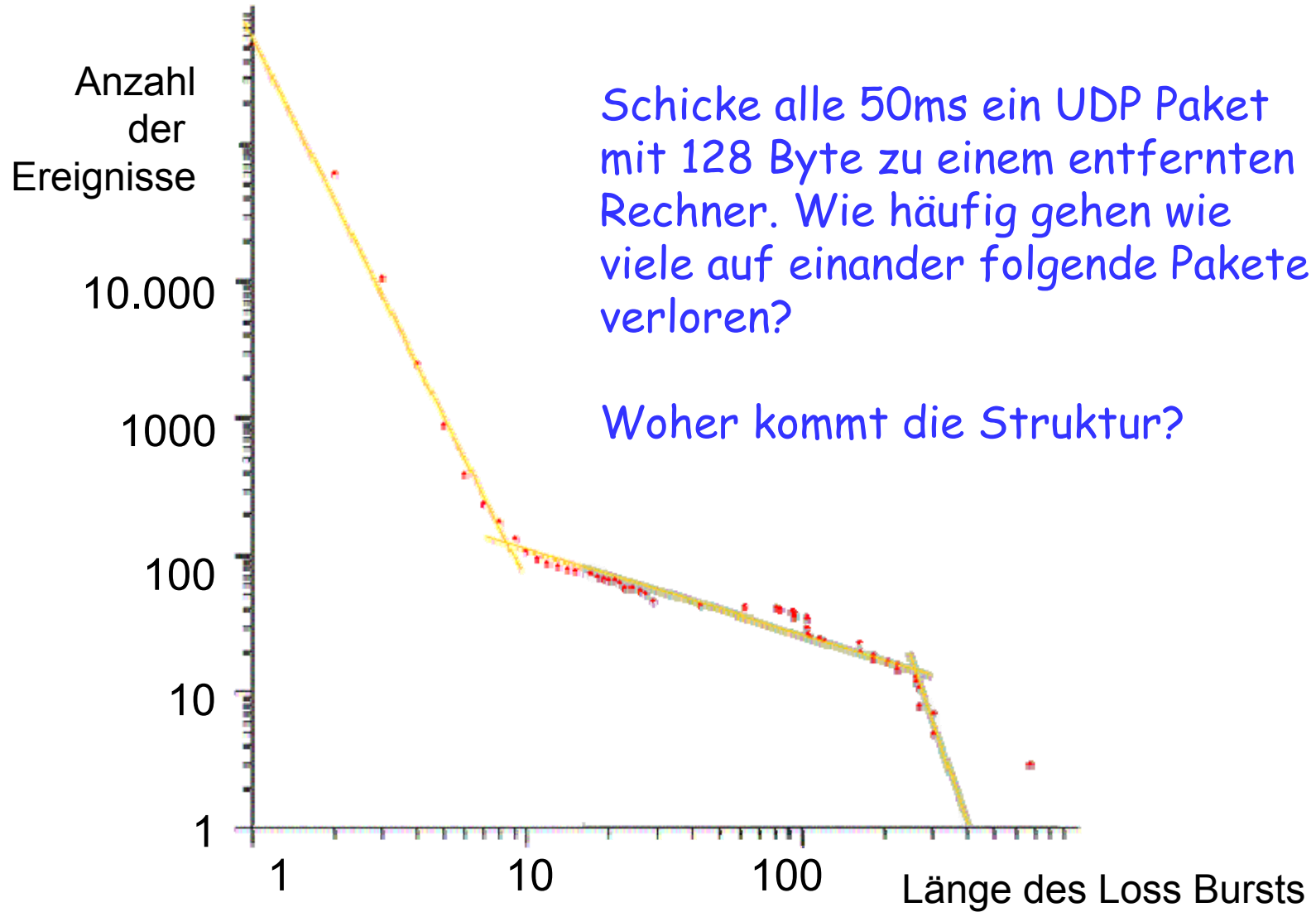


- Manchmal kann man bei dem zu untersuchenden System verschiedene Anteile unterscheiden.
- In diesen Fällen sollten die Anteile aufgrund des entsprechenden Unterscheidungskriteriums getrennt werden und jeweils für sich weiter untersucht werden.



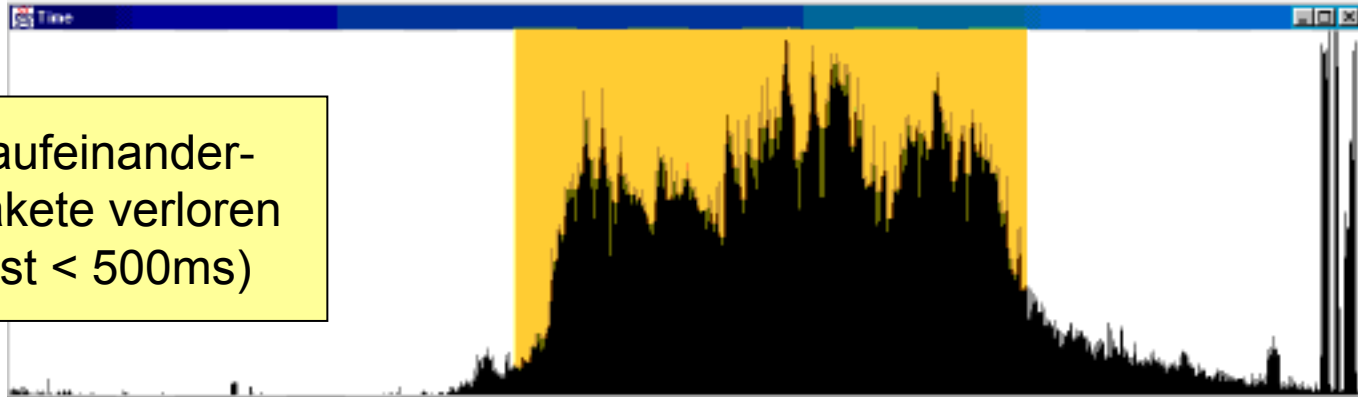
Eine gute Klassifikation erfordert ein Verständnis des Modells!  
Dann kann man daraus aber viele neue Einsichten zur  
Protokollverbesserung gewinnen!

# Klassifikation – Beispiel (1)

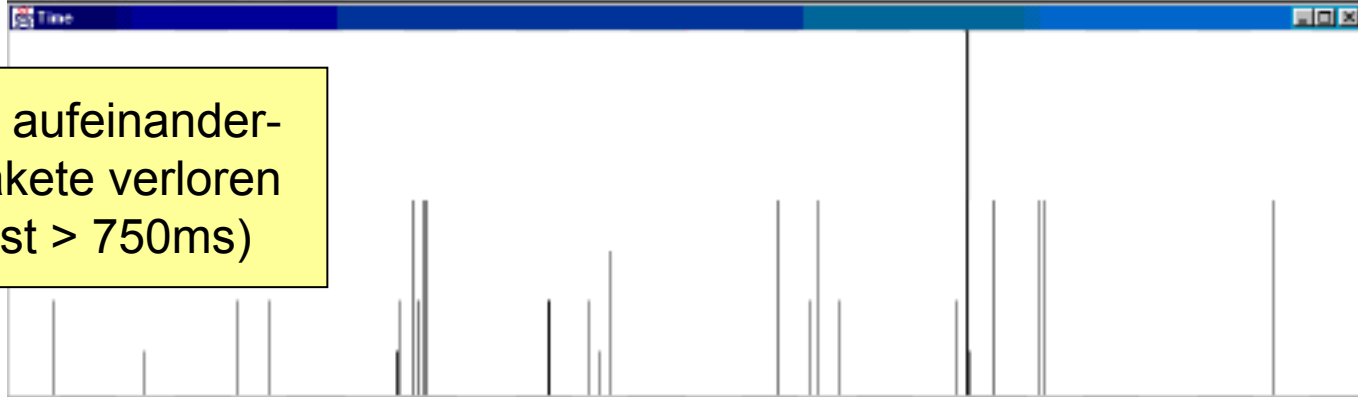


# Klassifikation – Beispiel (2)

Bis zu 10 aufeinanderfolgende Pakete verloren  
(d.h. Verlust < 500ms)



Mehr als 15 aufeinanderfolgende Pakete verloren  
(d.h. Verlust > 750ms)



24h Zeitachse

- Messungen sind unerlässlich, um Modelle zu verfeinern
  - Modell analysieren und Vorhersagen mit Messung vergleichen
  - Messung im Modell abbilden
- Jede Messung ist mit Fehlern behaftet
  - Grobe Fehler – Beheben durch kritisches Hinterfragen
  - Systematische Fehler – Aufspüren und im Modell reflektieren
  - Statistische Fehler – Beherrschen durch Fehlerrechnung
- Fit an vermutete Gesetzmäßigkeit: Methode der kleinsten Quadrate
- Wichtiges Hilfsmittel beim Auswerten ist das menschliche Auge
  - Daten auftragen und Strukturen suchen
  - Achtung: Das Auge sieht mehr als tatsächlich da ist!
  - Vermutete Gesetzmäßigkeit bestimmt Art der Auftragung
  - Ggf. Messungen weiter klassifizieren, z.B. Tageszeit, etc.

---

# *Simulationen*



- Nicht immer lässt sich ein Modell durch Messung am realen System überprüfen:
  - Bei der Protokollentwicklung liegt noch keine Implementierung vor.
  - Selbst wenn prototypische Implementierung vorhanden ist, ist das Testnetz zu klein, um realistische Aussagen machen zu können.
  - Auch bei weit verbreiteten Protokollen sind die nötigen Messdaten in der Praxis nicht immer zugänglich.
- Simulationen bilden die *relevanten* Aspekte eines Systems in einem algorithmisch („ausführbaren“) Modell ab.
  - Was ist relevant?
    - siehe Warnungen beim Kapitel „Modellierung“
  - Welche mathematischen Grundlagen gibt es?
  - Wie sieht ein „ausführbares“ Modell aus?



# Grundtypen von Simulationen

---

- Simulation kontinuierlicher Systeme durch numerische Integration
  - Beispiel: Strömungsmechanik, Elektrodynamik, ...
  - Erzeugt zufällige Anfangskonfiguration
  - Ablauf danach eindeutig bestimmt
  - Fehler durch endliche Rechengenauigkeit und Güte der Integrationsschritte bestimmt
- Simulation diskreter Systeme durch diskrete Ereignissimulation
  - Grundannahme: Alles im System passiert zu jeweils genau definierbaren Zeitpunkten (=Punkt ereignisse)
  - Beispiel: Packet vollständig empfangen, Paket vollständig gesendet
  - Ereignisse können andere Ereignisse verursachen
  - Art und Zeitpunkt dieser Ereignisse ggf. durch Zufallsgenerator bestimmt

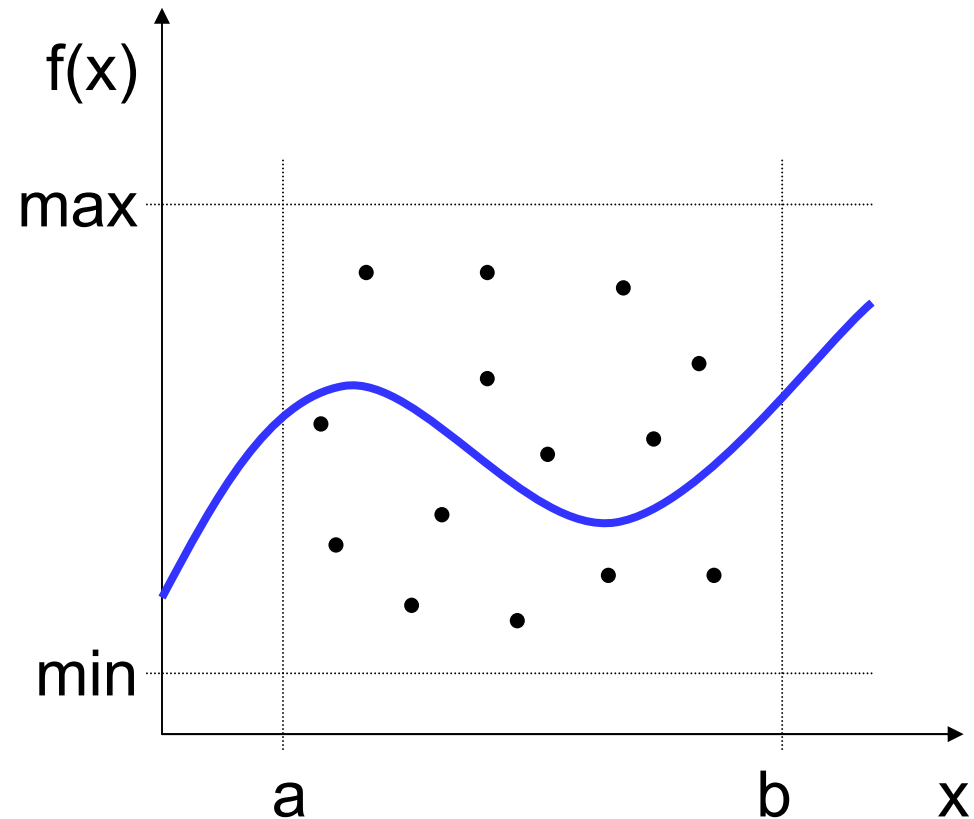
## Numerische Bestimmung einzelner Modellaspekte

- Beispiel: Numerische Integration mittels Monte-Carlo-Methode
  - Würfle gleichverteilte Zufallszahlen und teste das Ergebnis gegenüber einer vorgegebenen Funktion
  - Dabei wird die Fläche unter einer Kurve ausgemessen
  - Vorteil: Nicht immer muss die Funktion explizit gegeben sein!
- Beispiel: Simulation eines Markov-Ketten-Modells
  - Siehe Leistungsanalyse in der Telematik-Vorlesung: numerische Bestimmung des stationären Zustands
  - Wichtig: numerische Probleme beachten, z.B. Differenzen großer Zahlen erzeugen große Rundungsfehler
- Vorteil:
  - Einfach „Aufzusetzen“ der Simulation, d.h. keine Programmierung des Protokolls erforderlich
  - Oft keine Einschwingphase (siehe später)

# Numerische Integration (1)

## Monte Carlo Methode:

- Ziehe Tupel von gleichverteilten, unabhängigen Zufallszahlen  $(x,y)$  aus  $[a,b]$  und  $[\min,\max]$
- Bestimme Häufigkeit  $p$ , mit der  $y < f(x)$ .
- Integral von  $f(x)$  ist  
$$(p \max - (1-p) \min) (b-a)$$



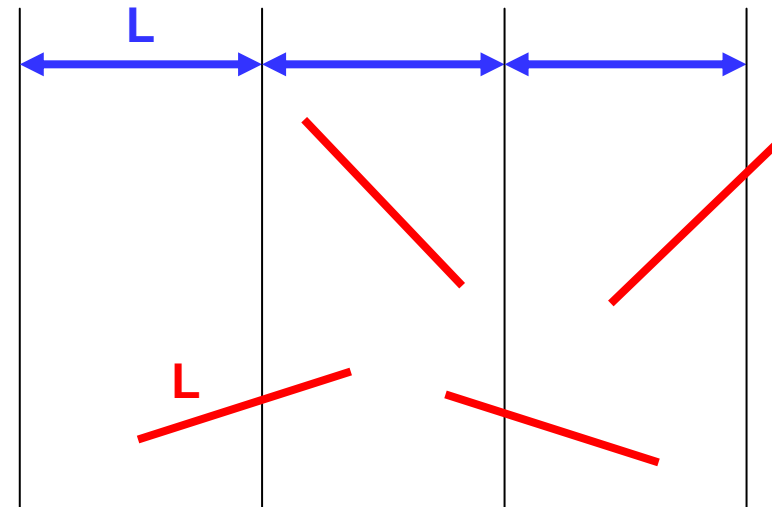
Nach diesem Schema können viele Erwartungswerte (=Integrale) experimentell bestimmt werden, auch ohne dass der modellierte Zusammenhang als geschlossene Formel beschrieben werden muss.

## Beispiel: $\pi$ empirisch bestimmen (1)

- In einem Raum besteht der Fußboden aus Brettern mit konstanter Breite  $L$ .
- Sie lassen einen Stift der Länge  $L$  auf den Boden fallen, so dass Winkel und Mittelpunkt gleichverteilt sind (=realistische Annahme).
- Mit welcher Wahrscheinlichkeit liegt ein Stift auf zwei Brettern?

- Simulation\*:

- $x \in [0;1)$  und  $\alpha \in [-\pi/2;\pi/2)$  gleichverteilt
- Zähle, wie oft  $x + \cos \alpha > 1$



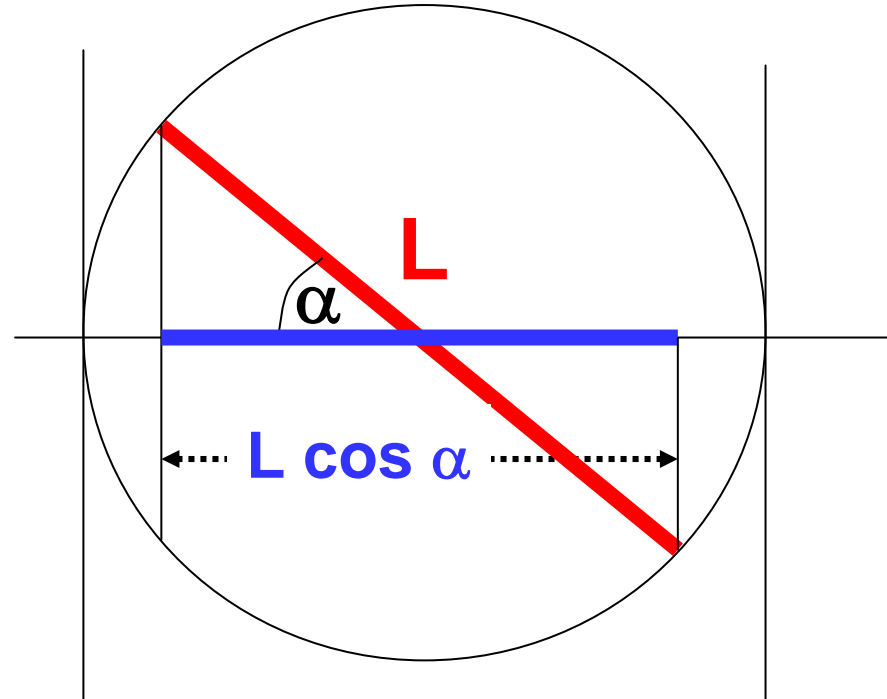
- \* Genaugenommen nur numerische Integration nach Monte-Carlo. Meist aber auch als Simulation bezeichnet.

## Beispiel: $\pi$ empirisch bestimmen (2)

- In diesem Beispiel ist auch eine analytische Behandlung möglich:

$$\frac{1}{2\pi} \int_0^{2\pi} |L \cos \alpha| d\alpha = \frac{4L}{2\pi}$$

- Hat man durch Messung oder Simulation die relative Häufigkeit bestimmt kann man damit umgekehrt einen Parameter des Modells bestimmen.
- Oft dient dieses einfache Beispiel dazu,  $\pi$  zu bestimmen.

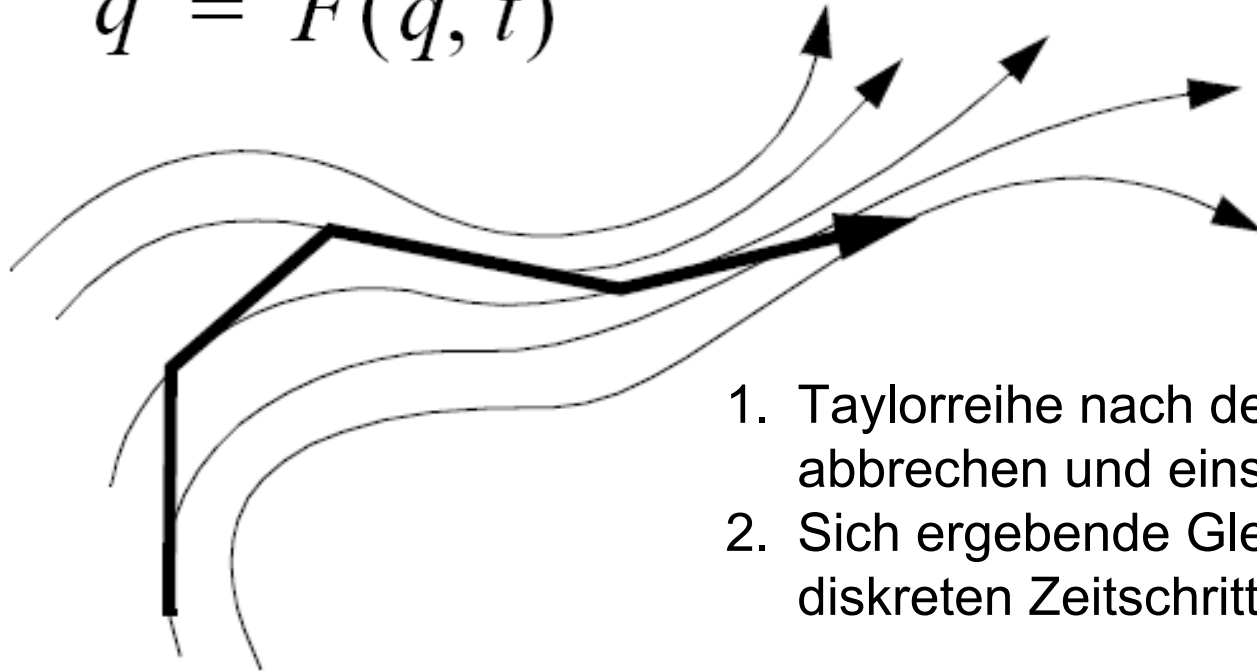


Solche einfachen „Simulationen“ können verwendet werden, um Modell-Parameter zu bestimmen.

## Numerische Integration (2)

Differentialgleichungen lassen sich durch numerische Integration (leicht) lösen:

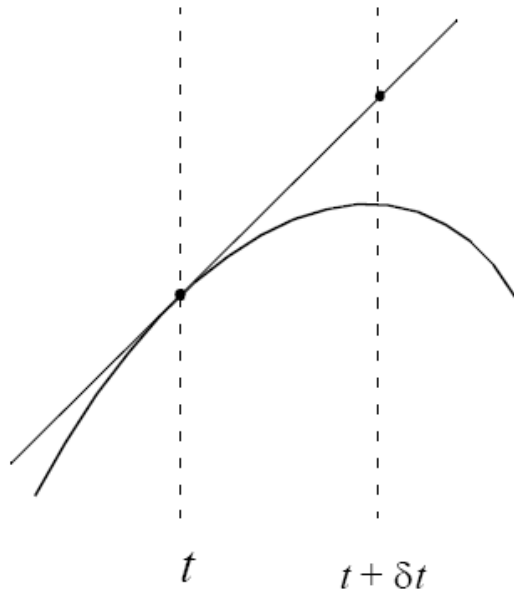
$$\dot{q} = F(q, t)$$



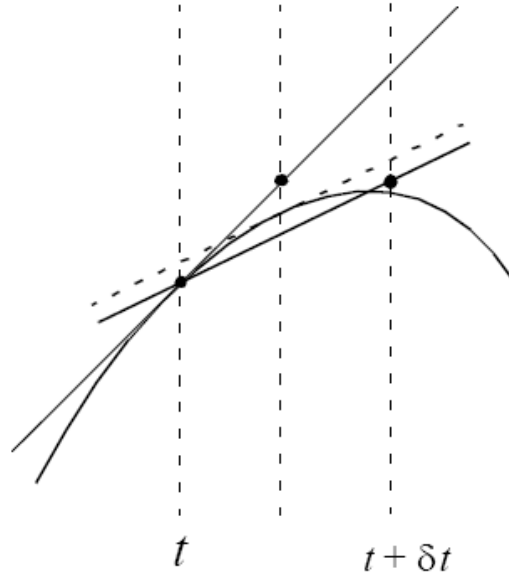
1. Taylorreihe nach dem zweiten Glied abbrechen und einsetzen.
2. Sich ergebende Gleichung kann (in diskreten Zeitschritten) iteriert werden.

$$\begin{aligned} q(t + \delta t) &= q(t) + \dot{q}(t)\delta t \\ &= q(t) + F(q(t), t)\delta t \end{aligned}$$

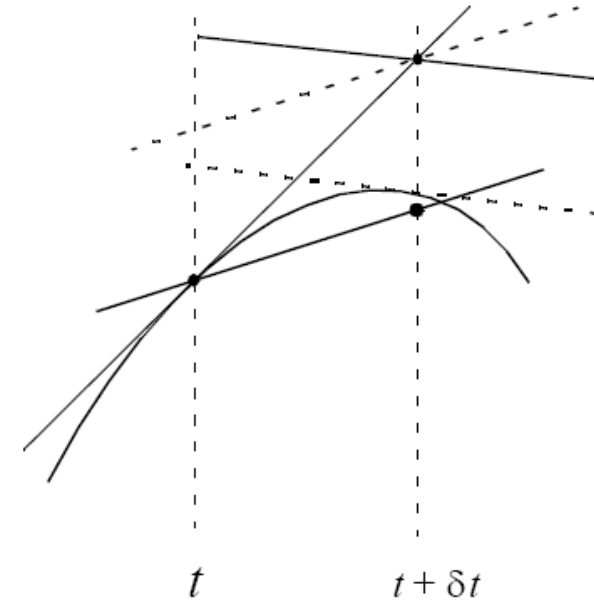
# Numerische Integration (3)



Einfaches Eulerverfahren



Modifiziertes Eulerverfahren



Verbessertes Eulerverfahren

Dieses Verfahren (Taylorreihe abbrechen und iterieren) kann weiter verfeinert werden, um die Genauigkeit zu steigern (vgl. z.B. Runge-Kutta). Weitere Verbesserungen betreffen z.B. eine adaptive Schrittweitenregeln, d.h.  $\delta t$  wird angepasst.

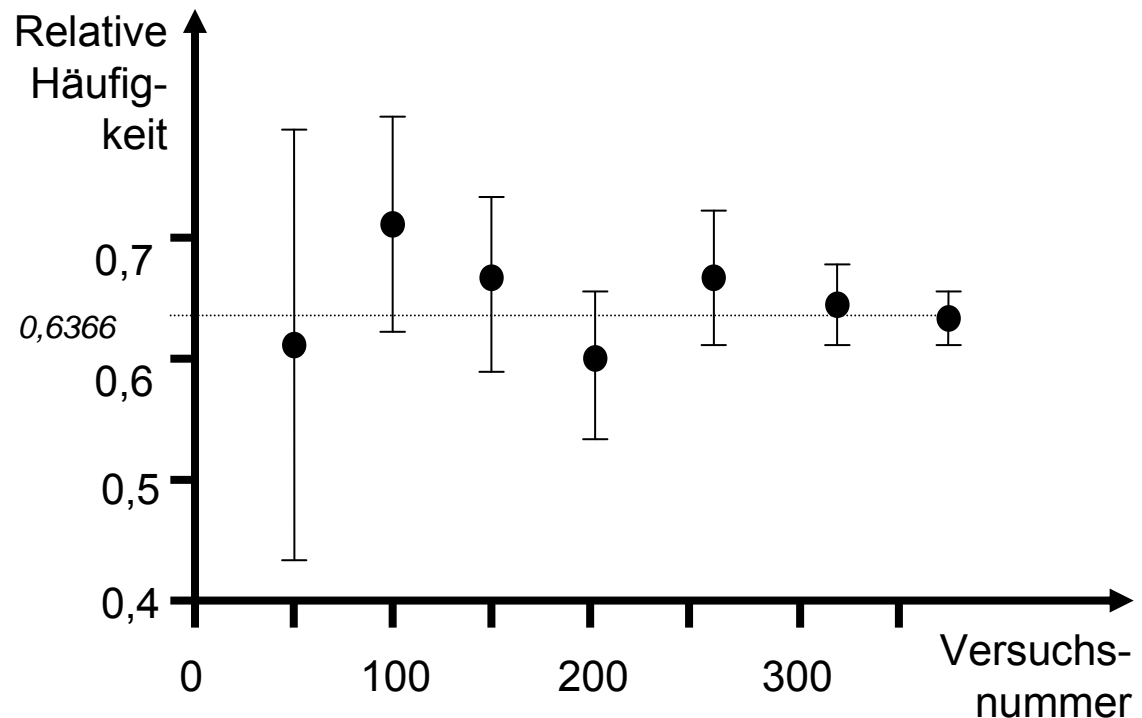
- Diese Grundidee (aus der numerischen Intergration) ist die Grundlage allgemeiner Simulationsverfahren:
  - Simuliertes System (beschrieben durch ein Modell) hat einen (zeitabhängigen) Zustand.
  - Eine mathematische Formel bzw. eine Rechenvorschrift (d.h. Algorithmus) überführen den Zustand zum Zeitpunkt  $t$  in den Zustand zum Zeitpunkt  $t+\delta t$ . (Statt Zeit könnte auch entlang eines anderen Parameters simuliert werden.)
  - Die Güte der Simulation hängt von der Güte des Modells und z.B. der Rechengenauigkeit ab.
- Dieses Verfahren setzt voraus, dass man einen Anfangszustand kennt, von dem aus man die Simulation beginnen kann.
- Häufig kennt man aber keinen realistischen Anfangszustand.



# Einschwingphase (1)

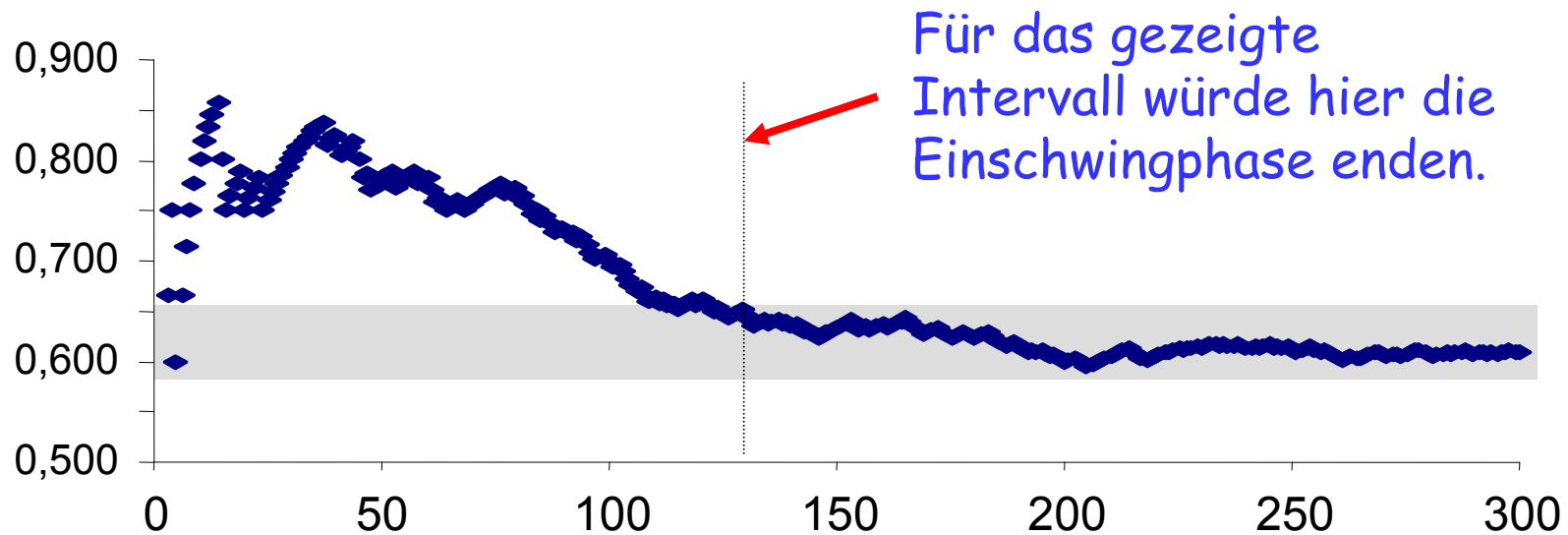
- Während die Simulation läuft, nähert sich die jeweils bestimmte relative Häufigkeit allmählich dem Erwartungswert an.
- Wie im Beispiel der Überraschungseier ergibt sich aufgrund der Unabhängigkeit der Ereignisse eine Binomialverteilung, die im Grenzwert in eine Normalverteilung übergeht.

- Man kann Fehlerbalken bestimmen und auftragen.
- Daraus lässt sich auch bestimmen, wie viele Versuche man für eine gegebene Genauigkeit durchführen muss.

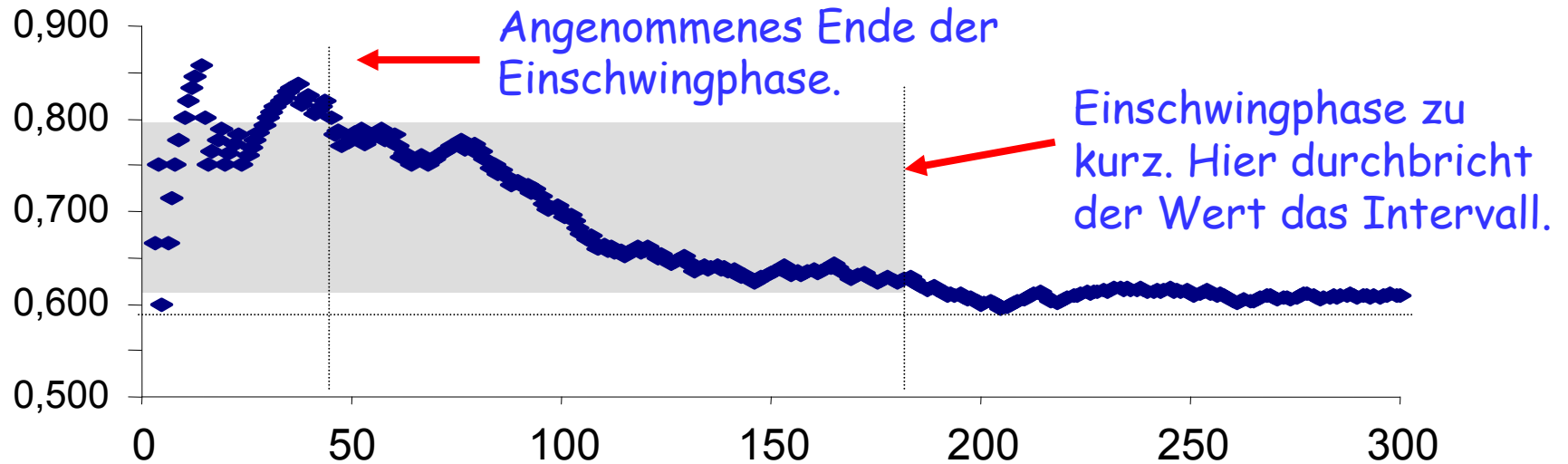
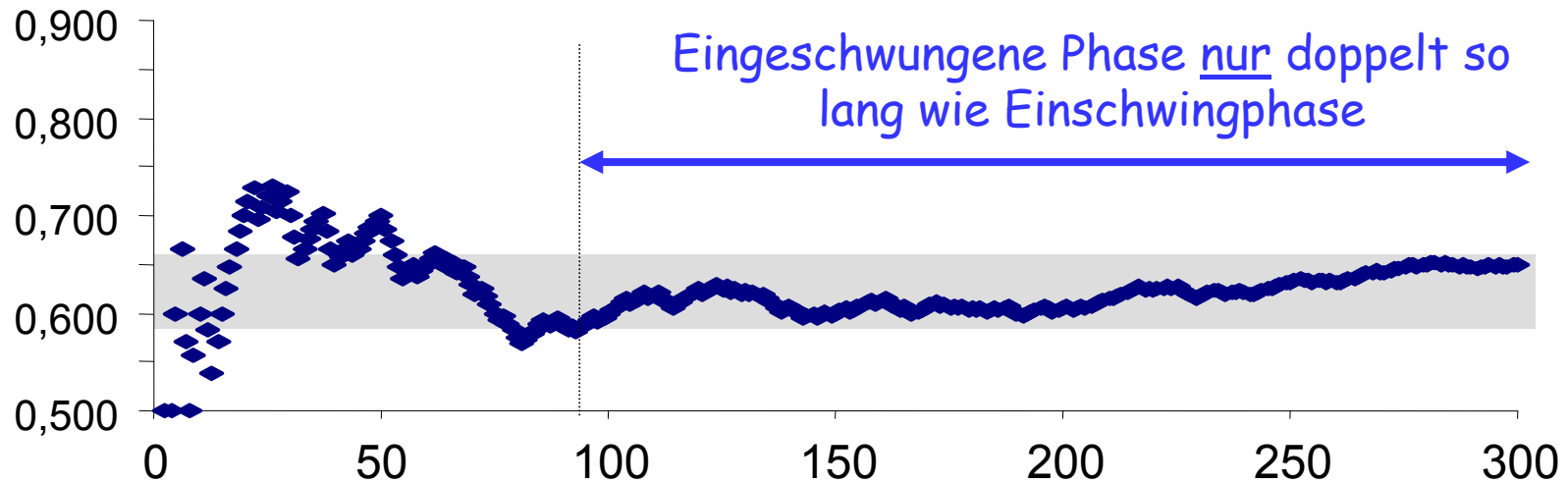


## Einschwingphase (2)

- Auch ohne Berechnung der Fehlerbalken kann man abschätzen, wo sich der tatsächliche Wert befindet.
- Für eine gegebene Genauigkeit legt man das Konfidenzintervall so in die Messungen hinein, dass möglichst viele (der letzten) Werte in diesem Intervall liegen
- Die gesamte Simulation sollte dabei deutlich länger sein als die Einschwingphase



# Einschwingphase (3)

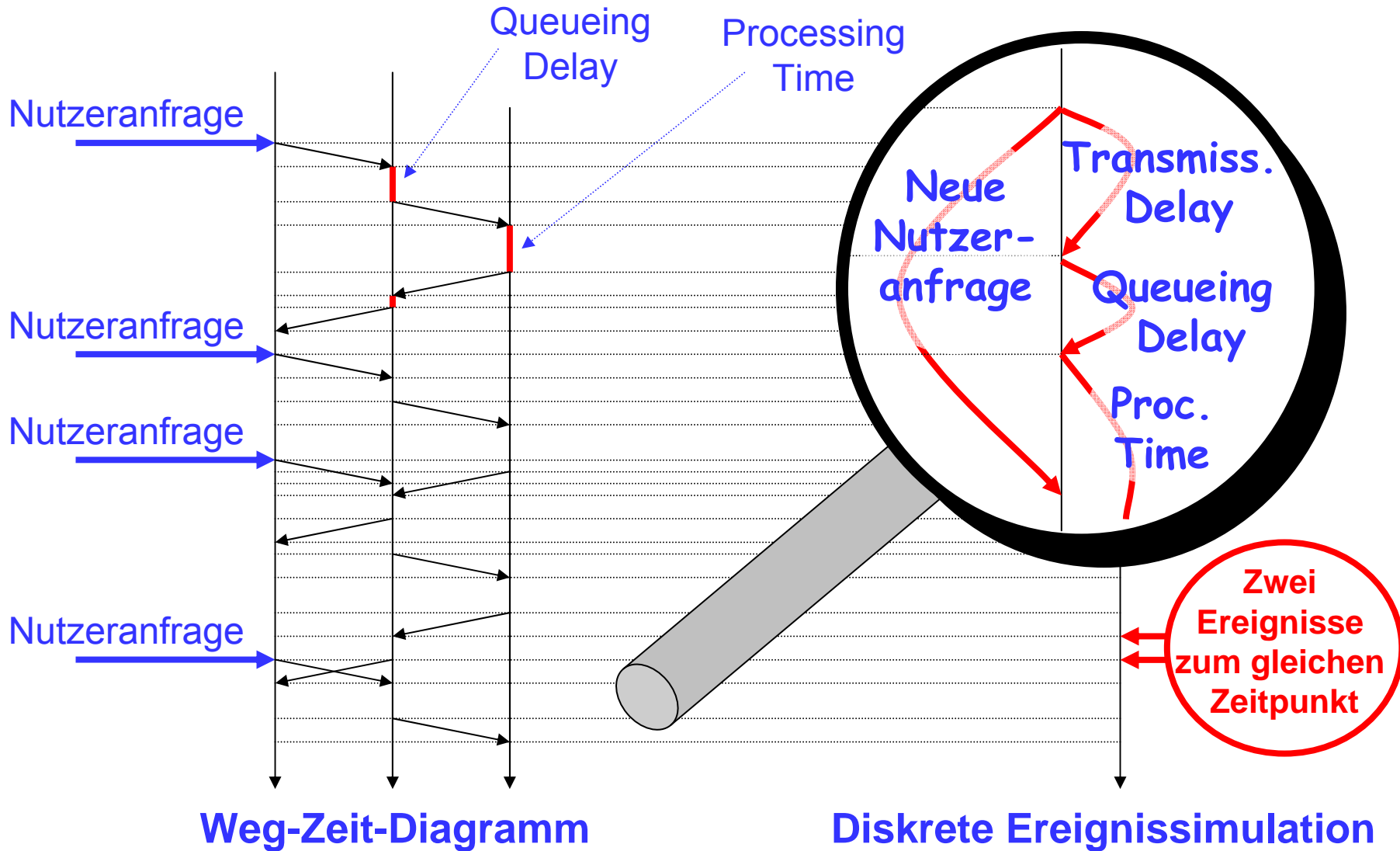


# Diskrete Ereignissimulation

---

- Verfeinerung der Grundidee der numerischen Integration.
- Allgemeiner Ablauf:
  - Liste (oder andere Datenstruktur) für Ereignisse und ihre jeweiligen Zeitpunkte
  - Frühestes Ereignis aus Liste entnehmen
  - Bearbeiten, d.h. gemäß des simulierten Protokollablaufs neue Ereignisse generieren und in Liste eintragen
  - Wieder von vorn ...
- Während der Simulation protokolliert man die jeweils interessanten Parameter
  - Für Mittelwerte müssen im Allgemeinen die Zeitstempel der entsprechenden Ereignisse mit ausgewertet werden

# Beispiel-Ablauf (1)



## Beispiel-Ablauf (2)

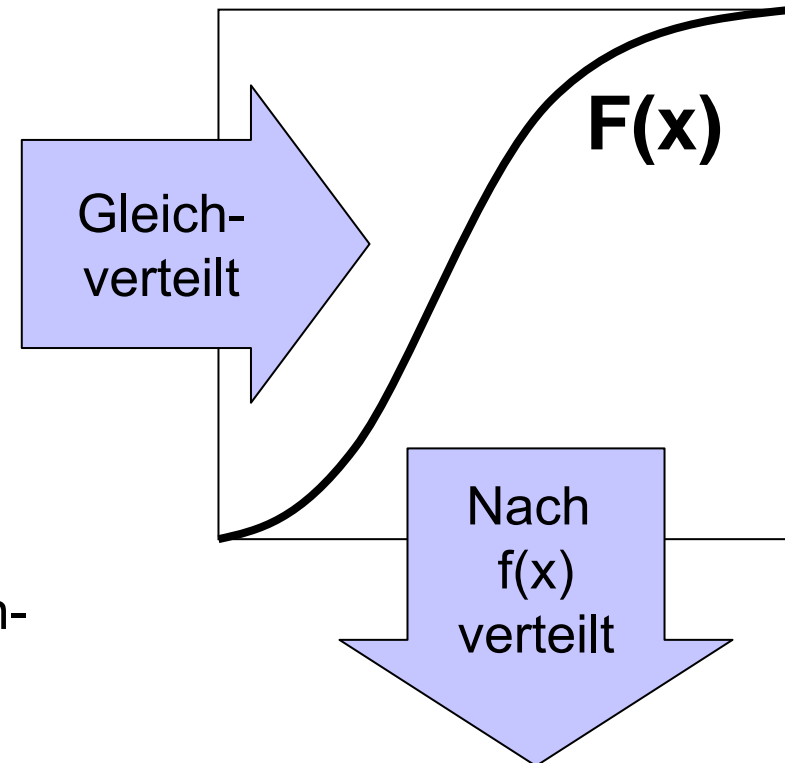
---

- Start der Simulation mit Nutzeranfrage zum Zeitpunkt  $t=0$
- Dieses Ereignis bewirkt zwei neue Ereignisse, für die Zeitpunkte eingetragen werden müssen:
  - Auswürfeln des Zeitpunkts, wann die nächste Nutzeranfrage kommt, z.B. Modellierung durch Poisson-Prozess.
  - Berechnung des Zeitpunkts, wann Nutzeranfrage vollständig vom ersten Zwischensystem empfangen wurde (Packetization Delay und Transmission Delay).
- Entsprechend können alle Ereignisse ihrerseits wieder weitere Ereignisse (eines oder mehrere) auslösen.
- Allerdings löst nicht jedes Ereignis immer weitere Ereignisse aus, z.B. Warteschlange voll, Paket verworfen.
- Bei der diskreten Ereignissimulation kann es passieren, dass zwei Ereignisse zum exakt gleichen Zeitpunkt zu bearbeiten wären.

- Zeitstempelkonflikte können wie folgt aufgelöst werden:
  - Gleichzeitige Ereignisse werden beim Eintragen hintereinander gereiht (FCFS-Strategie), d.h. Datenstruktur unterstützt geordnete Liste von Ereignissen mit dem selben Zeitstempel.
  - Gleichzeitige Ereignisse werden beim Bearbeiten zufällig ausgewählt und bearbeitet, d.h. Datenstruktur unterstützt ungeordnete Liste von Ereignissen mit dem selben Zeitstempel.
  - Existiert beim Eintragen bereits ein Ereignis mit diesem Zeitstempel wird der einzutragende Zeitstempel um eine Zeiteinheit erhöht, d.h. Datenstruktur kann Zeitstempel als eindeutigen Schlüssel verwenden.
- Je nach Wahl des Konfliktauflösungsmechanismus können unterschiedliche Verfälschungen der Simulation auftreten.
- Je feiner die Zeiteinheit, desto seltener sind Konflikte. Voraussetzung dazu sind allerdings hinreichend viele zufallsverteilte Zeitintervalle.

# Erzeugen von Zufallszahlen

- Häufig benötigt man für Simulationen Zufallszahlen mit anderer Verteilung als Gleichverteilung
- Pseudozufallszahlengeneratoren liefern aber gleichverteilte Zufallszahlen.
- Lösung: Umtransformieren
  - Sei  $f(x)$  die Wahrscheinlichkeitsdichte und  $F(x)$  die daraus per Integration gewonnene kumulierte Wahrscheinlichkeitsverteilung.
  - Invertiere  $y = F(x)$  zu  $x = F^{-1}(y)$ .
  - Ziehe gleichverteilte Zufallszahlen  $y$  aus dem Intervall  $[F(x_{\min}); F(x_{\max})]$
  - Dann ist  $x$  nach der Wahrscheinlichkeitsdichte  $f$  verteilt.





# Beispiel: Poisson-Prozess

- Poisson-Prozess ist typische Annahme für Nutzer-Interaktion,
- Für diskrete Ereignissimulation werden Zeitabstände zwischen Ankunftsereignissen benötigt. Diese sind (siehe oben) exponentialverteilt:

$$f(t) = \lambda \exp(-\lambda t)$$

- Schritt 1: Integrieren

$$F(t) = \int_0^t f(t) dt = 1 - \exp(-\lambda t)$$

- Schritt 2: Invertieren

$$t = \log(1 - u) / (-\lambda)$$

mit  $u$  gleichverteilt in  $[0;1)$

- Trick: Wenn  $u$  gleichverteilt in  $[0;1)$  dann auch  $(1 - u)$
- Somit:

```
RANDOM() := (double) rand() / RAND_MAX  
EXP_TIME(lambda) := -log(RANDOM()) / lambda
```

# Normalverteilte Zufallszahlen

- Nicht immer führt diese Methode zum Erfolg.
- Beispiel: Die Normalverteilung ist nicht geschlossen integrierbar.

$$F(x) = \int_{-\infty}^x f(t) dt \quad \text{mit} \quad f(x) = \frac{1}{\sqrt{2\Pi}} \cdot e^{-\frac{x^2}{2}}$$

- Lösungen:
  - Look-up Tabelle
    - Approximiere  $F(x)$  durch abschnittweise lineare Funktionen
  - Gesetz der großen Zahl
    - Ziehe 12 gleichverteilte Zufallszahlen  $x_i$  aus  $[0;1)$
    - Dann ist  $(\sum x_i) - 6$  näherungsweise normalverteilt
  - Box-Muller-Marsaglia Algorithmus

- Die diskrete Ereignissimulation ist ein Standardwerkzeug der Informatik. Entsprechend gibt es eine Reihe von Simulationsprogrammen, im Bereich der Telematik beispielsweise:
  - Network Simulator 2 („ns-2“) aus Berkeley
    - Vorteil: Umfangreiche Bibliothek an Protokollimplementierungen verfügbar
    - Nachteil: Hoher Speicherbedarf und langsam
  - OMnet++
    - Objektorientierte diskrete Ereignissimulation mit C++
    - Etabliert sich als Alternative zu NS-2
  - GlomoSim
    - Besonders für Simulation im Bereich mobile und drahtlose Kommunikation

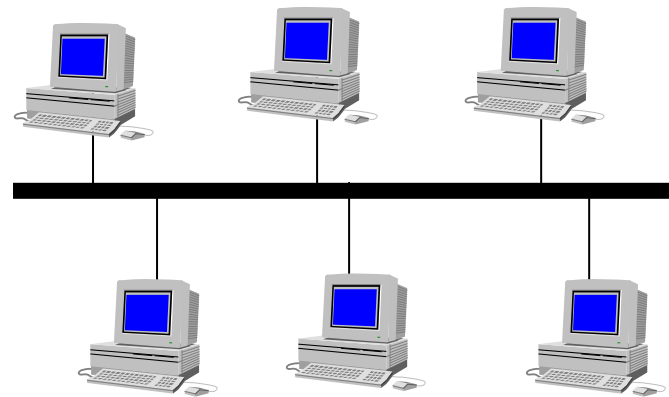
# Abstraktionsebenen (1)

---

- Ein wichtiges Problem der diskreten Ereignissimulation ist die Bestimmung der Feinheit, mit der ein Sachverhalt modelliert wird.
  - Extremes Beispiel: Simuliere Verhalten von MAC (z.B. Ethernet) und vollständigem Protokollstapel (z.B. IP, TCP, HTTP). Dann entspricht die Komplexität und Laufzeit dem realen System. Man hat nichts mehr gewonnen.
- Lösung: Einführen einer Abstraktionsebene. Feinere Aspekte werden nur noch über ihre Wahrscheinlichkeitsverteilung modelliert. Größere Aspekte werden mittels diskreter Ereignissimulation simuliert.
- Problem: Falls das gewählte Modell relevante Feinheiten nicht enthält, können die Ergebnisse wertlos sein.
  - Beispiel: Verkehrsströme als Poissonprozess modelliert, obwohl Komponenten nicht unabhängig.

## Abstraktionsebenen (2)

- Vor allem die unbekanntenen Wechselwirkungen zwischen den Komponenten können ein Problem darstellen.
- Beispiel:
  - Nutzereingaben entsprechen einem Poissonprozess.
  - Jede Anfrage erzeugt je eine Antwort aus  $N$  Komponenten (z.B. verschiedene Datenbanken).
  - Ergebnis: Jede einzelne Komponente erzeugt Poisson-Verkehr, aber die Komponenten sind nicht unabhängig.



## Nochmal: Einschwingphase

---

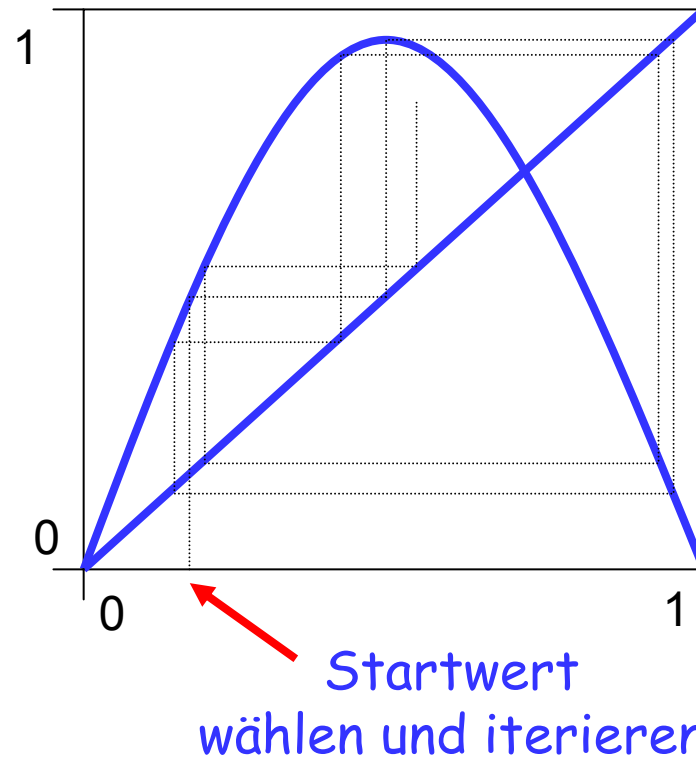
- Bei der numerischen Integration nähert sich die relative Häufigkeit allmählich dem Erwartungswert an.
  - Dies gilt insbesondere auch für die Parameter, die wiederum den Ablauf der Simulation beeinflussen, z.B. Warteschlangenlängen, etc.
- Bei der diskreten Ereignissimulation stellen sich daher erst allmählich die Verhältnisse ein, die man untersuchen möchte.
  - Da man oft die Wechselwirkungen im System nicht kennt, kann kaum bestimmen, wie lange die Simulation laufen muss, bis eine vorgegebene Fehlerschranke unterschritten wird.
  - Besonders bei nichtlinearen Systemen, können Sprünge und Stufen auftreten, die eine Vorhersage verhindern.
  - Lösung: Simulation lange laufen lassen und hinterher sehen, wann sich das System hinreichend gut dem stationären Zustand angenähert hatte.
  - Problem: Chaotische Systeme und Instabilitäten

# Deterministisches Chaos (1)

- Betrachte die Folge

$$x_{n+1} = ax_n(1-x_n)$$

- Anschaulich hat man für  $1 < a < 4$  eine in einem Quadrat eingeschriebene Parabel, zwischen der und der Diagonalen man immer hin- und herspringt.
- Man springt immer abwechselnd vertikal zur Parabel und horizontal zur Diagonalen.
- Für  $a=1$  läge die Parabel vollständig unterhalb der Diagonalen. Für  $a=4$  füllte die Parabel das Quadrat vollständig aus.



## Deterministisches Chaos (2)

- Bei diesem Beispiel macht man die folgende Beobachtung:
  - Für  $1 < a < 3$  strebt die Iterationsfolge nach einer Übergangsphase von einigen Iterationen gegen einen Fixpunkt (=Attraktor), der unabhängig vom Startwert ist. Die Folge der Iterierten verhält sich also dynamisch stabil.
  - Für  $3 < a < 3,5699456\dots$  strebt die Folge ebenfalls unabhängig vom Startwert gegen eine periodische Folge.
  - Für  $3,5699456\dots < a < 4$  erhält man chaotische Folgen, deren Verlauf jeweils vom genauen Startwert abhängt.
- Auf einem Digitalrechner sind die Folgen bei gegebenem Startwert jederzeit eindeutig reproduzierbar. Im realen System verhindern winzigste Abweichungen (z.B. Messfehler) die Wiederholbarkeit.
- Bei der Simulation solcher Systeme, die derart zum Chaos neigen, ist somit Vorsicht geboten, welche Schlüsse man aus den Simulationsergebnissen zieht.



Vergleichen Sie die Einschwingphase einer numerischen Integration mit der der diskreten Ereignissimulation.

Lösung:

- Bei der numerischen Integration sind die einzelnen „Ereignisse“ unabhängig. Es kann daher leicht ermittelt werden, wann die Ergebnisse innerhalb einer vorgegebener Fehlergrenzen korrekt sind.
- Startet man ein System aus einem Anfangszustand (vgl. Euler-Verfahren bzw. Simulation der Markov-Kette) muss man warten, bis Verfälschungen aufgrund dieses Anfangszustandes hinreichend klein geworden sind.
- Eine Möglichkeit dazu, ist das Mitteln über viele Simulationsläufe mit gleichverteilten Anfangszuständen.
- Bei komplexeren Simulationen (typisch: diskrete Ereignissimulation) ist nicht bekannt, wie Anfangszustände gleichverteilt erzeugt werden können. Hierzu wären ja die Simulationsergebnisse erforderlich.
- Man muss daher empirisch hinreichend lange die Simulation ablaufen lassen.

# Simulationen – Noch allgemeiner ...

- Simulationen sind häufig, aber nicht immer Simulationen über die Zeit.
- Gegenbeispiel:
  - Vgl. Beispiel der Integration einer Funktion mit der Monte-Carlo-Methode.
- Wie bei der Analogie zwischen Euler-Verfahren und diskreter Ereignissimulation braucht man keine mathematische Beschreibung sondern nur einen Algorithmus (=Modell), um diese Grundidee anzuwenden.
- Beispiel (als Übungsaufgabe für zu Hause):
  - Jeder Rechner eines Peer-to-Peer Systems kennt  $N$  gleichverteilt zufällige andere Rechner dieses Systems.
  - In diesem System werden Nachrichten geflutet.
  - Wie viele Schritte braucht eine Nachricht von der Quelle bis zum Ziel (Mittelwert, Standardabweichung)?
  - Ist die Verteilung überhaupt eine Normalverteilung?

- Simulationen sind „algorithmische Lösungen“ von Modellen.
- Es besteht eine enge Analogie zur numerischen Integration
  - Monte-Carlo-Methode
  - Euler-Verfahren
- Es muss keine geschlossene mathematische Formulierung vorliegen, ein Algorithmus reicht aus.
- Häufig wird das Verfahren der „diskreten Ereignissimulation“ genutzt.
- Es schreibt den Zustand eines Modells schrittweise in der Zeit fort.
- Dafür gibt es eine große Zahl von fertigen Simulationsumgebungen in die man das zu untersuchende Modell einbinden kann.
- Da man meist keinen gültigen Anfangszustand kennt, muss dabei die „Einschwingphase“ abgewartet werden.
- Manchmal entstehen dabei Oszillationen (vgl. deterministisches Chaos).

---

# Questions?



Thomas Fuhrmann

Department of Informatics  
Self-Organizing Systems Group  
c/o I8 Network Architectures and Services  
Technical University Munich, Germany

[fuhrmann@net.in.tum.de](mailto:fuhrmann@net.in.tum.de)