1 Zahlenrätsel

1.1 Einstiegs-Aufgabe

Im folgenden Zahlenrätsel sollen die Buchstaben so durch Ziffern ersetzt werden
(gleiche Buchstaben – gleichen Ziffern und verschiedene Buchstaben – verschiedene Ziffern),
dass die Rechnung korrekt wird.

Bemerkung:
Eine Ziffer ist Element der Menge ZF = {0,1,2,3,4,5,6,7,8,9}.

```
  S E N D
+ M O R E
---------
M O N E Y
```

Welche Möglichkeiten gibt es diese Aufgabe zu lösen ?

1.1.1 Lösungsmöglichkeiten

1.1.1.1 "geschicktes" Probieren.

Dies ist für diejenigen geeignet, die gerne Denksportaufgaben lösen.

1.1.1.2 Gausscher Algorithmus

In der Aufgabe kommen die 8 Unbekannten S, E, N, D, M, O, R, Y vor.
Mit diesen kann man ein lineares Gleichungssystem aufstellen, das sich nach dem Gausschen Algorithmus lösen lässt.
Dies ist für diejenigen geeignet, die gerne Mathematikaufgaben lösen.

1.1.1.3 Brute Force - Methode (systematisches Probieren aller Möglichkeiten)

Bei den 8 verschiedenen Buchstaben S, E, N, D, M, O, R, Y gibt es
$N = 10^8 = 100\,000\,000$ Möglichkeiten.
Da ein Mensch nicht so alt wird, alle Möglichkeiten auszutesten, lässt man diese einen Rechenknecht (Computer) mittels eines Programms durch rohe Rechengewalt (brute force) machen.
Dies ist für diejenigen geeignet, die gerne programmieren.

Wir konzentrieren uns hier im folgenden auf die Brute Force - Methode

1.1.2 Zeitliche Abschätzung der Rechenzeit

Man berechnet die **Mindestzeit**, die der schnellste heute sich auf dem Markt befindliche Computer benötigt. Wenn diese schon entsprechend groß ist (z.B. 100 Jahre), kann man die Aufgabe **praktisch** nicht mit der Brute Force - Methode lösen und man muss diesen Lösungsansatz verwerfen.

1.1.2.1 Optimale Annahmen

1) CPU-Takt: 10 GHz
D.h. die CPU benötigt für einen Takt die Zeit:
$T = 1/10^{10}$ Sekunden $= 10^{-10}$ s

2) Die CPU arbeitet jeweils in **einem** Prozessortakt eine Möglichkeit ab.
Dies kann auf keinen Fall unterboten werden. Damit ist die Rechenzeit t_M, die für eine Möglichkeit benötigt wird:
$t_M = 10^{-10}$ s

Damit lässt sich die Mindestrechenzeit berechnen:

1.1.2.2 Berechnung der Mindestrechenzeit

t_{min} = Anzahl aller Möglichkeiten · Rechenzeit pro Möglichkeit
$= N \cdot t_M = 10^8 \cdot 10^{-10}$ s = 0,01 s.

Das Ergebnis bestätigt uns, diesen Lösungsansatz weiter zu verfolgen !

1.1.2.3 Parallelisierbarkeit

Ist das Problem parallelisierbar, d.h. wie kann man die Lösung dieser Aufgabe auf mehrere Prozessoren verteilen, so dass diese parallel und gleichzeitig arbeiten ?

Lösung:
Man kann z.B. das Programm abändern und aus dem einen Programm 10 verschiedene Programme machen, in der Art, dass beim Programm0 S = 0 gesetzt wird, beim Programm1 S = 1, ... und beim Programm9 S = 9 gesetzt wird. Dann kann jedes dieser einzelnen Programme parallel und gleichzeitig auf 10 verschiedenen Computern ablaufen.

1.1.2.4 Maximale Anzahl von Möglichkeiten

Wie groß wird die maximale Anzahl der Möglichkeiten, bei Aufgaben dieser Form (Buchstaben werden durch Ziffern ersetzt) ?

Lösung:
Da maximal 10 verschiedene Ziffern in dieser Art von Aufgaben vorkommen, können auch nur maximal 10 verschiedene Unbekannte (Variablen) vorkommen. D.h. die maximale Anzahl der Möglichkeiten ist:
$N = 10^{10}$ Möglichkeiten.
und
$t_{min} = N \cdot t_M = 10^{10} \cdot 10^{-10}$ s = 1 s.

1.2 Weitere Aufgaben

1.2.1 Aufgabe Zahlenrätsel

Im folgenden Zahlenrätsel sollen die Buchstaben so durch Ziffern ersetzt werden
(gleiche Buchstaben – gleichen Ziffern und verschiedene Buchstaben – verschiedene Ziffern),
dass die Rechnung korrekt wird.

```
  R I E S E
+ G A U S S
 -----------
E U K L I D
```

1.2.2 Aufgabe magisches Quadrat

Wie muss man die Zahlen 1 bis 9 auf die 9 Zellen einer Tabelle mit 3 Zeilen und 3 Spalten verteilen, dass jeweils die Summe in den Spalten, Zeilen und Diagonalen gleich groß ist ?

1.2.3 Aufgabe Travelling Salesman

Ein Geschäftsmann will eine Rundreise durch alle N = 4 Städte in einem Land machen, so daß er wieder am Ausgangspunkt zurückkehrt und jede Stadt (außer dem Ausgangspunkt) genau einmal besucht hat.
Im Extremfall ist jede Stadt mit allen anderen Städten durch eine Straße verbunden.
Welche Route muß der Geschäftsmann wählen, damit er insgesamt ein Minimum an eines bestimmten Parameters (Entfernung, Kosten, Zeit) verbraucht hat ?

Projekt:
1) Verändern Sie die Anzahl der Städte und ermitteln Sie die jeweilige Rechenzeit für das Programm.

2) Ab einer bestimmten Anzahl von Städten wird die Anzahl der Rechenzeit zu groß.
Deshalb kann man diese auf mehrere Rechner in einem Netzwerk verteilen

2 Wer gewinnt im Schach ?

Literatur: 1) C'T 26/98 2) Komplexitätstheorie W.J. Paul Teubner Studienbücher

2.1 Aufgabe

Gibt es im Schach für "weiß" eine Gewinnstrategie ?

Um dieses Problem einen Computer berechnen zu lassen, müsste er alle möglichen Spiele analysieren ?
Wie viele Spiele gibt es ?

2.1.1 Lösung

Eine grobe Schätzung:
Ein Spiel geht durchschnittlich 40 Züge und bei jedem Spiel hat man durchschnittlich 20 Möglichkeiten eine Figur zu bewegen.
damit ergeben sich die N Möglichkeiten:
$N = 20^{40} = (2 \cdot 10)^{40} = 2^{40} \cdot 10^{40} = 2^{10 \cdot 4} \cdot 10^{40} = (2^{10})^4 \cdot 10^{40} > (10^3)^4 \cdot 10^{40} >= 10^{52}$

Um die Rechenzeit von solchen zeitintensiven Programmen zu verringern, gibt es theoretisch zwei Möglichkeiten:
1) Schnellere Prozessoren
2) Ein Programm auf mehrere Prozessoren verteilen.

2.2 Möglichkeiten der Verringerung von Rechenzeit

2.2.1 Schnellere Prozessoren

2.2.1.1 Idee

Man nimmt ein Programm und lässt es auf immer schnelleren Prozessoren ablaufen. Dadurch wird dann die gesamte Rechenzeit dieses Programms immer kleiner und nähert sich 0.
Leider geht dies aus folgendem Grund nicht:

2.2.1.2 Annahmen über die Elemente eines Prozessors

1) Ein Schaltelement besitzt mindestens 2 Zustände
2) Bei einem Schaltvorgang wird Energie umverteilt
3) Das Schaltelement hat endliche Größe.

2.2.1.3 Satz

Aus diesen Annahmen folgt aus einigen Resultaten der theoretischen Physik (Stichwort: Heisenberg'sche Unschärferelation, Schwarzschild-Radius, Gravitationskollaps), dass jeder Schaltvorgang mindestens die Zeit t_{min} braucht:

$$\boxed{t_{min} \approx 5{,}6 \cdot 10^{-33}\ \text{s}}$$

Das heißt der schnellste Prozessor hat eine Frequenz f_{max} von:
Frequenz
$f_{max} = 1/(5{,}6 \cdot 10^{-33}\ \text{s}) = 1{,}8 \cdot 10^{32}\ \text{Hz} = 1{,}8 \cdot 10^{23}\ \text{GHz}$

2.2.2 Multiprozessorsysteme

2.2.2.1 Idee

Man nimmt ein Programm und lässt es auf einem Computer ablaufen, in dem man immer mehr Prozessoren in den Computer einbaut (**n-Prozessorsystem**) und den Programmablauf auf alle Prozessoren des Computers gleichmäßig aufteilt. Alle Prozessoren arbeiten also parallel und gleichzeitig ihren Teil des Programms ab.
Dadurch wird dann die gesamte Rechenzeit dieses Programms immer kleiner und nähert sich 0.
(Probleme, die das gestatten, nennt man **massiv parallelisierbar**).
Leider geht dies im allgemeinen nicht immer, weil sich in fast jedem Programm Teile befinden, die sich nicht parallelisieren lassen und in dieser Phase des Programmablaufs der Computer die Leistung eines Einprozessorrechners hat (die anderen Prozessoren sind unbeschäftigt).
Dies wird im folgenden genauer behandelt.

2.2.2.2 Ein einführendes Beispiel

2.2.2.2.1 Zubereitung eines Kartoffelsalats

Eine Person braucht für das Zubereiten von Kartoffelsalat (Kartoffeln waschen, schälen und schneiden) 60 Minuten. Das Kochen der Kartoffeln dauert 30 Minuten.
Insgesamt benötigt man also 1,5 Stunden bis der Kartoffelsalat zubereitet ist

2.2.2.2.2 Frage

Wie lange dauert es im Idealfall, bis der Kartoffelsalat zubereitet ist, wenn statt einer Person 5 Personen den Kartoffelsalat zubereiten ?
Ist dies der fünfte Teil von der Zeit, die benötigt wird, wenn nur eine Person den Kartoffelsalat zubereitet ?

2.2.2.2.3 Lösung

60 Minuten : 5 Personen + 30 Minuten = 42 Minuten.

2.2.2.3 Amdahls Gesetz

Man gibt ein zu untersuchendes Programm **P** vor, zur Abarbeitung dieses Programms eine bestimmte Rechenzeiteinheit benötigt. Diese Zeit nennt man hier 1 PZE (Programmzeiteinheit). Das kann z.B. eine Stunde sein.
Den Anteil nicht parallelisierbarer Programmteile an der Gesamtrechenzeit (die also nur von einem Prozessor abgearbeitet werden können), bezeichnet man mit α
Dabei bedeutet z.B. $\alpha = 0{,}2$ dass 20 % der Anweisungen des Programms nicht parallelisierbar ist.
Vom Rest des Programms mit der Rechenzeit $1- \alpha$ nimmt man an, dass es vollständig parallel abgearbeitet werden kann.

Die Rechenzeit für die Abarbeitung des Programms P bei einem n-Prozessorsystem mit dem Anteil α ist:

$$\boxed{tR_n = \alpha + \frac{1-\alpha}{n}}$$

2.2.2.3.1 Definition Leistung

Die Leistung ist definiert, als die Anzahl der abgearbeiteten Programme P in einer Programmzeit PZ

$$\boxed{L = \frac{Anzahl\ \mathrm{Pr}ogramme\ P}{PZ}}$$

Beispiel:
Ein Computer mit einem Prozessor arbeitet 2 Programme P in einer Programmzeit PZ ab. Seine Leistung beträgt ist also 2 P/PZ. Die Einheit P/PZ wird oft weggelassen.

2.2.2.3.2 Zusammenhang zwischen t_R und L

Beispiel:
1 Programm P wird in 10 PZE (Programmzeiteinheiten) abgearbeitet.
Also wird in 1 PZE 1/10 Programm P abgearbeitet.
Also gilt für die Leistung allgemein:

$$\boxed{L = \frac{1}{t_R}}$$

2.2.2.3.3 Leistung L eines n-Prozessorsystems mit

Die Leistung L bei einem n-Prozessorsystem mit dem Anteil α ist:

$$\boxed{L_n = \frac{1}{\alpha + \frac{1-\alpha}{n}}}$$

2.2.2.3.4 Beispiele

1) $\alpha = 0{,}2$
$L_1 = 1$

2) $\alpha = 0{,}2$

n	L
2	1,67
3	2,14
4	2,5

Der Einbau eines zweiten Prozessors bringt also nur 67% mehr Leistung !

3) $\alpha = 0{,}2$
Wie viel mehr Leistung erbringt der Einbau eines 10-ten Prozessors, verglichen mit der Leistung eines einzelnen Prozessors ?

$$L_9 = \frac{1}{0{,}2 + \frac{1-0{,}2}{9}} \approx 3{,}46 \qquad L_{10} = \frac{1}{0{,}2 + \frac{1-0{,}2}{10}} \approx 3{,}57$$

$$also: \quad zu = \frac{L_{10} - L_9}{L_1} \approx \frac{0{,}11}{1} = 0{,}11$$

2.2.2.3.5 Verhalten der Leistung für große n

$$\lim_{n \to 00} L_n = \lim_{n \to 00} \frac{1}{\alpha + \frac{1-\alpha}{n}} = \frac{1}{\alpha}$$

Bemerkung:
Probleme, bei denen α möglichst nahe bei 0 liegen heißen massiv parallelisierbar.

2.2.2.4 Erweitertes Amdahls Gesetz

Die Verteilung der Abarbeitung von Programmteilen auf verschiedene Prozessoren benötigt einen Verwaltungsaufwand (Verwaltungsoverhead) bzw. Kommunikationsaufwand , der auch Rechenzeit kostet.
Die Verteilung von 20 Aufgaben braucht doppelt so viel Zeit, wie die Verteilung von 10 Aufgaben. Das erweiterte Gesetz von Amdahl geht von einem linearen Kommunikationsaufwand aus.
Den Anteil des Kommunikationsaufwands an der Gesamtrechenzeit , bezeichnet man mit β.

$$\boxed{L_n = \frac{1}{\alpha + n \cdot \beta + \frac{1-\alpha}{n}}}$$

Beim Einsatz von immer mehr Prozessoren wächst die relative Leistung zunächst, bis der Aufwand für die Kommunikation so stark anwächst, dass jede weitere CPU die Leistung senkt.
Man hätte also α und β möglichst gerne beim Wert 0 und versuch t linear anwachsenden Kommunikationsaufwand zu vermeiden.
Für spezielle Aufgaben wie z.B. das Knacken von Verschlüsselungen durch rohe Rechengewalt (brute force), kann man α und β so nahe bei Null halten (z.B. α = 0,001 β = 0,00001), dass Amdahls Gesetz kaum ins Gewicht fällt.

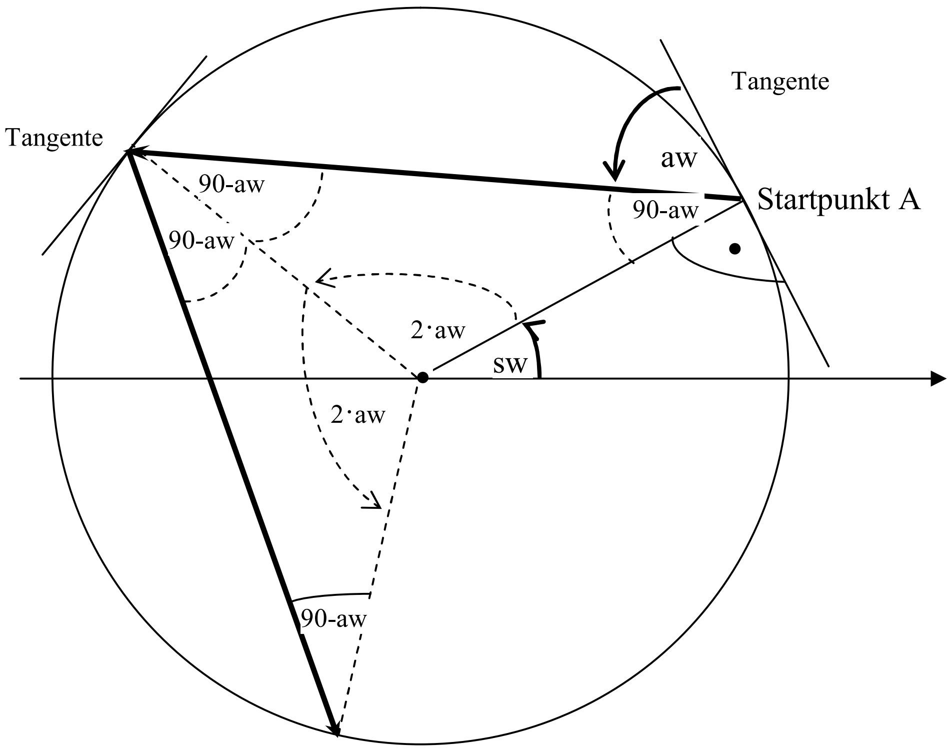
Aufgabe:
gegeben:
a) α = 0,2 β = 0,01
b) α = 0,001 β = 0,00001)
Bei wie viel Prozessoren hat das Multiprozessorsystem die maximale Leitung ?
Schreiben Sie ein Programm, das das Maximum berechnet.

3 Der Weg des Lichts in einem verspiegelten Kreis

3.1 Aufgabe

Schreiben Sie ein Programm, das den Verlauf des Lichtstrahls (siehe Zeichnung unten) grafisch darstellt.

Ein Kreis mit dem Radius R ist innen verspiegelt. Sein Mittelpunkt M liegt im Ursprung O(0|0) eines rechtwinkligen Koordinatensystems.
Auf der Kreislinie wird durch die Wahl des Startwinkels sw der Startpunkt A bestimmt.
Von diesem Startpunkt A aus wird ein Lichtstrahl unter dem Abschusswinkel aw (bzgl. der Tangente) in den verspiegelten Kreis "abgeschossen".



Die Folge der Punkte die der Lichtstrahl im Kreis berührt, lassen sich durch seine Koordinaten bzw. den Winkel w (von der positiven x-Achse aus gemnessen) darstellen.
Für die Folge der Punkte Pn(xn|yn) die der Lichtstrahl auf der Kreislinie berührt gilt:

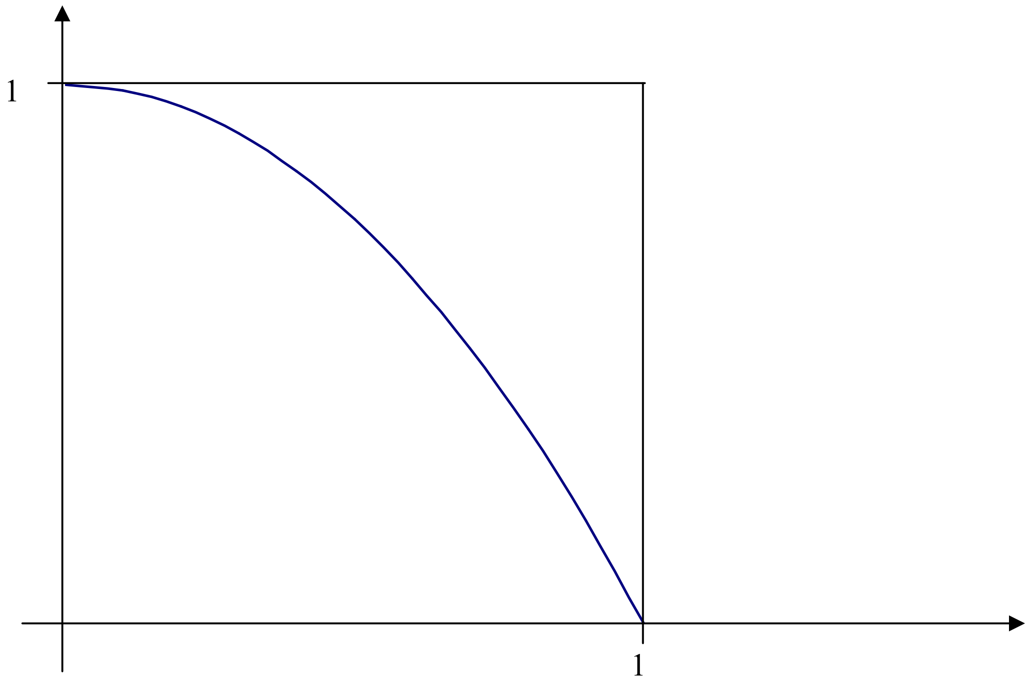
$$w_n = sw + 2n \cdot aw \qquad n=0,1,2,3,...$$
$$x_n = R \cdot \cos(w_n)$$
$$y_n = R \cdot \sin(w_n)$$

Bemerkung:
Als weitere Aufgabe kann statt des Kreises auch eine andere geometrische Figur (Rechteck, Ellipse, usw.) benutzt werden.

4 Flächenberechnung mit Dartspfeilen

4.1 Aufgabe

Berechnen Sie den Flächeninhalt des Schaubilds der Funktion $f(x) = 1 - x^2$, der positiven y-Achse und der positiven x-Achse.



4.1.1 Lösungsidee

Mit verbundenen Augen wirft man "blindwütig" sehr oft (N-mal) mit einem Pfeil auf eine Scheibe, auf der die obige Zeichnung eingezeichnet ist. Dabei trifft man n-mal die Fläche zwischen der Parabel, der y-Achse und der x-Achse.

.

Wenn man die Fläche unter der Parabel mit A_P und die Fläche des Quadrats mit A_Q bezeichnet, dann gilt für sehr grosse N:

$$\frac{A_P}{A_Q} \approx \frac{n}{N}$$

Da hier $A_Q = 1$, gilt:

$$A_P \approx \frac{n}{N}$$

4.1.2 Weitere Lösungsidee

Man nähert die Fläche der Parabel durch die Summe der Flächen der in die Parabel einbeschriebenen Rechtecke an.

Konkret:
Man unterteilt das Intervall der Länge 1 auf der x-Achse in gleich lange Intervall der Länge b. Am Ende des jeweiligen Intervalls betrachtet man den Funktionswert an der jeweiligen Intervallstrecke und kann dann den Flächeninhalt des entsprechenden Rechtecks berechnen.

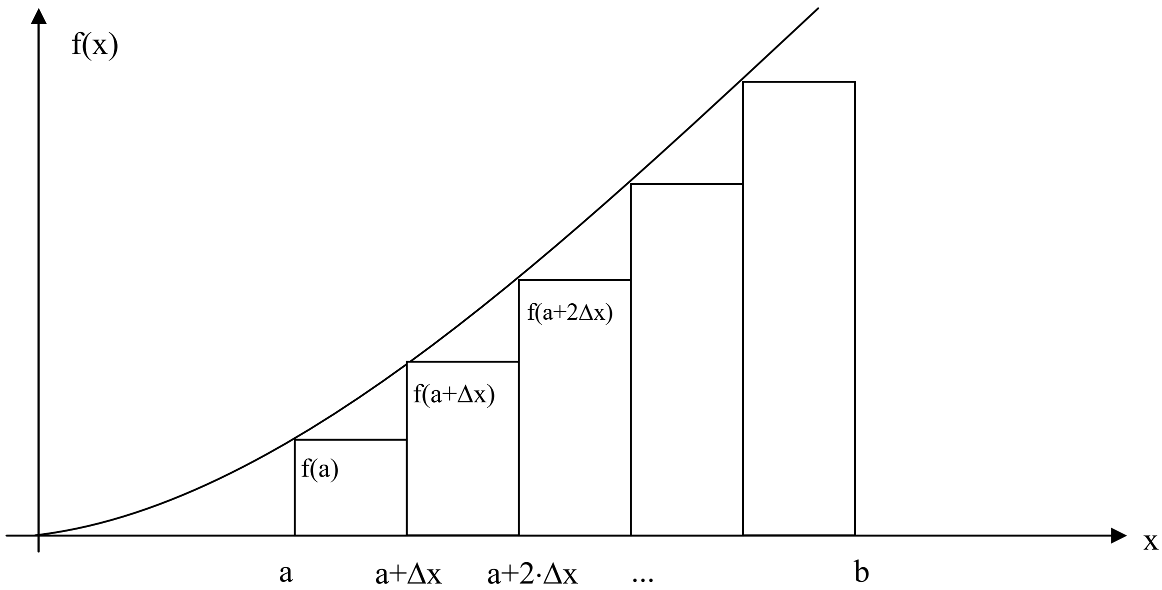
4.1.2.1 Beispiel

Wenn man das Intervall [a ; b] z.B. in n = 5 gleich lange Intervalle mit der Breite

$$\Delta x = \frac{b-a}{n}$$

unterteilt, dann ist die Summe der Flächen der Rechtecke:
$A = f(a)\cdot\Delta x + f(a+\Delta x)\cdot\Delta x + f(a+2\Delta x)\cdot\Delta x + f(a+3\Delta x)\cdot\Delta x + f(a+4\cdot\Delta x)\cdot\Delta x$

allgemein gilt (für beliebiges n):
$A = f(a)\cdot\Delta x + f(a+\Delta x)\cdot\Delta x + f(a+2\Delta x)\cdot\Delta x + f(a+3\Delta x)\cdot\Delta x + \ldots + f(a+(n-1\cdot\Delta x)\cdot\Delta x$



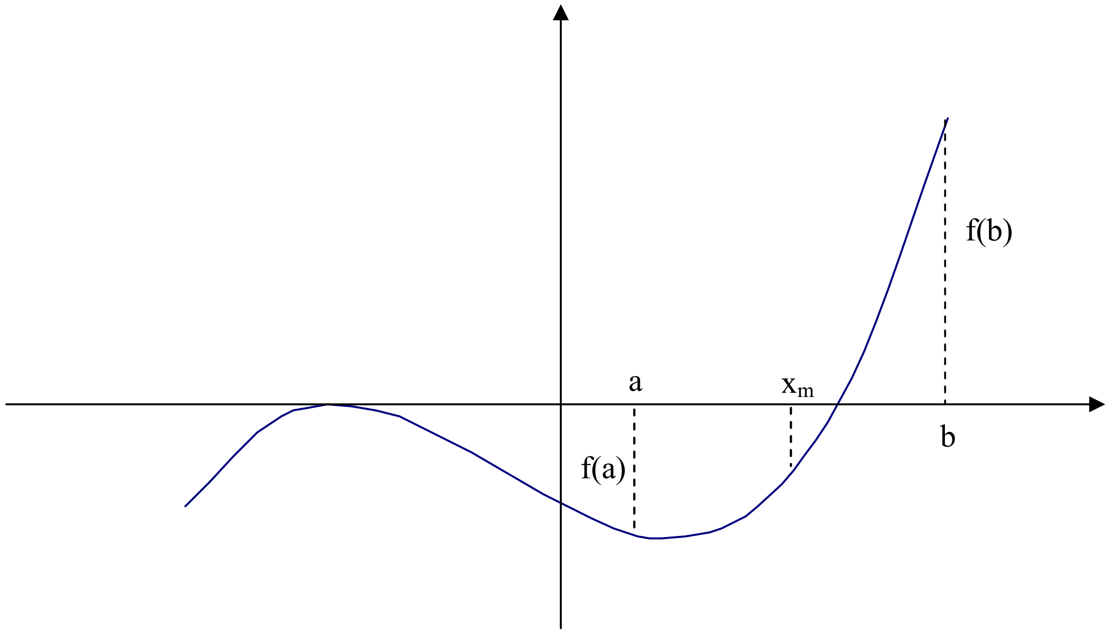
4.1.2.2 Weitere Idee

Berechnen Sie die Länge des obigen Parabelsegments, das durch die x-Achse und y-Achse begrenzt wird.

5 Kurvendiskussionen

5.1 Aufgabe

Berechnen Sie im Intervall [-1, 2] die Nullstelle der Funktion:
`f(x) = x³+ x²-2x-2`



5.1.1 Lösungsidee (zu einem Näherungsverfahren)

Setze $x_l = a$, $x_r = b$ (x_l steht für xlinks, x_r für xrechts).
Zuerst bestimmt man die Mitte x_m des Intervalls $[x_l, x_r]$.
Ist $f(x_m) < 0$, dann setzt man $x_l = x_m$ und bildet die Mitte x_m des Intervalls $[x_l, x_r]$.
Ist $f(x_m) < 0$, dann setzt man $x_r = x_m$ und bildet die Mitte x_m des Intervalls $[x_l, x_r]$
Ist $f(x_m) = 0$, dann ist man fertig, ansonsten wiederholt diese Intervallhalbierung hinreichend oft (z.B. 1000 mal).

5.2 Aufgabe

Berechnen Sie im Intervall [-1, 2] das Minimum (Maximum) der Funktion:
`f(x) = x³+ x²-2x-2`

5.2.1 Lösungsidee (zu einem Näherungsverfahren)

Unterteilen Sie das Intervall [a, b] in z.B. 1000 Unterintervalle und berechnen Sie dort jeweils am Ende des Unterintervalls den Funktionswert. Bestimmen Sie dann das Minimum (Maximum) dieser Funktionswerte.

5.3 Aufgabe

Berechnen Sie im Intervall [a, b] das Minimum (Maximum) eines Polynoms 3. Grades:
`f(x) = ax³ + bx² + cx + d`

5.4 Aufgabe

Erstellen Sie ein C-Programm, das feststellt, ob ein Polynom 4. Grades achsensymmetrisch zur y-Achse, punktsymmetrisch zum Ursprung bzw. weder achsensymmetrisch zur y-Achse noch punktsymmetrisch zum Ursprung ist.

5.4.1 Lösungsidee (zu einem Näherungsverfahren)

Unterteilen Sie ein Intervall [-a, a] in z.B. 1000 Unterintervalle und berechnen Sie dort jeweils am Ende des Unterintervalls den Funktionswert f(x) und vergleichen ihn mit f(-x).

6 Kryptolgie oder wie man eine Datei verschlüsselt

6.1 Aufgabe

1) Schreiben Sie ein Programm, das die Datei "geheim.txt" anlegt und darin den Text "Dies ist eine geheime Mitteilung" abspeichert.

2) Schreiben Sie das Programm "kodierung", das die Datei verschlüsselt.

Eine Verschlüsselungsmethode:
Jedes Byte wird durch den in der ASCII-Tabelle nachfolgenden ASCII-Wert kodiert.
Der ASCII-Wert 65 (entspricht dem ASCII-Zeichen A) wird z.B. kodiert durch den ASCII-Wert 66

Beispiel:
Die Datei mit dem Inhalt (Bytes) mit den zugehörigen ASCII-Werten

65	71	255	97	105

wird kodiert zu der folgenden Datei:

66	77	0	98	106

3) Schreiben Sie das Programm "dekodierung", das die oben verschlüsselte Datei wieder dekodiert.

4) Überlegen Sie sich andere Verschlüsselungsverfahren und implementieren Sie diese.
Beispiel:
Die Reihenfolge der Bytes in einer Datei wird umgekehrt. Das letzte Byte wird zum ersten.

7 Was ist die bessere Strategie ?

7.1 Aufgabe

Im Spiel A gewinnt man, sobald das **erste** Mal "Kopf" beim Werfen einer Münze erscheint. Maximal zweimal darf die Münze geworfen werden.
Im Spiel B gewinnt man, sobald das **erste** Mal die "Sechs" beim Werfen eines Würfels erscheint. Maximal sechsmal darf gewürfelt werden.
In welchem Spiel ist die Gewinnwahrscheinlichkeit höher ?

7.1.1 Mathematische Lösung

Wenn A verliert, hat er 6 mal nicht die "Sechs" gewürfelt. Dies insgesamt 6 mal jeweils (unabhängig voneinander) mit der Wahrscheinlichkeit 5/6.

Also ist die Wahrscheinlichkeit für A zu verlieren: $p_{VA} = (\frac{5}{6})^6 \approx 0{,}335$

und zu gewinnen: $\boxed{p_{GA} = 1-(\frac{5}{6})^6 \approx 0{,}665}$

Entsprechend ist die Wahrscheinlichkeit für B zu verlieren: $p_{VB} = (\frac{1}{2})^2 = 0{,}25$

und zu gewinnen: $\boxed{p_{GB} = 1-(\frac{1}{2})^2 = 0{,}75}$

7.1.2 EDV-Lösung (Simulation)

Man macht mit Hilfe eines Zufallgenerators eine große Anzahl N_W von "Würfelspielen" und zählt die Anzahl n_W der Erfolge und bildet dann das Ergebnis $H_W = n_W / N_W$
Dann macht man mit Hilfe eines Zufallgenerators genau die gleich große Anzahl N_W von "Münzspielen" und zählt die Anzahl n_M der Erfolge und bildet dann das Ergebnis $H_M = n_M / N_M$

7.1.2.1 Zufallszahlen und die Programmiersprache C

Beispiel für die Verwendung des Zufallsgenerators: (siehe Hilfe-Menü in VC++)

```
#include "stdafx.h"
#include <stdio.h>
#include <time.h>
#include <stdlib.h>

void main(void)
{
int erg;
srand( (unsigned)time( NULL ) );
erg = rand();
printf("Zufallszahl=%d\n",erg);
}
```

8 Wie viele Zufallszahlen gibt es ?

Will man in einem Programm einen Zufall simulieren, benutzt man dazu eine Funktion, die einen Zufall erzeugt. In C ist dies die Funktion rand.
Ab wann wiederholen sich die Zufallszahlen, d.h. wie groß ist die Periode ?

9 Wo selbst ein Mathe-Professor irrte

9.1 Aufgabe

Vor einigen Jahren gab es das folgende Quiz in den USA.
Genau hinter einer von 3 geschlossenen Türen steht ein Auto, das der Kandidat gewinnen kann. Der Kandidat muss einen Tipp abgeben, wo sich das Auto befindet.
Dann öffnet der Moderator eine Türe hinter der sich das Auto **nicht** befindet. Der Kandidat steht jetzt also noch vor zwei geschlossenen Türen.
Dann hat der Kandidat folgende zwei Möglichkeiten (Strategien) weiterzuspielen:
Strategie S1:
Er bleibt bei seinem vorher abgegeben Tipp.
Strategie S2:
Er ändert seinen Tipp.

Gibt es eine Strategie, die besser ist, oder ist es egal, für welche man sich entscheidet.

Bemerkung:
Laut einem SPIEGEL Artikel hat sogar ein Mathematik-Professor in einem Lesebrief eine falsche Lösung dazu angegeben.

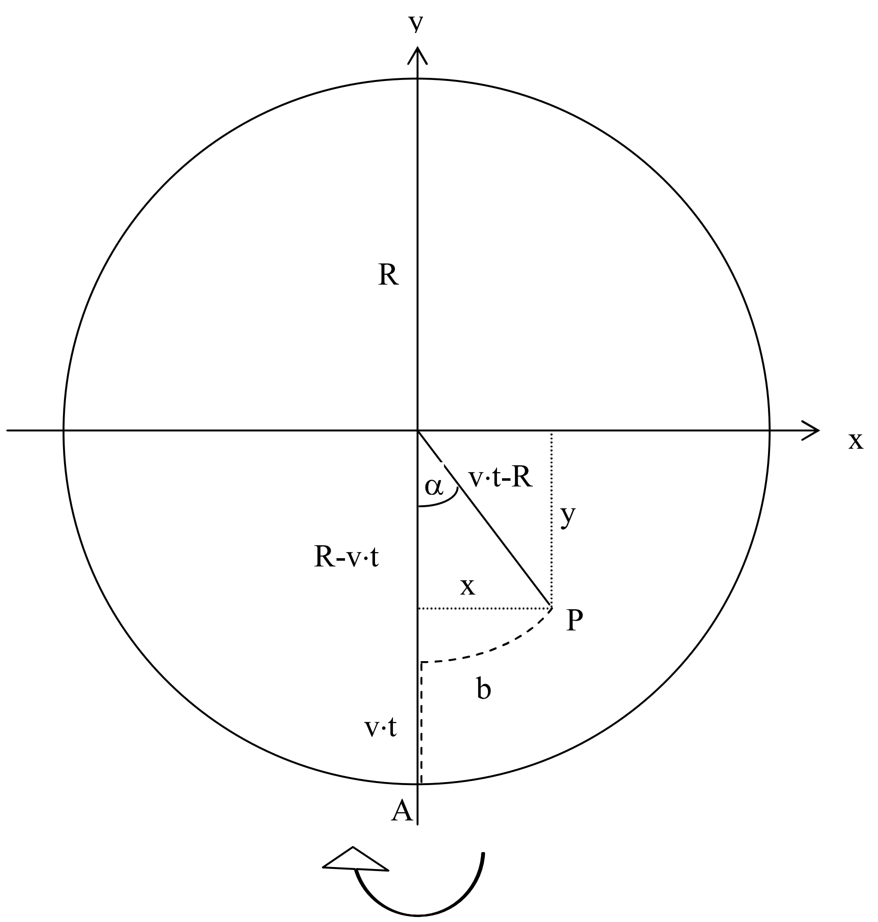
10 Coriolis Kraft

10.1 Motivation

Die Erde dreht sich von oben über dem Nordpol aus gesehen entgegen des Uhrzeigersinns. Auf der nördlichen Halbkugel wehe ein Wind von Süd nach Nord. Aus den äquatornäheren Gegenden bringt er eine größere Umlaufgeschwindigkeit mit und eilt dem Längengrad im Norden voraus, er erscheint nach Osten, d.h. in der Bewegungsrichtung gesehen nach rechts abgelenkt.
Auf der nördlichen Halbkugel erfahren also bewegte Körper eine Kraft nach rechts.
Diese nennt man die sogenannte Corioliskraft.
Sie wird als Grund für die stärkere Erosion der rechten Flussufer und das schnellere Abnutzen der rechten Eisenbahnschienen auf der nördlichen Halbkugel angegeben.

10.2 Physik

Eine mit weißem Paper und Kohlepapier belegte, waagrechte Scheibe (z.B. Schallplatte) mit Radius R dreht sich mit der Winkelgeschwindigkeit ω von oben gesehen (entspricht der südlichen Halbkugel) mit dem Uhrzeigersinn.
Aus einer geneigten Rinne lässt man eine schwere Kugel längs des Durchmessers in Richtung y-Achse über die Scheibe rollen.
Welche Spur hinterlässt die Kugel auf der Scheibe ?



In dem Moment. wo die Kugel die Kreisscheibe bei A betritt und sich in Richtung y-Achse bewegt, trifft sie nach einer bestimmten Zeit mit einem Punkt P der Kreisscheibe zusammen, der sich innerhalb dieser Zeit auf sie zubewegt.

10.3 Mathematik

Es gilt:
$\alpha = \omega \cdot t$

und damit:

$$y = -(R - vt) \cdot \cos(\omega \cdot t)$$
$$x = (R - vt) \cdot \sin(\omega \cdot t)$$

10.4 Informatik

Erstellen Sie ein Programm, das diese Linie zeichnet.

11 Simulationen

11.1 Chemische Simulationen

11.1.1 Einführung (Konservative und dissipative Strukturen)

Konservative Strukturen
Konservative Kräfte sind statischen Kraftwirkungen zwischen materiellen Teilen, die dem jeweiligen Ganzen ihre mehr oder weniger symmetrische Gestalt aufprägen. Konservative Kräfte ergeben konservative Strukturen.
Beispiele:
die exakt definierte Verteilung der Moleküle in einem Atom,
die räumliche Struktur eines Proteins,
die symmetrische Anordnung der Bausteine im Kristallgitter,
die bizarre Form eines Gebirgsmassivs
das sichtbare Muster eines Sternensystems am Nachthimmel.
der molekulare Aufbau des gesamten Informationspeichers (die Nukleinsäure-Doppelspirale)

Dissipative Strukturen
Ein in der Zeit periodischer Vorgang kann unter geeigneten Bedingungen sehr leicht auch ein periodisches Raummuster erzeugen. Im oszillierenden Reaktionssystem können bei ständigem Nachschub der Reaktionspartner und Abdiffundieren der Reaktionsprodukte räumlich stabile Muster entsehen. Man bezeichnet sie als dissipative Strukturen, weil zu ihrer Aufrechterhaltung ständig Energie dissipiert (zerstreut) werden muß.
Fragen wir hier nach der Ursache räumlicher und zeitlicher Ordnung, so muß die Antwort lauten: Sie resultiert allein aus den dem System inhärenten Reaktionseigenschaften sowie der Art der Randbedingungen und wird durch den stationären Zustrom und die Dissipation von Energie (das heißt durch den Zufluß energiereicher und den Abfluß energiearmer Materie) aufrechterhalten.
Man hat hier ein zweites fundamentales Ordnungsprinzip der Natur vor uns. Für die Gestaltbildung in der belebten Natur ist die dissipative Struktur von ebenso großer Bedeutung wie die allein auf statische Kraftwirkungen gründende konservative Ordnung.
Beispiele:
1) In einer chemischen Reaktion, der sogenannten Zhaboutinsky-Reaktion, werden komplexe Muster ausgebildet.
2) Gott der Amöben
Einzellige Schleimpilze werden (durch eine disspipative Strukturausbildung) veranlasst zu einem vielzelligen Plasmodium zusammenzutreten, das sich wie ein Organismus verhält.

Es gibt zwei fundamentale Prinzipien der Morphogenese (Ausformung eines Lebewesens bzw. seiner Organe während seiner Entwicklung von der Eizelle bis zum geschlechtsreifen Individuum) : ein konservatives und ein dissipatives.
Nach dem erstgenannten Prinzip gehen Struktur und Gestalt aus einer Überlagerung von anziehenden und abstoßenden konservativen Kräften hervor, wobei die in permanenter Wechselwirkung stehenden Untereinheiten des Gesamtsystems stabile räumliche Lagen einnehmen bzw. sich auf stabilen Bahnen (zum Beispiel Planetenbahnen) um einen Schwerpunkt bewegen. Dieses Ordnungsgefüge wird ohne Dissipation von Energie aufrechterhalten.
Zum Unterschied hiervon sind dissipative Strukturen dynamische Ordnungszustände, die nur durch einen Metabolismus (Stoffwechsel), eine ständige Energiedissipation, unterhalten werden können. Sie resultieren in Form räumlicher Muster — ähnlich wie stehende Wellen - aus der Überlagerung von Materietransport und synchronisierter, periodischer Umwandlung

und sind als solche nicht in additiver Weise aus Unterstrukturen zusammensetzbar. In der Morphogenese sorgen sie für eine räumliche Organisation und Determinierung der konservativen Strukturelemente, deren Aufbau durch das genetische Programm der Zelle festgelegt ist. Als Erregungsmuster im Netzwerk der Nervenzellen überlagern sie übersummenhaft verschiedene Teilinformationen und stellen so das materielle Korrelat von »Gestalt« dar. Die zur Ausbildung der dissipativen Strukturen notwendigen Wechselbeziehungen beruhen auf konservativen Kraftwirkungen, wie auch die permanente räumliche Fixierung dissipativer Muster der stabilisierenden konservativen Kraft bedarf. Gemeinsam ist dem konservativen und dissipativen Gestaltbildungskonzept die Kooperativität der statischen bzw. dynamischen Wechselwirkungen.

11.1.2 Modelle

Es gibt verschiedene Modelle um die "Realität" nachzubilden (simulieren):

11.1.2.1 Spieltheoretisches Modell (chemische Reaktionsspiele)

Zwei Reaktionspartner (Moleküle bzw. Atome) müssen einander begegnen, denn sonst kann keine Reaktion stattfinden. Alle chemischen Reaktionskräfte haben aber nur sehr kurze Reichweiten, deswegen kann diese Begegnung nicht dadurch veranlasst werden.
Die Moleküle bzw. Atome sind in ständiger ungeordneter Bewegung (Brownsche Molekularbewegung) und irren regellos umher (random walk).
Dadurch überstreicht jeder Reaktionspartner im Laufe der Zeit ein bestimmtes Areal, dessen radiale Ausdehnung mit der Quadratwurzel - wie Einstein zum ersten Mal zeigte - aus der Zeit anwächst.
Nicht jeder Zusammenstoß zweier Moleküle bzw. Atome führt zu einer Reaktion. Der Zusammenstoß muß so heftig sein, damit die Moleküle bzw. Atome hinreichend aktiviert werden. (Aktivierungsenergie).

Eine chemische Reaktion kann man durch ein Spiel simulieren:

Wirklichkeit	Modell
Reaktionsgefäß	Urne
Molekül	Kugel
Molekülsorte	Kugelfarbe
Konzentration eines Stoffes	Zahl der Kugeln einer Farbe
Brownsche Bewegung	Durchmischung
Aktivierung durch Stoß	lotterieartige Ziehung
Elementarreaktion	Kugelaustausch
Reaktionsdauer	Zahl der Ziehungen
Aktivierungsenergie	Wahrscheinlichkeitswürfel zur Entscheidung, ob eine gezogene Kugel hinreichend aktiviert ist.

11.1.2.2 analoges Modell

Mit Hilfe von Differentialgleichungen kann man die Abhängigkeit der Konzentrationen der verschiedenen Stoffe angeben.

11.1.3 Konkrete Modelle

11.1.3.1 Ein einfaches chemisches Gleichgewicht

11.1.3.1.1 Analoges Modell

Ein Stoff A wird mit einer Geschwindigkeit v_1, die proportional zur Konzentration [A] des Stoffes A ist, in den Stoff B umgewandelt.
Der Stoff B wird mit der Geschwindigkeit v_2, die proportional zur Konzentration [B] des Stoffes B ist, in den Stoff A umgewandelt.

$$A \underset{k_2}{\overset{k_1}{\longleftrightarrow}} B$$

$v_1 = k_1\ [A]$
$v_2 = k_2\ [B]$
k1, k2 sind Geschwindigkeitskonstanten.
[A], [B] in: g/cm^3
v_1, v_2 in : $g/cm^3/s$

11.1.3.1.2 Spieltheoretisches Modell (Kugelspiel)

Erwischt man aus der lotterieartigen Ziehung aus einer Urne eine Kugel A, dann wird diese mit einer bestimmten Wahrscheinlichkeit (z.B. wenn mit einem Würfel 6 gewürfelt wird) gegen eine Kugel B ausgetauscht.
Erwischt man eine Kugel B, dann wird diese mit einer bestimmten Wahrscheinlichkeit (z.B. wenn mit einer Münze "Kopf" gewürfelt wird), gegen eine Kugel A ausgetauscht.
Die durchschnittliche Anzahl der Umwandlungen einer Kugel A in eine Kugel B (und umgekehrt) hängt von der Konzentration der Kugel A und von der Wahrscheinlichkeit k_1 ab.

Nach einer bestimmten Anzahl Ziehungen wird das Spiel abgebrochen. Jede Ziehung - ob erfolgreich oder nicht - zählt als eine Zeiteinheit.

Konkrete Werte:
k1 = 1
k2 = 0,5
Anzahl der Kugeln A zum Zeitpunkt 0: A(0) = 150
Anzahl der Kugeln B zum Zeitpunkt 0: B(0) = 0

11.1.3.2 Lotka-Volterra-Modell

11.1.3.2.1 Analoges Modell

$$A+X \xrightarrow{k_1} 2X$$
$$X+Y \xrightarrow{k_2} 2Y$$
$$Y \xrightarrow{k_3} E$$

Dabei sind k_1,k_2,k_3 die Geschwindigkeitskonstanten der entsprechenden Prozesse. Die Konzentration von A wird konstant gehalten.

Aus dem Modell folgen die entsprechenden kinetischen Gesetze:

$$\dot{X} = k_1 \cdot A \cdot X - k_2 \cdot X \cdot Y$$

$$\dot{Y} = k_2 \cdot X \cdot Y - k_3 \cdot Y$$

Für den stationären Zustand $S(x_s|y_s)$ gilt:

$$x_s = \frac{k_3}{k_2} \qquad y_s = \frac{k_1}{k_2} \cdot A$$

Bei kleinen Abweichungen (von X und Y) vom stationären Zustand gilt für die Schwingungsdauer T (um den stationären Zustand):

$$T = \frac{2\pi}{\sqrt{k_1 \cdot k_3 \cdot A}}$$

11.1.3.3 Das Brüsselator-Modell:

11.1.3.3.1 Analoges Modell

$$A \xrightarrow{k_1} X$$
$$2X+Y \xrightarrow{k_2} 3X$$
$$B+X \xrightarrow{k_3} Y+D$$
$$X \xrightarrow{k_4} E$$

Die Konzentrationen von A,B,D,E werden wieder konstant gehalten.

Aus dem Modell folgen die entsprechenden kinetischen Gesetze:
$dX/dt = k1 \cdot A + k2 \cdot X \cdot X \cdot Y - k3 \cdot B \cdot X - k4 \cdot X$
$dY/dt = k3 \cdot B \cdot X - k2 \cdot X \cdot X \cdot Y$

12 Warum ist das Wetter prinzipiell nicht vorhersagbar ?

12.1 Philosophie und Erkenntnistheorie

Der Mensch kann sich von einem Objekt der "Realität" (z.B. elektrischer Widerstand) ein Abbild machen.

	Objekt	Modell
Beispiel 1:	Widerstand	$R = U \cdot I$
Beispiel 2:	Massenbeschleunigung	$F = m \cdot a$

In den Naturwissenschaften sagt man zu diesem Abbild **Modell**. Dieses Modell wird mit formal-operationalen, mathematischen Methoden beschrieben. Um herauszufinden, wie gut dieses Modell das Objekt abbildet, werden an dem Objekt praktische und am Modell theoretische (mathematische Berechnungen) Experimente gemacht.
Das Modell ist desto besser, je mehr sich diese entsprechen. Entsprechen sich die gemachten Experimente nur wenig, dann muss das Modell verändert werden.
Die (theoretischen) Experimente am Modell nennt man auch **Simulationen** (des Objekts).
Simulationen sind besonders wichtig bei Situationen, in denen praktische Experimente am Objekt nicht gemacht werden können oder sollten (z.B. Menschenversuche, Flugzeugabsturz, Versuche mit dem Wetter).
Simulationen von Zufallsexperimenten (das Objekt Zufall steckt auf der untersten Ebene in der Materie drin) durch Zufallsziffern nennt man **Monte-Carlo-Methoden**.

12.2 Das Masern-Problem

In einem Kinderheim sind die Masern ausgebrochen. Jeden Tag wird sich die Anzahl der kranken Kinder erhöhen, weil nicht zu vermeiden ist, dass kranke und gesunde Kinder miteinander Kontakt haben.
Das folgende mathematische Modell soll versuchen, diese Krankheit zu simulieren, um das Verhalten und die Gesetzmäßigkeiten eines solchen Systems zu verstehen:

Die Anzahl der kranken Kinder am Tag i+1 hängt einerseits von der Anzahl der bereits kranken Kinder am Vortag i ab. Je mehr kranke Kinder es am Tag i hat, desto mehr kranke Kinder hat es einen Tag später.
Genauer: a_{i+1} ist proportional a_i
Kurz: $a_{i+1} \sim a_i$
Die Anzahl der kranken Kinder am Tag i+1 hängt auch von der Anzahl der noch gesunden Kinder am Vortag i ab, denn es ist kein Zuwachs mehr möglich, wenn alle Kinder krank im Bett liegen.
Wenn 30% krank sind, sind noch 100%- 30% gesund.
Allgemein sind 100% - p = 1 - p gesund.
Genauer: $a_{i+1} \sim (1 - a_i)$
Da sich die Kinder sicher nicht alle treffen und auch nicht jeder Kontakt zu einer Ansteckung führt, taucht in der Formel für z auch noch ein Ansteckungsfaktor k auf.
Berücksichtigt man alle diese Überlegungen für eine Gesamtformel, so gilt:
$a_{i+1} = a_i \cdot (1-a_i) \cdot k$

Für jeweils einen festen Ansteckungsfaktor k kann man ausgehend von einem Startwert a_0 den Verlauf der Krankheit berechnen.

12.2.1 Beispiel

a_0 = 0,25
k = 2

n	a_n
0	0,25
1	0,375
2	0,46875
3	0,49999237
...	...

12.2.2 Aufgaben

1) Berechnen Sie für den festen Startwert $a_0 = 0{,}3$ für die folgenden k-Werte
0,5; 1; 1,5; 3; 2,5; 3
jeweils in einer Tabelle den Krankheitsverlauf mit einem Taschenrechner.

2) Machen Sie das gleiche mit einem Tabellenkalkulationsprogramm.

3) Schreiben Sie ein Programm:
Wählen Sie einen festen Startwert a_0 im Bereich [0, 1], d.h. zwischen 0 und 1.
Auf der x-Achse werden die Werte von k im Bereich [0, 4] aufgetragen.
Auf der y-Achse werden die Werte von a_n im "eingeschwungenen" Zustand, z.B. im Bereich zwischen a_{100} und a_{200} aufgetragen.

12.2.3 Ergebnis

Wenn k zwischen 0 und 3 liegt, streben die a_n für große n gegen **einen** festen Wert.
Wenn k zwischen 3 und $1+\sqrt{6} \approx 3{,}5$ liegt, "pendeln" die a_n für große n zwischen 2 Werten
Man sagt a_n hat die Periode 2.
Wenn k größer wird, wird auch die Periode größer. Ab dem $k \approx 3{,}5699$ springen die Werte a_n wild durcheinander. Vorhersagen sind nicht mehr erkennbar.
Es findet ein Übergang von der Ordnung ins Chaos statt.

Wegen der starken Abhängigkeit der Anzahl a_n von k - kleinste Änderungen von k können völlig andere Ausgangswerte und Szenarien ergeben- sagt man, dass ein **deterministisches Chaos** vorliegt.
Das mathematische Modell, das das Wetter beschreibt, ist auch ein deterministisches Chaos.
Weil alle von Wetterstationen gemessene Systemparameter nur einen bestimmten Messfehler haben, ist es möglich, dass der Wert des Parameters gerade einen kritischen Wert hat und deswegen an dieser Stelle nicht mehr vorhersagbar ist !

13 Warteschlangen

13.1 Einführung

Einen Großteil unserer Zeit verbringen wir in der Warteschlange: an der Kasse im Supermarkt, im Wartezimmer beim Arzt oder im Verkehrsstau auf dem Weg zur Arbeit. Jeder von uns kennt das Unbehagen, in einer langen Warteschlange zu stehen. Oftmals lässt sich nicht einmal abschätzen, wie lange man noch ausharren muss und ob es sich überhaupt lohnt abzuwarten.
Abgesehen von der psychischen Beeinträchtigung, die von einer langen Warteschlange ausgeht, sind Warteschlangen auch aus volks- bzw. betriebswirtschaftlicher Sicht nicht wünschenswert. Denn die Zeit, die man in der Warteschlange zubringt, ist Untätigkeitszeit, die weder den Kunden noch dem Betreiber des Systems zugute kommt. Den wirtschaftlichen Einfluss von Warteschlangen kann man am besten am Beispiel der Produktion verdeutlichen. Lange Durchlaufzeiten durch die Produktion haben zur Folge, dass man neue Produkte nicht schnell genug auf den Markt bringen kann und der Konkurrenz das Feld überlassen muss. Da lange Durchlaufzeiten mit hohen Beständen korreliert sind, entstehen durch die auf Bearbeitung wartenden Halbfertigfabrikate außerdem hohe Kapitalbindungskosten, die sich negativ auf das Betriebsergebnis auswirken.
Das Phänomen des Wartens wird seit fast einem Jahrhundert wissenschaftlich erforscht. Bereits 1917 publizierte der dänische Ingenieur und Mathematiker A.K. Erlang, der bei einer Kopenhagener Telefongesellschaft beschäftigt war, eine mathematische Formel, mit deren Hilfe man Fernsprechvermittlungsstellen dimensionieren kann. Nach Erlang waren es hauptsächlich Nachrichtentechniker, die mathematische Verfahren benutzten, um den Telefonverkehr durchgängiger und effizienter zu machen. Die Fachleute erzählen sich, dass gerade diejenigen Länder, die Mitte des vergangenen Jahrhunderts über die schlechtesten Telefonsysteme verfügten, zumindest die besten Mathematiker auf dem Gebiet der Warteschlangentheorie hervorgebracht hätten. Mit dem Aufkommen der Datenverarbeitung werden diese Methoden auch zur Konzeption von Rechensystemen verwendet. Ziel der Analysen ist es, bereits im Vorfeld der Planung Engpässe und Schwachstellen zu erkennen. Ganz langsam fängt man an, die Warteschlangentheorie auch auf Fragen der Produktion, des Verkehrs und der Modellierung von Geschäftsprozessen auszudehnen. Inzwischen sind mehr als 10 000 wissenschaftliche Publikationen über Warteschlangenprobleme erschienen, die sich auf die unterschiedlichsten Bereiche unseres täglichen Lebens erstrecken. Angesichts einer so großen Zahl gesicherter Erkenntnisse, lässt sich kaum erklären, warum wir heute noch so oft in der Warteschlange stehen...

13.2 Anwendungsbereiche der Warteschlangentheorie

Warteschlangen lassen sich in vielen Bereichen unseres Alltagslebens beobachten. Die nachfolgende Tabelle stellt einige dieser Bereiche vor:

Kunde	**Bedienstation**	**Warteschlange**
Zwischenprodukte	Maschine	Puffer vor der Maschine
Skifahrer	Ski-Lift	Ski-Fahrer vor dem Ski-Lift
Flugzeuge	Startbahn	Flugzeuge auf dem Rollfeld
Flugzeuge	Landebahn	Flugzeuge in der Warteschleife
Touristen	Airline-Schalter	Touristen vor dem Schalter
Reisende	Fahrkartenschalter/-automat	Reisende vor dem Schalter
Kunden	Supermarkt-Kasse	Kundenschlange
Fahrzeuge	Tankstelle oder Ampel	Fahrzeugschlange
Patienten	Arzt	Wartezimmer
Bestellungen	Verkaufsabteilung	Lieferverpflichtungen
Druckaufträge	Drucker	Druck-Jobs
Studenten	Mensa	Studenten vor der Essensausgabe
Maschinen	Instandhaltung	Reparaturaufträge
Transportgüter	Transportsystem	Zwischenläger
Lastwagen	Laderampe	Kolonne
Anrufe	Vermittlungsstelle	Anrufwiederholer
Prozesse	CPU	Prozess-Warteschlange

13.3 Definitionen und Grundlagen

13.3.1 Jobs

Es sind nicht bloß Menschen, denen im Rahmen der verschiedenen betrieblichen Abläufe das Schicksal des Wartens zuteil wird. Man spricht allgemein von **Jobs** und das können nun sein:
1) Kunden eines Unternehmens, die auf die Erledigung eines Auftrages warten.
2) Produkte, die im Zuge ihrer Produktion auf die Durchführung eines weiteren Verarbeitungsschrittes warten.
3) Nachrichten, Anfragen, Reparatur- und Instandhaltungsaufgaben etc.

Ein Job ist die Zeit T_q in der Queue bevor seine Bedienung (Service) beginnt und die Zeit S dauert. Die Gesamtzeit T eines Jobs im System ist dann $T = T_q + S$

13.3.2 Warteschlangensystem

Man kann schematisch das folgende Bild eines **Warteschlangensystems** zeichnen:

```
   Ankünfte                                 Abfertigungen

  ---> ....        Servicestation              .. --->
```

man erkennt, dass ein Warteschlangensystem die folgenden wesentlichen Bestandteile umfasst:

1) einen **Ankunftsstrom** von Jobs
2) eine **Servicestation**, in der diese Jobs auf irgendeine Art und Weise weiterverarbeitet werden
3) gegebenenfalls eine Warteschlange aus Jobs, die auf ihre Weiterverarbeitung warten müssen
4) Jobs, die das System verlassen, nachdem sie ihr Service erhalten haben.

13.3.3 Zufall

Der Zufall setzt in diesem einfachen Modell an zwei Punkten an:

1) Im Ankunftsstrom: im allgemeinen sind die Zeitpunkte, zu denen neue Jobs eintreffen **nicht exakt vorhersagbar**.
2) In der Servicestation: die Dauer eines Services wird im allgemeinen keine Konstante sein, sondern ebenfalls **zufälligen Charakter** haben.

13.3.4 Zahl der parallelen Servicekanäle

Die Leistungsfähigkeit einer Servicestation hängt entscheidend davon ab, wie viele **parallele** Servicekanäle in dieser Station vorhanden sind. Damit stellt sich sofort ein wichtiges Designproblem ein: soll man für jeden Kanal eine eigene Queue vorsehen? Schematisch würde das so aussehen:

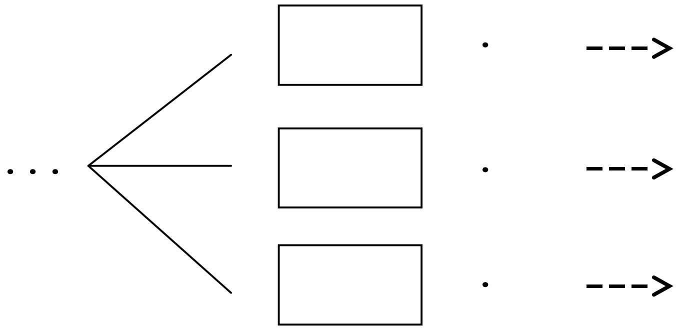
```
...  [      ]  .   --->

...  [      ]  .   --->

...  [      ]  .   --->
```

Ein Beispiel dafür wären die Kassen in einem Supermarkt.

Oder sollen alle Kanäle eine gemeinsame Queue bedienen? Schematisch:



Dies ist z.B. in einem Frisiersalon der Fall, in welchem die Kunden warten, bis irgendeiner der Friseure frei wird.
Aus theoretischen, hier nicht bewiesenen Überlegungen folgt, dass das System mit **einer** Queue jenem mit mehreren Queues überlegen ist!
Im folgenden wird immer von Warteschlangensystemen mit **genau einer Queue** ausgegangen.

13.3.5 Notationen

$E(W_q)$: Erwartungswert (=Mittelwert) der Wartezeit (Zeit vom Eintreffen bis einschließlich der Bedienung eines Kunden) = durchschnittliche Wartezeit
$E(L_q)$: Erwartungswert (=Mittelwert) der Länge der Warteschlange (ausschließlich der Anforderungen, die gerade bedient werden) = durchschnittliche Länge der Warteschlange
$E(W)$: Erwartungswert (=Mittelwert) der Wartezeit (Zeit vom Eintreffen bis einschließlich der Bedienung eines Kunden) = durchschnittliche Wartezeit
$E(L)$: Erwartungswert (=Mittelwert) der Länge der Warteschlange (einschließlich der Anforderungen, die gerade bedient werden) = durchschnittliche Länge der Warteschlange
λ : Ankunftsrate = Durchschnittliche Anzahl der Ankünfte pro Zeiteinheit.
μ : Bedienungsrate = Durchschnittliche Zahl der von einem Bedienungskanal abgefertigten Jobs pro Zeiteinheit.
$1/\lambda$: Mittlere Zeitdauer zwischen zwei aufeinanderfolgenden Ankünften.
$1/\mu$: Mittlere Zeitdauer der Bedienung.
c : Anzahl der Bedienungselemente
r : λ/μ
ρ : $= \lambda/(n\cdot\mu)$ Verkehrsintensität
T_q: Zeit eines Jobs in der Queue bevor seine Bedienung (Service) beginnt.
S: Dauer der Bedienung (Sevice).
T: Die Gesamtzeit T eines Jobs im System ist dann $T = T_q + S$
G/G/c: Ein Warteschlangensystem, das aus c Bedienungselementen besteht und bei dem die Ankunftszeiten und die Bedienungszeiten irgendeine (G: General) Verteilung haben.

13.4 Mathematik

Seit 1917 hat man unter Einsatz überaus **komplizierter** und **schwieriger** mathematischer Methoden eine ganze Reihe von überraschend einfachen Formeln gefunden, die einem substantielle Aussagen über komplexe Systeme erlauben wie z.B. Little's Formula. Damit kann man bereits erste quantitative Analysen erstellen und einfache Leistungsparameter von Warteschlangensystemen berechnen.

Das wichtigste Belastungsmaß ist die **Verkehrsintensität** ρ, die definiert ist als

$$p = \frac{\lambda}{c\mu}$$

Wenn nun ρ>1 oder gleichbedeutend damit λ>cμ, dann wird die Länge der Queue im Mittel unbeschränkt zunehmen, es strömen ja mehr Jobs in das System, als abgefertigt werden können. Das System erweist sich als instabil, man sagt auch, es existiert kein Steady State. Erstaunlicherweise gilt das auch dann, wenn ρ>1, also im Mittel pro Zeiteinheit genauso viele Jobs ankommen, als erledigt werden können, es sei denn, Service-und Zwischenankunftszeiten sind konstant und die Ankünfte perfekt auf die Services zeitlich abgestimmt!

13.4.1 Stabilität eines G/G/c Systems

Ein G/G/c System ist nur dann stabil (= ein Steady State existiert), wenn die Verkehrsintensität ρ>1 ist, also:

$$\frac{\lambda}{c\mu} < 1$$

1961 hat J.D.C. Little die folgenden erstaunlichen Formeln gefunden, die eine Fundamentalbeziehung zwischen den Mittelwerten L, L_q und W, W_q herstellen.
Also gilt:

13.4.2 Die Formeln von Little

Im Steady State eines **G/G/c** Systems gilt immer:

$$\boxed{\begin{array}{l} L = \lambda W \text{ bzw.} \\ L_q = \lambda W_q \end{array}}$$

13.5 Konkrete Bespiele

13.5.1 Tankstelle

13.5.1.1 Voraussetzungen

An einer Tankstelle kommt durchschnittlich alle 2 Minuten ein Autofahrer an und reiht sich in einer Warteschlange ein.
Der Kunde, der "dran" ist, wird von der Zapfsäule bedient, die zuerst frei wird (dies ist das heute übliche Warteverfahren in Postämtern oder beim Friseur).
Im Durchschnitt dauert ein Tankvorgang 4 Minuten.

13.5.1.1.1 Frage

Wie groß ist die mittlere Länge der Warteschlange E(L) ?

Bemerkung:
Es gibt mehrere Möglichkeiten die mittlere Länge der Warteschlange zu "berechnen":
a) Bei jedem Simulationsschritt wird die aktuelle Länge der Warteschlange (Anzahl der wartenden Kunden) aufsummiert (über alle Simulationsschritte) und beim letzten Simulationsschritt durch die Anzahl der Simulationsschritte dividiert.

b) Bei jedem Programmlauf wird Länge der Warteschlange nach dem letzten Simulationsschritt in einer Datei auf der Festplatte abgespeichert, so dass man z.B. nach 1000-maligem Ablauf (Aufruf) dieses Programms auch 1000 Werte in dieser Datei hat. Zusätzlich speichert man auch noch bei jedem Programmlauf den Mittelwert aller dieser bisher sich in dieser Datei befindlichen Werte zusätzlich auf der Festplatte.
Dann schreibt man ein Skript (Batch-Programm), das dieses Programm z.B. 10000 mal startet und kann sich danach den sich in dieser Datei befindlichen Mittelwert anschauen.

13.5.1.1.2 Frage

Wie groß ist die mittlere Wartezeit E(W) ?

Bemerkung:
Die mittlere Wartezeit kann man wie folgt "berechnen":
Bei jedem Simulationsschritt wird die Zeit, die die Kunden in der Warteschlange in diesem (einen) Simulationsschritt warten müssen, aufsummiert (über alle Simulationsschritte) und beim letzten Simulationsschritt durch die Anzahl aller Kunden dividiert, die jemals an der Tankstelle angekommen sind.

13.5.1.1.3 Frage

Wie groß ist die Wahrscheinlichkeit, dass ein Autofahrer länger als 10 Minuten warten muß ?

13.5.1.1.4 Frage

Was ändert sich, wenn an der Tankstelle noch eine (mehrere) Zapfsäulen installiert werden ?
Wie groß wird E(L) und E(W), bei $\lambda = 0{,}5$ $\mu{=}0{,}25$ und $n = 2$?

13.5.1.2 Mathematische Bemerkungen

1)
Ein Würfel wird n = 120 mal geworfen. Mit der Wahrscheinlichkeit p = 1/6 erscheint die Zahl 6. Wie viele Male wird im Durchschnitt die Zahl 6 gewürfelt ?
Ergebnis:
Im Durchschnitt wird $M = 120 \cdot 1/6 = 20$ mal die 1 geworfen.

Allgemein:

$$M = n \cdot p$$

2)
Im Durchschnitt treffen $\lambda = 1/2$ Autofahrer pro Zeiteinheit (hier: Minute) ein.
Im Durchschnitt werden $\mu = 1/4$ Autofahrer pro Zeiteinheit bedient.
Die Tankstelle wird insgesamt die vorgegebene Zeitsdauer T beobachtet (ausgewertet).

Im Durchschnitt kommen also insgesamt die folgende Anzahl Autofahrer pro Zeiteinheit an:

$$M = \lambda \cdot T$$

Man kann nun die Ankunft eines Autofahrers so simulieren, dass man am Anfang einer Zeiteinheit mit einer gewissen Wahrscheinlichkeit p annimmt, dass ein Autofahrer ankommt (bzw. mit 1-p annimmt, dass kein Autofahrer ankommt). Dies wird nun insgesamt n mal gemacht. Insgesamt kommen also im Durchschnitt $n \cdot p$ Autofahrer während dieser n Zeitintervalle an.

Es muss gelten:

$$n \cdot p = \lambda \cdot T$$

oder nach p aufgelöst:

$$p = \frac{\lambda \cdot T}{n}$$