

# Lebenslauf einer Pfandflasche

## Die geometrische Verteilung

von Benno Grabinger

In der Unterrichtsreihe "Lebenslauf einer Pfandflasche" wird für das Pfandflaschenproblem ein Modell erstellt. Zunächst werden Materialien präsentiert, welche die Überlegenheit der Pfandflasche im Vergleich zur Einwegflasche hinsichtlich Energieverbrauch zur Herstellung und Vorteile bei der Abfallbeseitigung aufzeigen. Diese Überlegenheit besteht aber nur dann, wenn die Wiederverwendung der Pfandflasche eine bestimmte Mindestanzahl aufweist. Ansonsten machen Reinigungs- und Rückführungskosten der Flaschen die zuvor angedeuteten Vorteile zunichte.



Die Pfandflaschenproblematik führt auf Fragen wie:

Wie groß ist die Rückgabewahrscheinlichkeit für eine Pfandflasche?

Wie groß ist die Wahrscheinlichkeit, daß eine Pfandflasche mindestens  $n$  mal benutzt wird?

Wie oft wird eine Pfandflasche durchschnittlich wiederverwendet?

Welcher Zusammenhang besteht zwischen der Rückgabewahrscheinlichkeit für Pfandflaschen und der durchschnittlichen Anzahl der Benutzungen der Flasche?

Nach Entwurf eines einfachen Modells werden Antworten auf diese Fragen zunächst mit Hilfe von Simulationen gesucht. Diese Simulationen werden mit Hilfe von DERIVE durchgeführt. Dabei wird ausgehend von einfachen Methoden schrittweise zu immer komplexeren Verfahren übergegangen. Vorlagen in Form von Arbeitsblättern zeigen, wie sich diese Verfahren für den Unterricht aufbereiten lassen.

Die in den Simulationen gewonnenen Erkenntnisse führen in einem sich anschließenden Theorieteil zur geometrischen Verteilung. Es wird dort gezeigt, wie die Verwendung der Algebrafähigkeiten von DERIVE lästige und langwierige Rechnungen überflüssig macht.

Eine abschließende Betrachtung ermöglicht einen Vergleich zwischen Ergebnissen der Theorie und konkreten Angaben aus der Getränkeindustrie.

Ein umfangreicher Anhang bietet dem Lehrer ergänzende Materialien über Zufallszahlen mit DERIVE. Darin werden Werkzeuge zur Herstellung von Zufallszahlen vorgestellt.

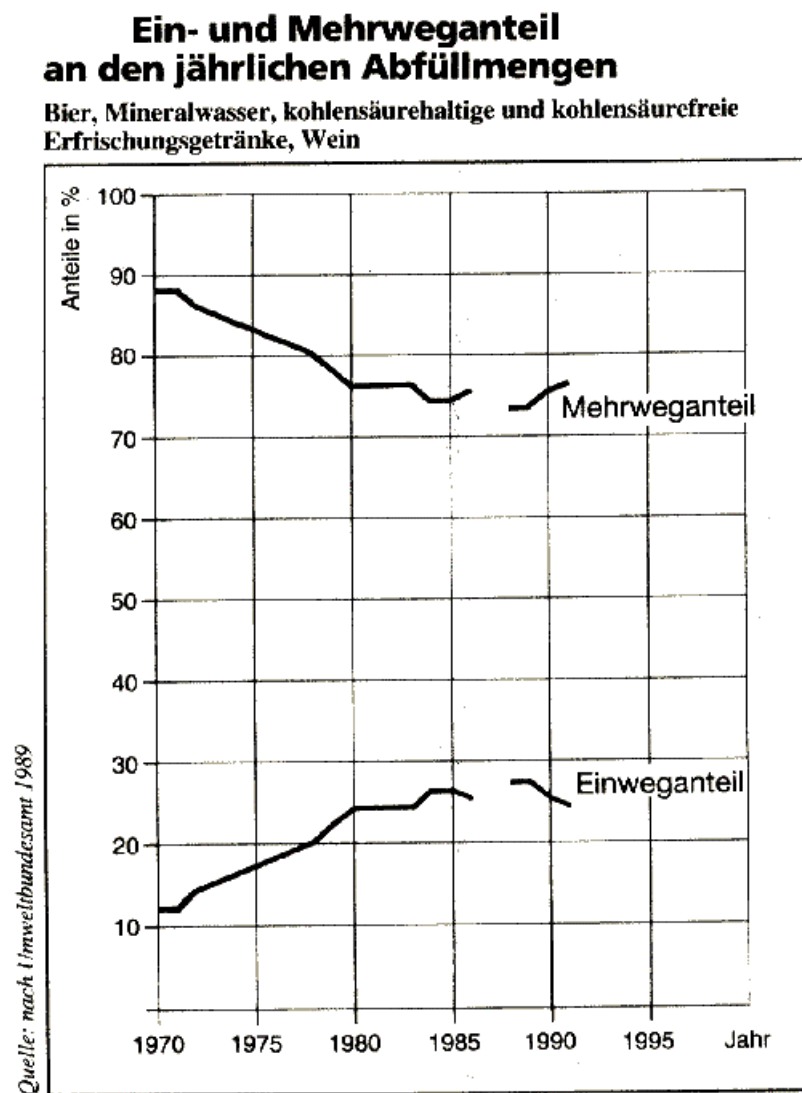
Außerdem wird gezeigt, mit welchem Verfahren in DERIVE Zufallszahlen erzeugt werden. Die Güte der erzeugten Zufallszahlen wird durch graphische Methoden und mit Hilfe des  $\chi^2$ -Tests untersucht. Diese Informationen sind speziell für den Lehrer gedacht, können aber auch als Themen für den Unterricht fruchtbar gemacht werden.

## Einweg- und Mehrwegflaschen

Der Bundesbürger trinkt im Jahr durchschnittlich 150 Liter Bier. (Stand 1994) Neben dem Konsum offenen Biers werden diese Mengen in Einwegglasflaschen, Mehrwegglasflaschen, Weißblechdosen oder Aluminiumdosen nach Hause getragen.

Bis Mitte der achtziger Jahre erlebte die bequeme Einwegflasche eine ständige Zunahme. Entgegen anfänglicher Befürchtungen der Einzelhändler, waren die Verbraucher jedoch bereit, von den Einwegflaschen und Dosen vermehrt auf Mehrwegflaschen umzusteigen. Somit ergab sich eine Trendumkehr. Der Trend ging weg vom "Ex und Hopp" und hin zur Mehrwegflasche.

Man betrachte dazu das folgende Diagramm:



Die Mehrwegflasche ist ihren Konkurrenten überlegen in Energieverbrauch bei der Herstellung, Luft- und Umweltbelastung, Abfallaufkommen.

Der Spareffekt bei Mehrwegflaschen hängt davon ab, wie oft eine Flasche wiederbenutzt wird. Diese Anzahl wird als die Umlaufdauer der Flasche bezeichnet.

Um die 150 Liter Bier des Bundesbürgers zu verpacken, sind

- 500 Büchsen zu 0,33 Liter oder
- 300 Büchsen zu 0,5 Liter oder
- 300 Einwegflaschen zu 0,5 Liter erforderlich.

Bei der Benutzung von Mehrwegflaschen zu 0,5 Liter kommt man

- bei einer durchschnittlichen Zahl von 50 Umläufen pro Flasche mit nur

6 Mehrwegflaschen aus.

Zu berücksichtigen sind bei der Verwendung von Mehrwegflaschen die Kosten für Rücknahme, Transport und Reinigung.

Genauere Analysen ergaben, daß erst ab 25 Umläufen die Mehrwegflasche bezüglich Umweltbelastung und entstehenden Kosten der Einwegflasche überlegen ist.

Man betrachte dazu auch die folgende Abbildung. (Aus Bundruf, April 1990)



Über die Umlaufdauer in der Praxis geben die beiden folgenden Antworten von Bierbrauereien Auskunft:

Über die Umlaufdauer in der Praxis geben die beiden folgenden Antworten von Bierbrauereien Auskunft:



Herrn  
Benno Grabinger  
Am Häuselberg 1

67434 Neustadt/Hambach

Bellheim, 13.01.94 we/kr

Ihr Schreiben vom 08. 01. 94

Sehr geehrter Herr Grabinger,

Ihre Fragen wollen wir gerne beantworten:

1. Unsere Mehrweg-Bierflaschen werden ca. 40-50 mal benutzt.  
Danach können sie nicht mehr zur Verwendung kommen.
2. Eine Mehrweg-Bierflasche kostet DM 0,22 im Einkauf, Einwegflaschen sind teurer da weniger gekauft werden.
3. Die Reinigung kostet ca. 0,8 Pf / Flasche.

Falls Sie noch weitere Fragen haben sind wir gerne bereit, Auskunft zu geben.  
Wir wünschen Ihnen weiterhin viel Freude beim Genießen der Bellheimer Biere.



13.04.94

Ihr Schreiben vom 12.03.94 - "Pfandflasche"

Sehr geehrter Herr Grabinger,

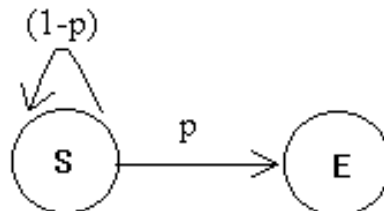
bezüglich Ihrer Bitte um Zahlen über Rückgabe von Pfandflaschen müssen wir Ihnen leider mitteilen, daß in unserem Hause keine laufenden, detaillierten Aufzeichnungen hierüber geführt werden.

Vielleicht können Ihnen aber folgende Informationen weiterhelfen:

Von der Finanzverwaltung werden zum Zwecke der Bildung von Pfandrückstellungen Umlaufzeiten von ca. 6 Wochen angenommen. Diese Zahl beruht auf repräsentativen Erhebungen und kann als realistisch angesehen werden. Nach unseren Erfahrungen beträgt die durchschnittliche Rücklaufquote ca. 96 - 98 %. Im Laufe ihres Lebens kann eine Pfandflasche ca. 20 mal befüllt und in den Nutzungskreislauf gebracht werden.

## Modell für den Lebenslauf einer Mehrwegflasche

Mit  $p$  werde die Wahrscheinlichkeit bezeichnet, daß eine gerade benutzte Flasche nicht wieder zur Verwendung kommt, sondern durch Bruch oder andersweitige Verwendung aus dem Umlaufvorgang ausscheidet. Der Lebenslauf der Flasche läßt sich als eine Irrfahrt auf dem folgenden Graphen deuten.



Zu Beginn, d.h. nach dem 1. Kauf, befindet sich die Flasche im Startzustand S. Mit Wahrscheinlichkeit  $p$  gerät die Flasche dann in den Endzustand E, d.h. sie wird nicht zurückgegeben. Mit Wahrscheinlichkeit  $q=1-p$  wird sie ein weiteres Mal verkauft, d.h. sie bleibt im Zustand S.

## Problemlösung durch Simulation

Zunächst soll die Situation durch eine Simulation nachgebildet werden: Dazu betrachte man eine Zufallsgröße  $Z$ , die nur die Werte 0 und 1 annehmen kann.

Der Wert 0 soll mit der Wahrscheinlichkeit  $p$ , der Wert 1 mit der Wahrscheinlichkeit  $q=(1-p)$  auftreten.

$$Z = \begin{cases} 0 & \text{mit Wahrscheinlichkeit } p \\ 1 & \text{mit Wahrscheinlichkeit } q \end{cases}$$

Durch Ziehen einer Zufallszahl wird entschieden, ob in den Zustand S zurückgekehrt wird, oder ob in den Zustand E übergegangen wird. Hat die gezogene Zufallszahl den Wert 1, so wird der Zustand S angenommen, tritt jedoch der Wert 0 auf, dann tritt der Endzustand E ein.

Auf dem folgenden Arbeitsblatt soll der Schüler mit Hilfe einer Zufallszahlentabelle das Pfandflaschenproblem simulieren. Die Simulation "per Hand" soll den Schüler mit dem Modell vertraut machen.

## Arbeitsblatt 1

### Simulation mit einer Tabelle von Zufallszahlen

Zufallszahlen									
12920	37375	82211	84059	30983	25736	83674	36822	57880	54822
98115	34078	69321	37344	34876	26423	12698	57308	24210	91260
80758	82137	96662	23487	33816	17475	75844	93090	56434	34356
33015	11804	59519	23422	76918	76058	63761	55873	77311	78367
25362	55858	45625	04372	89983	13960	21184	57711	85853	96499
36441	79914	91978	30312	59642	57497	42380	67716	38375	33457
52314	61327	92436	65346	77392	03436	29450	97119	41554	09621
81677	36930	94829	51285	87454	58083	86810	66807	65702	65304
66800	05993	56931	93547	46547	87517	29193	31492	86516	74037
72416	86946	91638	94674	29543	36325	27464	69543	37938	25404
92671	16019	79473	13525	08845	41433	62752	78449	13500	51880
85898	24042	67245	31823	53341	57976	11053	74197	45050	31642
34569	65673	61648	59453	41385	56799	93836	59736	22224	71683
89577	84385	71981	36921	17464	29476	11539	34444	49729	44025
00314	07984	13304	79080	50456	77836	82398	44332	78650	43613
11008	64428	56452	79673	28812	52188	45097	52913	48877	40892
25720	17607	60624	42851	60918	66993	06479	26422	48602	46273
65656	28673	54077	56739	93512	45884	31266	05688	89891	69057
69051	81835	94927	56646	23634	60898	35712	61670	22881	71041
72291	55413	53567	72880	16400	14900	67477	33712	21406	66313

Es wird angenommen, daß die betrachtete Flasche mit  $p=0,1$  nicht mehr zurückgegeben wird.

1. Wie läßt sich durch Ziehen einer Zufallszahl nachbilden, ob die gerade benutzte Flasche wieder zurückgegeben wird?
  
  
  
  
  
  
  
  
  
  
2. Simuliere den Lebenslauf einer Flasche, d.h. ziehe solange Zufallszahlen, bis sich erstmals ergibt, daß die Flasche nicht mehr zurückgegeben wird.
  
  
  
  
  
  
  
  
  
  
3. Wiederhole den unter 2. beschriebenen Vorgang möglichst oft. (Simulationen der Mitschüler sammeln) Ermittle für jede Simulation die Umlaufzahl. Bilde den Mittelwert.



Eine mögliche Lösung von Aufgabe 1:

Wandle die gezogene Zahl - durch Vorschalten von 0 in ein Element des Intervalls  $[0 ; 1 ]$  um.

Beispiel: Die ausgewählte Zahl: 56739 wird in  $x=0,56739$  umgewandelt.

Da das Ereignis

A: "Die Flasche wird nicht zurückgegeben"

mit  $p=0,1$  eintritt, ergibt sich folgende Interpretation:

A tritt ein, falls  $x \in [0 ; 0,1]$ ,

A tritt nicht ein, falls  $x \in ]0,1 ; 1)$

## Simulation in DERIVE

In **DERIVE** läßt sich die Erzeugung der Zufallsgröße  $Z$  mit der **IF**-Funktion erreichen:

$z := \mathbf{IF}(\mathbf{RANDOM}(1) \leq p, 0, 1)$

Liefert die Funktion **RANDOM(1)** einen Wert, der kleiner oder gleich  $p$  ist, dann nimmt  $z$  den Wert 0 an, sonst liefert  $z$  den Wert 1.

Anwendung von **Simplify** oder von **approximate** auf diese Zeile löst die Berechnung aus.

Die Funktionsweise von  $z := \mathbf{IF}(\mathbf{RANDOM}(1) \leq p, 0, 1)$  entspricht damit genau dem in der Lösung von Aufgabe 1 beschriebenen Vorgehen.

Auf dem folgenden Arbeitsblatt wird das Ziehen von Zufallszahlen "per Hand" auf die gerade dargestellte Art und Weise von **DERIVE** ersetzt.

## Arbeitsblatt 2

## Aufgabe 4

(Vergleiche zur Durchführung der Schritte 1-5 den unten abgebildeten DERIVE-Bildschirm.)

1. Weise der Variablen **p** den Wert **0.1** zu:
2. Gib die Zeile **z := IF(RANDOM(1) ≤ p, 0, 1)** ein.
3. Stelle den Leuchtbalken auf die Zeile **z := IF(RANDOM(1) ≤ p, 0, 1)** und wende **Simplify** an.
4. Wiederhole Schritt 3 solange, bis erstmals die Zahl **0** als Wert zurückgegeben wird.
5. Ermittle die Nummer des Experimentes, bei dem die Null erscheint.

**Die Nummer des Experimentes ist:**

```

1:  "p gibt die Wahrscheinlichkeit an, mit der die Flasche nicht"
2:  "mehr zurückgegeben wird."
3:  p := 0.1
4:  z := IF(RANDOM(1) ≤ p, 0, 1)
5:  1
6:  1
7:  1
8:  1
9:  1
10: 1
11: 0

```

---

COMMAND: **Author** Build Calculus Declare Expand Factor Help Jump soLve Manage  
Options Plot Quit Remove Simplify Transfer moVe Window approX  
Compute time: 0.0 seconds  
Simp(4) Free:100% Ins Derive Algebra

## Aufgabe 5

Hole bei den Mitschülern möglichst viele Simulationsergebnisse von Aufgabe 4 ein. Bestimme aus diesen Daten einen Näherungswert für die mittlere Rückgabezahl der Pfandflasche.

Simulation Nr.	1	2	3	4										
Zahl der Rückgaben														

**Die "Mittlere Umlaufdauer" der Flasche ist:**

## Aufgabe 6

Ermittle für verschiedene Werte von **p** die mittlere Umlaufdauer der Flasche. Ändere dazu die Zeile #3 des abgebildeten **DERIVE**-Bildschirms. Gehe dann wie in Aufgabe 4 vor. Stelle die Umlaufdauer in Abhängigkeit von **p** graphisch dar.

## Vereinfachung des Verfahrens

Die mühsame Simulation auf Arbeitsblatt 2 soll nun von **DERIVE** übernommen werden. Bei der Simulation, die auf dem abgebildeten **DERIVE**-Bildschirm (Arbeitsblatt 2) dargestellt ist, folgen auf 6 Einsen die erste Null, d.h. es ergab sich die Folge 1,1,1,1,1,1,0.

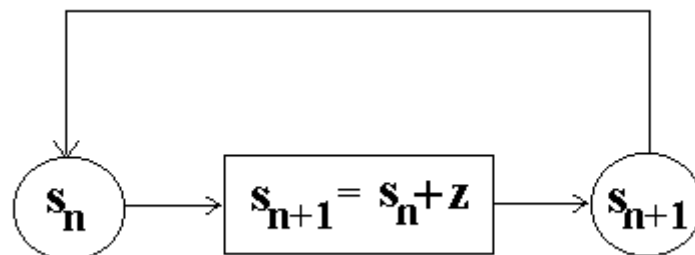
In einer Variablen  $s$  soll nun die Gesamtzahl aller Einsen der jeweils bis dahin durchgeführten Einzelsimulationen stehen.

$s$  hat zu Beginn der Simulation den Wert 0. Den weiteren Verlauf der Werte von  $s$  kann man bei dem betrachteten Beispiel 1,1,1,1,1,1,0 der folgenden Tabelle entnehmen.

Ergebnis $z$ der Einzelsimulation		1	1	1	1	1	1	0
Wert von $s$	0	1	2	3	4	5	6	7

Der neue Wert von  $s$  ergibt sich als Summe aus dem vorherigen Wert von  $s$  und dem Ergebnis  $z$  der Simulation. Beschreibt man diesen Zusammenhang mit einer Iterationsformel, so erhält man die Rechenvorschrift  $s_{n+1} = s_n + z$ , mit  $s_1 = 0$ .

Das nächste Diagramm veranschaulicht diesen Prozeß:



Diese Iteration wird durch die **ITERATES**-Anweisung **ITERATES(s+z, s, 0)** durchgeführt:

**ITERATES** führt die Iteration solange durch, bis der in der Iteration neu berechnete Wert erstmals einem schon in der Iteration dagewesenen Wert gleich wird.

Im vorliegenden Zahlenbeispiel wäre das Ergebnis von **ITERATES** der Vektor [0,1,2,3,4,5,6,7,7], d.h. die Flasche wurde 7 mal zurückgegeben, denn beim 8-ten Iterationsschritt hat  $z$  den Wert 0 geliefert.

Verwendet man statt **ITERATES(s+z, s, 0)** den Ausdruck **ITERATE(s+z, s, 0)**, so wird statt des ganzen Vektors nur dessen letztes Element zurückgegeben. (Im Beispiel die Zahl 7) Diese Zahl gibt an, wie oft die Flasche zurückgegeben wird.

Zur Durchführung einer großen Zahl von Simulationen eignet sich die **VECTOR**-Funktion. **VECTOR(ITERATE(s+z,s,0),i,1,10)** erzeugt einen Vektor, der die Zahl der Rückgaben von 10 Simulationen liefert.

## Arbeitsblatt 3

## Aufgabe 7

Gib die ersten drei Zeilen so in **DERIVE** ein, wie es der unten abgebildete **DERIVE**-Bildschirm zeigt. Wende dann auf Zeile #3 mehrmals **Simplify** an.

Führe durch Anwendung von Simplify mehrere Simulationen durch. Vergleiche dazu den abgebildeten **DERIVE**-Bildschirm:

```

1:  p := 0.1
2:  z := IF(RANDOM(1) ≤ p, 0, 1)
3:  ITERATES(s + z, s, 0)
4:  [0, 1, 1]
5:  [0, 1, 2, 3, 4, 5, 6, 7, 7]
6:  [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 9]
7:  ITERATE(s + z, s, 0)
8:  3
9:  18
10: 15
11: 4
12: 7

```

---

COMMAND: **Author** Build Calculus Declare Expand Factor Help Jump solve Manage  
Options Plot Quit Remove Simplify Transfer move Window approx

Compute time: 0.0 seconds

Simp(7) Free:100% Ins Derive Algebra

## Aufgabe 8

- 8.1 Erzeuge (mehrmals) durch **VECTOR(ITERATE(s+z,s,0),i,1,10)** einen Vektor, der die Ergebnisse von 10 Simulationen enthält.
- 8.2 Berechne aus den in 8.1 erzeugten Vektoren den Mittelwert der Umlaufdauer.
- 8.3 Benutze die **AVERAGE**-Funktion um den Mittelwert von 500 Simulationen zu berechnen: **AVERAGE(VECTOR(ITERATE(s+z,s,0),i,1,500))** .

## Lösung von Aufgabe 8

Eine mögliche Bearbeitung von Aufgabe 8 zeigt der folgende DERIVE-Bildschirm.

Fünfmalige Anwendung von **Simplify** auf Zeile #3 liefert 50 Simulationen. Mit dem Taschenrechner ergibt sich daraus ein Mittelwert von 7,2.

In Zeile #9 wird die Funktion **mittelwert** definiert. Wendet man **approx** auf **mittelwert** an, so erhält man jeweils den Mittelwert aus 500 Simulationen. Es ergibt sich dabei eine mittlere Umlaufdauer von etwa 9.

```
1:  p := 0.1
2:  z := IF(RANDOM(1) ≤ p, 0, 1)
3:  VECTOR(ITERATE(s + z, s, 0), i, 1, 10)
4:  [14, 0, 7, 4, 15, 18, 3, 9, 7, 11]
5:  [11, 7, 20, 23, 4, 6, 0, 0, 5, 16]
6:  [2, 8, 3, 3, 3, 7, 16, 4, 14, 11]
7:  [2, 3, 25, 32, 9, 3, 2, 2, 3, 9]
8:  [4, 1, 0, 7, 2, 2, 2, 4, 8, 3]
9:  mittelwert := AVERAGE(VECTOR(ITERATE(s + z, s, 0), i, 1, 500))
10: mittelwert
11: 9
12: 8.99
```

---

COMMAND: **Author** Build Calculus Declare Expand Factor Help Jump solve Manage  
Options Plot Quit Remove Simplify Transfer move Window approx  
Compute time: 16.0 seconds  
Approx(10) Free:98% Ins Derive Algebra

## Arbeitsblatt 4

## Aufgabe 9

Ändere den Wert von  $p$  mehrmals und führe die Aufgabe 8.3 erneut durch. Fülle eine Tabelle der folgenden Form aus.

$p$	0.1	0.2	0.3	0.4	0.5
Mittelwert der Umlaufdauer $u$					

## Aufgabe 10

Erzeuge eine Tabelle wie in Aufgabe 9. Dabei soll der Wert von  $p$  zwischen 0,1 und 0,5 mit der Schrittweite 0,02 variiert werden. Die sich ergebenden Wertepaare  $(p,u)$  sollen graphisch dargestellt werden.

## Aufgabe 11

Wie hängt die Umlaufdauer  $u$  von  $p$  ab?

1. Für sehr kleine Werte von  $p$  wird die Flasche fast immer zurückgegeben, d.h. für  $p$  gegen Null ist eine senkrechte Asymptote zu erwarten.
2. Für den Maximalwert, den  $p$  annehmen kann, nämlich  $p=1$ , erlebt die Flasche keine einzige Rückgabe, d.h. die Umlaufdauer muß Null sein.

Diese beiden Feststellungen führen zu der Vermutung, daß der Zusammenhang zwischen Umlaufdauer  $u$  und Wahrscheinlichkeit  $p$  durch eine verschobene Hyperbel der Form

$f(p) = \frac{1}{p} - c$  gegeben sein könnte. Welchen Wert müßte  $c$  haben, damit eine

Übereinstimmung mit der zuvor getroffenen Aussage 2 besteht?

Zeichne verschobene Hyperbeln der Form

$$f(p) = \frac{1}{p} - c; c \in \{0; 0,5; 1; 1,5; 2\}$$

und vergleiche mit den aus der Simulation gewonnenen Werten.

Lösung von Aufgabe 9 Zeilen #1-#4

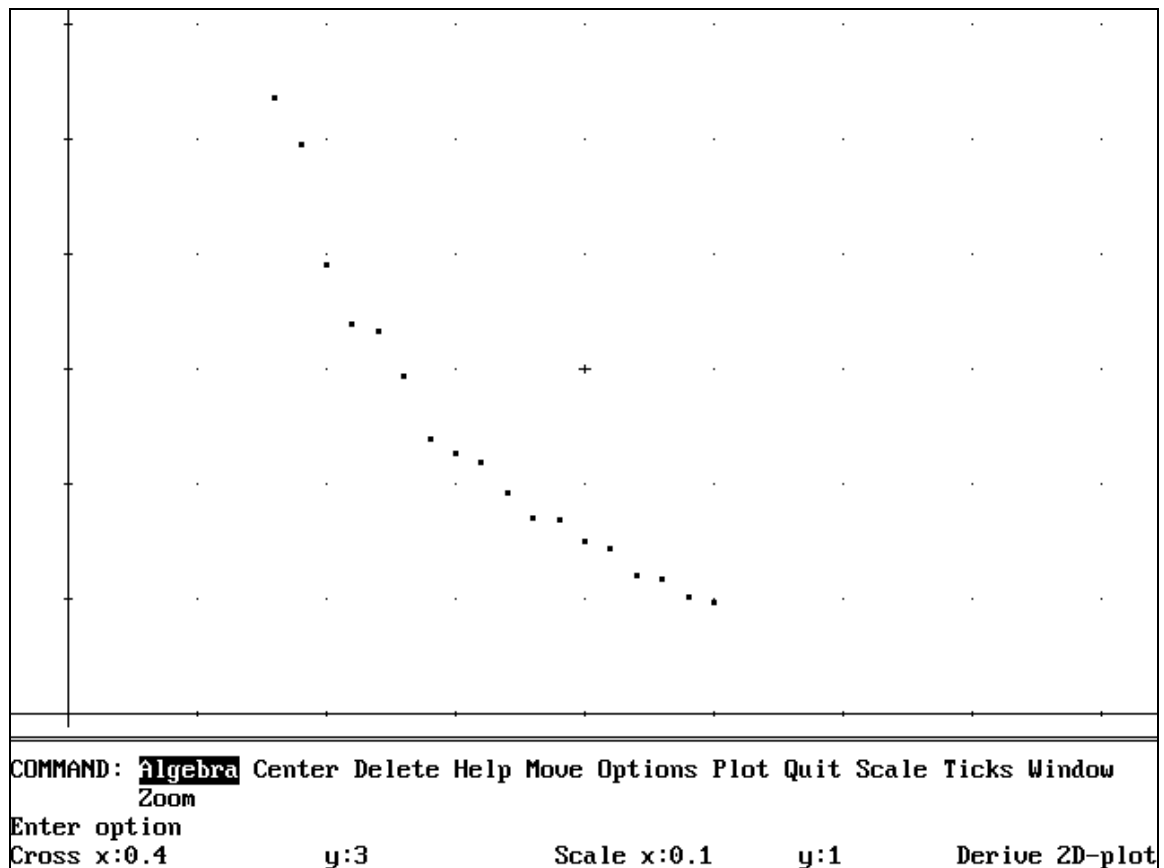
Lösung von Aufgabe 10 Zeilen #1-#6, die Rechenzeit kann einige Minuten dauern.

```

1: z := IF(RANDOM(1) ≤ p, 0, 1)
2: mittelwert := AVERAGE(VECTOR(ITERATE(s + z, s, 0), i, 1, 500))
3: VECTOR([p, mittelwert], p, 0.1, 0.5, 0.1)
4: [ 0.1  8.758
    0.2  3.85
    0.3  2.214
    0.4  1.516
    0.5  1.06 ]
5: VECTOR([p, mittelwert], p, 0.1, 0.5, 0.02)
6: [[0.1, 9.884], [0.12, 7.42], [0.14, 6.318], [0.16, 5.352], [0.18, 4.946], [0.2, 4.544]]

COMMAND: Author Build Calculus Declare Expand Factor Help Jump solve Manage
Options Plot Quit Remove Simplify Transfer move Window approx
Compute time: 136.2 seconds
Approx(5) Free:98% Ins Derive Algebra

```

Anwendung von **Plot** auf die Zeile #6 führt zur gewünschten graphischen Darstellung:

## Lösung von Aufgabe 11

Anwendung von **Simplify** auf die Zeile #7 liefert die zu untersuchende Funktionenschar in Zeile #8.

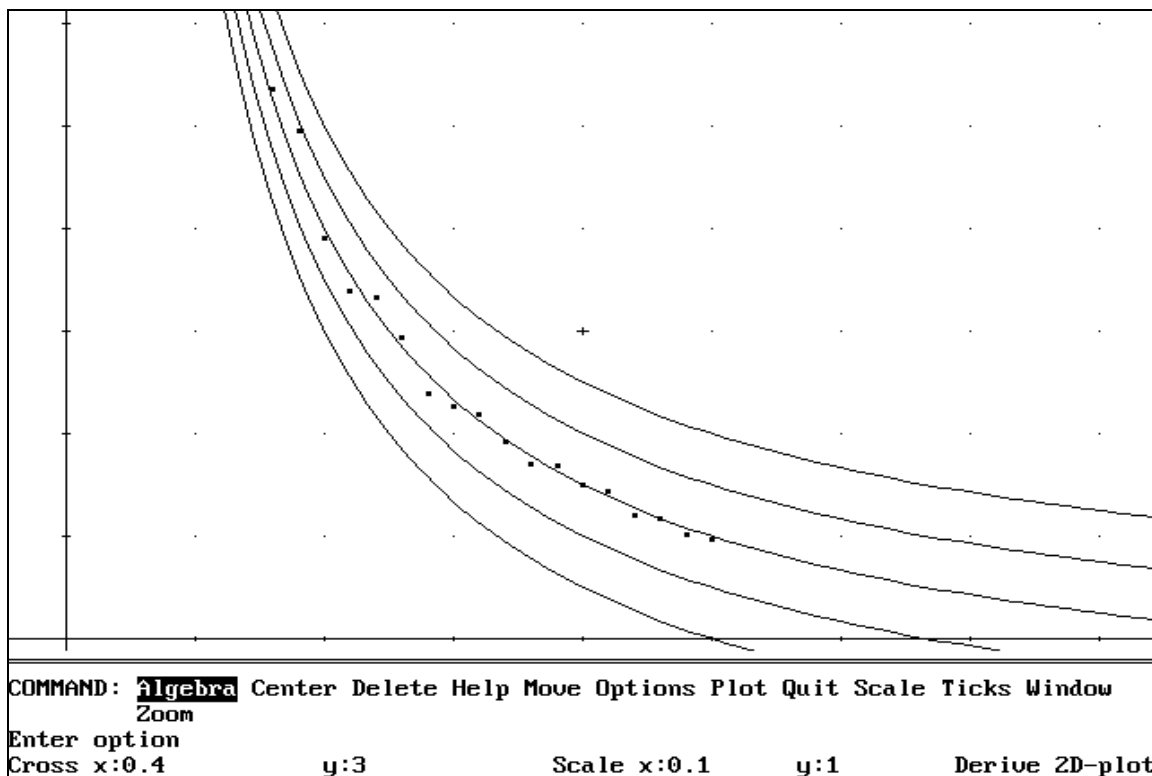
```

2: mittelwert := AVERAGE(VECTOR(ITERATE(s + z, s, 0), i, 1, 500))
3: VECTOR([p, mittelwert], p, 0.1, 0.5, 0.1)
4: [ 0.1  8.758
    0.2  3.85
    0.3  2.214
    0.4  1.516
    0.5  1.06 ]
5: VECTOR([p, mittelwert], p, 0.1, 0.5, 0.02)
6: [[0.1, 9.884], [0.12, 7.42], [0.14, 6.318], [0.16, 5.352], [0.18, 4.946], [
7: VECTOR[ $\frac{1}{x} - c, c, 0, 2, 0.5$ ]
8: [ $\frac{1}{x}, \frac{1}{x} - \frac{1}{2}, \frac{1}{x} - 1, \frac{1}{x} - \frac{3}{2}, \frac{1}{x} - 2$ ]

COMMAND: Author Build Calculus Declare Expand Factor Help Jump solve Manage
Options Plot Quit Remove Simplify Transfer move Window approx
Enter option
Simp(7) Free:98% Ins Derive Algebra

```

c muß den Wert 1 haben. Das zeigt auch die folgende graphische Darstellung, die durch plotten von #8 entstand.





## Theorie

Die Bearbeitung von Aufgabe 11 läßt vermuten, daß die Funktion  $f(x) = \frac{1}{x} - 1$  die Punktmenge am besten annähert. Daraus würde folgen, daß der gesuchte Zusammenhang zwischen Umlaufdauer  $u$  und Wahrscheinlichkeit  $p$  durch  $u = \mathbf{Fehler! - 1}$  gegeben ist. Läßt sich dieses Ergebnis auch aus der Theorie erhalten?

Mit  $w(k)$  werde die Wahrscheinlichkeit dafür bezeichnet, daß die Flasche nach dem  $k$ -ten Verkauf nicht mehr zurückgegeben wird. Mit der Pfadregel erhält man:

$$w(k) = p(1-p)^{(k-1)}.$$

Eine Zufallsgröße  $X$  mit der Wahrscheinlichkeitsfunktion  $P(X=k) = w(k)$ ,  $k \in \{1, 2, 3, \dots\}$  heißt geometrisch verteilt.

Folgende Probleme stellen sich in diesem Zusammenhang:

1. Zeige, daß  $\sum_{k=1}^{\infty} w(k) = 1$  gilt.
2. Berechne die Verteilungsfunktion  $P(X \leq n) = \sum_{k=1}^n w(k)$ .
3. Berechne den Erwartungswert  $E(X) = \sum_{k=1}^{\infty} k w(k)$  von  $X$ .
4. Berechne die Varianz  $\text{VAR}(X) = \sum_{k=1}^{\infty} (k - E(X))^2 w(k)$  von  $X$ .
5. Welche Folgerung ergibt sich aus 3. für die Umlaufdauer  $u$  der Pfandflasche?

## Arbeitsblatt 5

## Aufgabe 12

Definiere in **DERIVE** eine Variable **p** aus dem offenen Intervall  $]0; 1[$   
 Führe dazu folgende Schritte im **DERIVE**-Menüsystem durch:

- **Declare:** **Variable**
- **DECLARE VARIABLE name:** **p**
- **DECLARE VARIABLE:** **Interval**

(Der folgende Ausschnitt des **DERIVE**-Bildschirms zeigt, wie nach diesen Eingaben noch die Intervallgrenzen einzustellen sind:)

```

DECLARE VARIABLE: Bounds: 0      (<)≤  p (<)≤  1
Enter right bound
Free: 100% Ins

```

**Beachte, daß bei der Durchführung der Aufgaben 13-17 die Variable p wie angegeben definiert sein muß!**

## Aufgabe 13

Definiere - nach der Durchführung von Aufgabe 12 - die Funktion  $w(k)=p(1-p)^{(k-1)}$  in **DERIVE**.

## Aufgabe 14

Zeige, daß  $\sum_{k=1}^{\infty} w(k) = 1$  gilt

- a. durch Rechnen in **DERIVE**.  
(Die Eingabe der Summe geschieht durch **SUM(w(k), k, 1, inf)**)
- b. unter Benutzung der Summenformel für die unendliche geometrische Reihe.

## Aufgabe 15

Bestimme die Wahrscheinlichkeit, daß das Ereignis "Die Flasche wird nicht zurückgegeben" spätestens nach dem n-ten Verkauf eintritt,

- a. durch Rechnen in **DERIVE**.
- b. mit Hilfe der Summenformel für die endliche geometrische Reihe.

## Aufgabe 16

Wie lange dauert es im Mittel, bis das Ereignis "Die Flasche wird nicht zurückgegeben" eintritt? Bilde dazu den Erwartungswert  $\sum_{k=1}^{\infty} k w(k)$ . Lasse die Berechnung dieser Summe von **DERIVE** durchführen.

## Aufgabe 17

Der in Aufgabe 16 zu berechnende Erwartungswert ist **Fehler!**. Benutze dieses Ergebnis, um mit Hilfe von **DERIVE** die Varianz  $\sum_{k=1}^{\infty} (k - \frac{1}{p})^2 w(k)$  zu berechnen.

## Lösung der Aufgaben 13-16

Nach Bearbeitung dieser Aufgaben sollte der **DERIVE**-Bildschirm das folgende Aussehen haben:

```

1:  W(k) := p (1 - p)k - 1
2:  ∑k=1∞ W(k)
3:  1
4:  ∑k=1n W(k)
5:  1 - (1 - p)n
6:  ∑k=1∞ k W(k)
7:   $\frac{1}{p}$ 

```

---

COMMAND: **Author** Build Calculus Declare Expand Factor Help Jump solve Manage  
Options Plot Quit Remove Simplify Transfer move Window approx  
Compute time: 0.1 seconds  
Simp(6) Free:100% Ins Derive Algebra

Als Lösung von Aufgabe 17 ergibt sich  $\frac{1-p}{p^2}$ .

## Schlußbetrachtung

Aus der Zeile #7 des letzten **DERIVE** - Bildschirms ergibt sich für die Umlaufdauer  $u$  einer Pfandflasche:

$$u = E(X) - 1 = \mathbf{Fehler!} - 1.$$

Dies steht in guter Übereinstimmung mit den aus der Simulation gewonnenen Ergebnissen.

Es wird nun auch ein Vergleich mit den aus der Praxis beobachteten Werten möglich. In den Materialien des Abschnitts "Einweg- und Mehrwegflaschen" findet sich die Angabe einer Brauerei, daß "die durchschnittliche Rücklaufquote der Flaschen ca 96%-98% beträgt."

Weiter liest man: "Im Laufe ihres Lebens kann eine Pfandflasche ca. 20 mal befüllt und in den Nutzungskreislauf gebracht werden.

Setzt man für  $u$  den Wert 20 an, so erhält man aus  $u = \mathbf{Fehler!} - 1$  für  $p$ :  $p \approx 4,8\%$ . Dem entspricht eine Rückgabequote von  $q = 1 - p = 1 - 4,8\% = 95,2\%$ . Dies steht in Einklang mit den Angaben der Brauerei.

## Anhang

### Der Zufallszahlengenerator von DERIVE

---

Die Erzeugung von Zufallszahlen hat historische Dimensionen. Lange vor der Verwendung des Computers wurden Zufallszahlen erzeugt. Hier einige Beispiele:

Wie die Evangelisten übereinstimmend berichten, wurde schon um den Rock Christi gewürfelt. "Sie haben meine Kleider unter sich geteilt und das Los um mein Gewand geworfen." (Johannes 19,23 f)

Die Beliebtheit des Lottospiels ist nicht auf unsere Tage beschränkt. Selbst Casanova handelte nach dem Motto "erst Lotto, dann Liebe": Er ging auf seinen Reisen in den Städten zunächst immer erst Lotto spielen, was seine ausgeprägtere Leidenschaft war. (W. Böhm, Vom Glück der Zahlen, Mannheimer Morgen, 19.3.1988)

Auch die Kirche scheint dem Losverfahren - und damit den Zufallszahlen - nicht abwertend gegenüber zu stehen, liest man doch bei Augustinus (Enarrationes in Psalmos), daß "das Los kein Übel an sich ist, sondern etwas, das dem zweifelnden Menschen den Willen Gottes anzeigt."

Der Anspruch von **DERIVE** bei der Erzeugung von Zufallszahlen ist wesentlich bescheidener. Hier waltet nicht der Wille Gottes sondern ein "Linearer Kongruenzgenerator". Durch einen derartigen Generator wird eine Zahlenfolge geliefert, die scheinbar zufällig ist, tatsächlich aber ist der Erzeugungsprozeß streng deterministisch. Man spricht deshalb auch von Pseudozufallszahlen. Zunächst soll gezeigt werden, wie der **DERIVE**-Generator benutzt wird. Daran schließen sich Untersuchungen zur Güte der erzeugten Zufallsziffern an. In einem weiteren Abschnitt wird vorgestellt, wie der **DERIVE**-Generator aufgebaut ist. Zu jedem Abschnitt werden Aufgaben mit Lösungen angeboten.

#### Die Benutzung des DERIVE-Generators

Folgende Funktionen stehen in **DERIVE** zur Verfügung:

- |                                   |   |
|-----------------------------------|---|
| <b>RANDOM(m), m&gt;1, ganz</b>    | Eine ganzzahlige gleichverteilte Zufallszahl aus [0 ; m[ wird erzeugt.  |
| <b>RANDOM(1)</b>                  | Eine gleichverteilte Zufallszahl aus [0 ; 1[ wird erzeugt.  |
| <b>RANDOM(m), m negativ, ganz</b> | Der Generator wird mit (- m) initialisiert. Beispiel: RANDOM(-10) sorgt dafür, daß die erste erzeugte Zahl +10 ist. |

## Aufgaben

1. Simuliere mit dem **DERIVE**-Generator 5 Würfelwürfe.
2. Erzeuge eine auf  $[3 ; 5[$  gleichverteilte Zufallszahl.
3. Erzeuge eine gleichverteilte ganze Zufallszahl aus der Menge  $\{r, r+1, \dots, r+s\}$ , für  $r=2$  und  $s=5$ .
4. Erzeuge eine bernoulli-verteilte Zufallszahl mit folgender Wahrscheinlichkeitsfunktion:

$x_j$	0	1
$P(X=x_j)$	0,5	1-0,5

5. Erzeuge eine binomial-verteilte Zufallszahl mit Parameter  $n=10$  und  $p=0,5$ .  
Anleitung: Die binomial-verteilte Zufallszahl gibt die Zahl der "Erfolge" in einer Kette von  $n$  Bernoulli-Experimenten an. Deshalb sind  $n$  bernoulli-verteilte Zufallszahlen - mit einer Wahrscheinlichkeitsfunktion wie aus Aufgabe 4 - zu summieren. Die Summation kann mit der **ITERATE** -Anweisung durchgeführt werden.
6. Es werden  $n=100$  Glühbirnen mit einer Ausschußrate von  $p=5\%$  produziert. Ermittle durch Simulation die zu erwartende Anzahl defekter Glühbirnen.

Eine mögliche Bearbeitung der Aufgaben 1-6 zeigen die beiden folgenden **DERIVE**-Bildschirme.:

Aufgabe 1 #1-#2      Aufgabe 2 #3-#4      Aufgabe 3 #5-#6  
Aufgabe 4 #7-#8      Aufgabe 5 #9-#10

```
1:  VECTOR(1 + RANDOM(6), i, 5)
2:  [5, 1, 5, 4, 2]
3:  (5 - 3) RANDOM(1) + 3
4:  4.28934
5:  FLOOR((2 + 1) RANDOM(1)) + 5
6:  7
7:  z := IF(RANDOM(1) ≤ 0.5, 0, 1)
8:  0
9:  ITERATE(s + z, s, 0, 10)
10:  4
```

```
COMMAND: Author Build Calculus Declare Expand Factor Help Jump solve Manage
Options Plot Quit Remove Simplify Transfer move Window approx
Compute time: 0.0 seconds
Simp(9)                    C:\DERIVE\GEN.MTH      Free:100% Ins                    Derive Algebra
```

Aufgabe 6:

```
1:  "Bernoulli-verteilte Zufallszahl mit Parameter p:"
2:  Z(p) := IF(RANDOM(1) < p, 1, 0)
3:  "Binomial-verteilte Zufallszahl mit Parametern n und p:"
4:  B(n, p) := ITERATE(s + Z(p), s, 0, n)
5:  B(100, 0.05)
6:  "approx #5: "
7:  6
8:  VECTOR(B(100, 0.05), i, 10)
9:  [6, 7, 3, 2, 5, 5, 4, 9, 9, 7]
10: AVERAGE(VECTOR(B(100, 0.05), i, 100))
11: 4.92
```

```
COMMAND: Author Build Calculus Declare Expand Factor Help Jump solve Manage
Options Plot Quit Remove Simplify Transfer move Window approx
Enter option
User                    C:\DERIVE\BINZ.MTH      Free:99% Ins                    Derive Algebra
```

## Die Güte des DERIVE-Generators

Wie gut sind die vom **DERIVE**-Generator erzeugten Zufallszahlen?

1. Einen ersten Eindruck erhält man durch eine zeichnerische Darstellung. Dazu interpretiert man je zwei aufeinanderfolgende Zufallszahlen als Koordinaten eines Punktes der Ebene. Eine unregelmäßige Verteilung der Zufallspunkte zeugt von der Güte der hergestellten Zufallszahlen.
2. Einen weiteren Hinweis für die Güte der Zufallszahlen kann man man durch die Bildung ihrer relativen Häufigkeiten erhalten. Erzeugt man beispielsweise eine größere Anzahl von Zufallsziffern zwischen 0 und 9 (einschließlich), so sollten deren relative Häufigkeiten etwa  $\frac{1}{10}$  sein.
3. Schließlich läßt sich mit dem  $\chi^2$ -Test die Hypothese einer Gleichverteilung der erzeugten Zufallszahlen überprüfen.

In den folgenden Aufgaben werden die erwähnten Möglichkeiten bearbeitet.

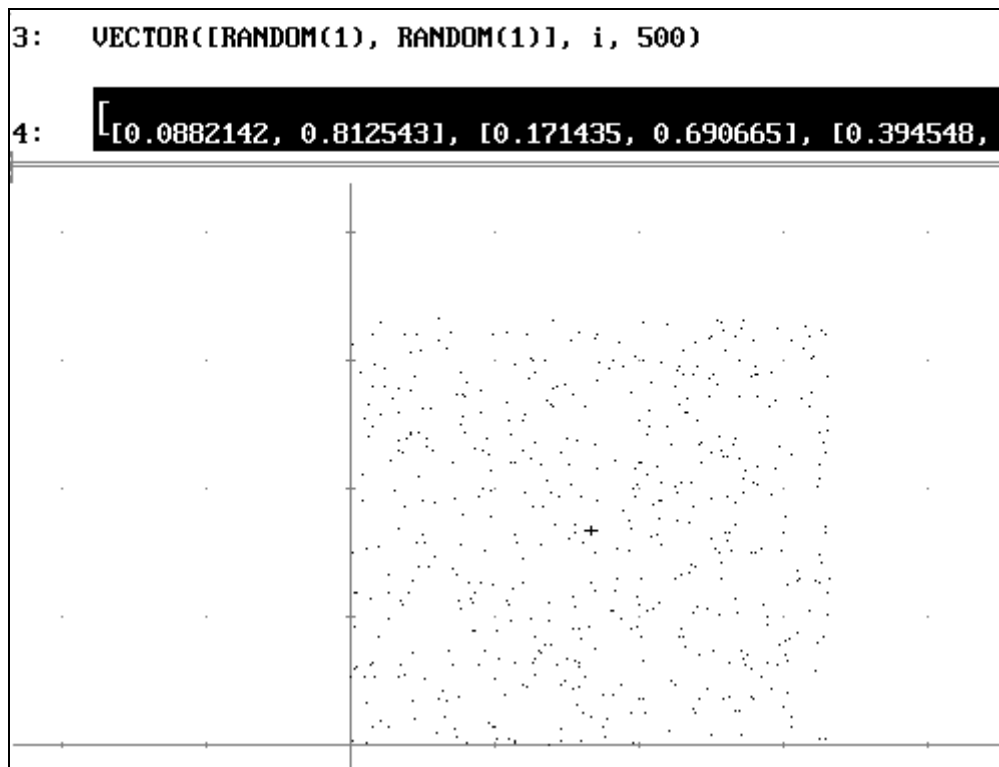
### Aufgaben

1. Erzeuge 500 Zufallspunkte im Einheitsquadrat der xy-Ebene.  
Anleitung: Einen einzelnen Zufallspunkt erhält man mit [**RANDOM(1)** , **RANDOM(1)**]. Bilde einen Vektor aus 500 Punkten und wende den **Plot**-Befehl an.
2. Erzeuge 1000 Zufallszahlen aus der Menge {0,1,...,9} und ermittle für jede Zahl die Häufigkeit, mit der sie auftritt.
3. Wende den  $\chi^2$ -Test an, um zu testen, ob die in der Aufgabe 2 ermittelten relativen Häufigkeiten mit einer Gleichverteilung vereinbar sind.



Eine mögliche Bearbeitung der Aufgaben 1-3

Zu Aufgabe 1:



Zu Aufgabe 2:

Die Indikatorfunktion **IND(x,y)** liefert den Wert 1, wenn x und y gleich sind. Um zu zählen, wie oft die Zahl x im Vektor v auftritt, kann man wie folgt vorgehen:

Für jedes Element des Vektors v ermittelt man mit der **IND**-Funktion, ob dieses gleich x ist. Die Aufsummierung dieser Indikatorwerte liefert das absolute Vorkommen **ANZ(x,v)** von x im Vektor v.

```
1: IND(x, y) := IF(x = y, 1, 0)
```

```
2: ANZ(x, v) :=  $\sum_{i=1}^{\text{DIMENSION}(v)}$  IND(ELEMENT(v, i), x)
```

```
3: VECTOR(RANDOM(10), i, 1000)
```

```
4: [9, 0, 5, 8, 1, 2, 0, 9, 8, 8, 7, 1, 5, 2, 9, 8, 5, 2, 5, 6, 6, 1, 6, 3
```

```
5: v := [9, 0, 5, 8, 1, 2, 0, 9, 8, 8, 7, 1, 5, 2, 9, 8, 5, 2, 5, 6, 6, 1,
```

```
6: VECTOR(ANZ(x, v), x, 0, 9)
```

```
7: [91, 92, 110, 98, 96, 108, 98, 85, 110, 112]
```

Zu Aufgabe 3:

Die zu prüfende Hypothese lautet:

$H_0$ : Die Ziffern 0,1,...,9 treten mit den Wahrscheinlichkeiten  $p_0 = p_1 = \dots = p_9 = 0,1$  auf.

Die Gegenhypothese  $H_1$  besagt, daß eine andere Verteilung der Wahrscheinlichkeiten vorliegt.

Es war sicher nicht zu erwarten, daß jede Ziffer mit der Häufigkeit 10 auftritt. Zur Überprüfung der Hypothese  $H_0$  ist zunächst eine Testgröße festzulegen, mit der die Abweichung vom Idealfall gemessen werden kann. Mit  $N_i$  wird die absolute Anzahl des Auftretens von Ziffer  $i$  bezeichnet. Insgesamt werden  $n$  Ziffern erzeugt. Der Erwartungswert für das Auftreten der Ziffer  $i$  ist dann  $E(N_i) = n \cdot p_i = n \cdot 0,1$ . ( $N_i$  ist binomialverteilt !)

Die Abweichung der Anzahl  $N_i$  vom Idealfall ist deshalb durch  $N_i - np_i$  gegeben. Bildet man die Summe aller 10 Abweichungen, so ist darauf zu achten, daß die Differenzen  $N_i - np_i$  unterschiedliche Vorzeichen haben können. Bei reiner Addition dieser Größen würde sich dann eine zu kleine Gesamtabweichung ergeben. Um dies zu verhindern, bildet man die Summe der quadrierten Abweichungen  $(N_i - np_i)^2$ .

Außerdem wird jeder einzelne Summand noch durch  $np_i$  geteilt. Die so gebildete Testgröße wird mit  $\chi^2$  bezeichnet.

$$\chi^2 = \frac{(N_0 - np_0)^2}{np_0} + \frac{(N_1 - np_1)^2}{np_1} + \dots + \frac{(N_9 - np_9)^2}{np_9}$$

Wenn  $\chi^2$  einen großen Wert annimmt, so heißt dies, daß die beobachteten Abweichungen groß sind. In diesem Fall muß  $H_0$  verworfen werden. Wann liegen "große Werte" von  $\chi^2$  vor?

Wie bei Testkonstruktionen üblich, wird der Rückweisbereich des Tests so bestimmt, daß die Wahrscheinlichkeit für einen Fehler 1.Art eine bestimmte Schranke  $\alpha$  nicht übersteigt. Die Zahl  $x$ , ab der  $H_0$  abgelehnt wird, ist deshalb so zu bestimmen, daß die folgende Ungleichung erfüllt ist:  $P_{H_0}(\chi^2 > x) < \alpha$ .

Der Ausdruck auf der linken Seite des Kleiner-Zeichens kann umgeformt werden zu:

$$P_{H_0}(\chi^2 > x) = 1 - P_{H_0}(\chi^2 \leq x)$$

Die Wahrscheinlichkeiten  $P_{H_0}(\chi^2 \leq x)$  können einer Tafel der  $\chi^2$ -Verteilung entnommen werden. Sie können aber auch direkt mit Hilfe von **DERIVE** ermittelt werden:

Nach Laden der **DERIVE**-Hilfsdatei **PROBABIL** steht die Funktion **CHI\_SQ(x,f)** zur Verfügung. Diese liefert als Ergebnis  $P_{H_0}(\chi^2 \leq x)$ . Dabei bedeutet  $f$  die Zahl der sogenannten Freiheitsgrade:  $f = n-1$ .

Der folgende **DERIVE**-Bildschirm zeigt  $P_{H_0}(\chi^2 \leq x)$  in Abhängigkeit von  $x$ . Es ist  $f=10-1$  gewählt. Daraus ist ersichtlich, daß  $P_{H_0}(\chi^2 \leq x)$  ab  $x=17$  kleiner als 5% wird. Nimmt  $\chi^2$  einen Wert an, der größer oder gleich 17 ist (Literaturwert: 16,92), so muß  $H_0$  verworfen werden.

```

1:  VECTOR([x, 1 - CHI_SQ(x, 9)]1, x, 10, 20)

      10  0.350432
      11  0.275702
      12  0.213219
      13  0.162494
      14  0.122409
2:  15  0.0912927
      16  0.0668508
      17  0.0486621
      18  0.0351450
      19  0.0254849
      20  0.0181755

COMMAND: Author Build Calculus Declare Expand Factor Help Jump solve Manage
          Options Plot Quit Remove Simplify Transfer move Window approx
Compute time: 6.9 seconds
User      B:CHI.MTH      Free:96%  Ins      Derive Algebra

```

Die nächsten Überlegungen zeigen, wie  $\chi^2$  aus den ermittelten Zufallsziffern gebildet werden kann.

Die Testgröße  $\chi^2$  ist gegeben durch  $\chi^2 = \sum_{i=1}^n \frac{(N_i - np_i)^2}{np_i}$ , wobei  $N_i$  die absoluten Häufigkeiten und  $p_i$  die theoretischen Wahrscheinlichkeiten der betrachteten Merkmalsausprägungen der  $n$  Werte sind. Führt man die relativen Häufigkeiten  $h_i = \frac{N_i}{n}$  ein, so läßt sich  $\chi^2$  in folgender Form

schreiben:  $\chi^2 = n \sum_{i=1}^n \frac{(h_i - p_i)^2}{p_i}$ .

Auf diese Weise ist **CHIQUADRAT** im abgebildeten **DERIVE**-Bildschirm erklärt. Die relativen Häufigkeiten der erzeugten Zufallszahlen sind im Vektor **hk** gesammelt worden. Die Theoriewerte der Wahrscheinlichkeiten der Zufallsziffern sind jeweils 0,1. Aus ihnen wurde der Vektor **pk** gebildet. Die Zahlen  $h_i$  und  $p_i$  ergeben sich dann jeweils als  $i$ -tes Element dieser Vektoren.

```

7:  [91, 92, 110, 98, 96, 108, 98, 85, 110, 112]
8:  CHIQUADRAT(hk, pk, n, r) := n * Σi=1r  $\frac{(\text{ELEMENT}(hk, i) - \text{ELEMENT}(pk, i))^2}{\text{ELEMENT}(pk, i)}$ 
9:  hk := [0.091, 0.092, 0.11, 0.098, 0.096, 0.108, 0.098, 0.085, 0.11, 0.112]
10: pk := [0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]
11: CHIQUADRAT(hk, pk, 1000, 10)
12: 8.02

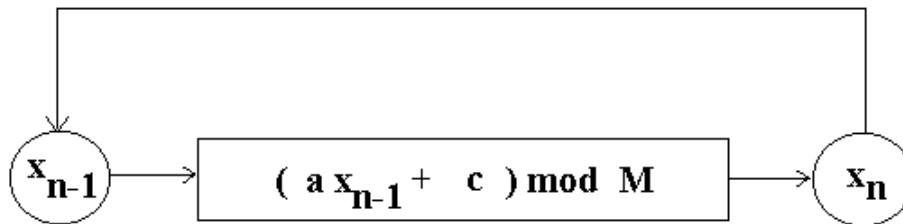
```

Für  $f=10-1 = 9$  und  $\alpha=5\%$  ist die Schranke für den Rückweisbereich des  $\chi^2$ -Tests durch 16,92 gegeben.

Da  $8,02 < 16,92$  ist, kann die Hypothese einer Gleichverteilung nicht abgelehnt werden.

## Die Funktionsweise des DERIVE-Generators

Bei dem **DERIVE**-Zufallszahlengenerator handelt es sich um einen linearen Kongruenzgenerator. Die Folge der Zufallsziffern wird bei einem linearen Kongruenzgenerator iterativ durch das Verfahren  $x_n = (a x_{n-1} + c) \bmod M$  mit  $0 \leq x_0, a, c < M$  gebildet. Dabei ist  $x_0$  eine willkürliche Startzahl, die den Zufallszahlengenerator initialisiert. Die Vorschrift "mod" gibt an, daß die neue Zufallszahl  $x_n$  sich als Rest der ganzzahligen Division von  $(a x_{n-1} + c)$  durch  $M$  ergibt. Das folgende Diagramm veranschaulicht den Vorgang der Bildung von Zufallszahlen mit einem Kongruenzgenerator.



Für die Güte des Generators ist die Wahl der Zahlen  $a, c$  und  $M$  entscheidend.

### Beispiel

Wählt man  $x_0 = a = c = 7$  und  $m = 10$ , so ergibt sich die Zahlenfolge  $7, 6, 9, 0, 7, 6, 9, 0, \dots$ , die sicher nicht als zufällig erscheint. Schon bei der vierten Zahl gerät die Folge in einen Zyklus.

### Beispiel

Der Generator  $x_n = (x_{n-1} + 3) \bmod 10$  mit  $x_0 = 0$ , liefert der Reihe nach  $0, 3, 6, 9, 2, 5, 8, 1, 4, 7, 0, 3, \dots$ . Da bei dem Prozeß der Restbildung modulo 10 höchstens die 10 Reste  $0, 1, \dots, 9$  auftreten können, hat dieser Generator eine maximale Periodenlänge.

Es läßt sich zeigen (Knuth 1981), daß folgende Bedingungen für eine maximale Periodenlänge hinreichend und notwendig sind:

- $c$  und  $M$  sind teilerfremd.
- $a-1$  ist Vielfaches von  $p$ , für jeden Primfaktor  $p$  von  $M$ .
- $a-1$  ist Vielfaches von 4, falls  $M$  ein Vielfaches von 4 ist.

Der **DERIVE**-Zufallszahlengenerator das folgende Aussehen:

$$x_n = (3141592653 x_{n-1} + 1) \bmod 2^{32}.$$

## Aufgaben

1. Bilde den Term  $(3141592653 x_{n-1} + 1) \bmod 2^{32}$  in **DERIVE** nach..
2. Initialisiere den Generator mit 100000 und erzeuge die ersten fünf Zufallszahlen.
3. Zeige, daß der **DERIVE**-Generator maximale Periodenlänge besitzt.
4. Bestimme die Zeit, die das **DERIVE**-System benötigt um 1000 Zufallszahlen zu erzeugen. (Diese Zeit ist natürlich von der Hardware abhängig.)
5. Wie lange dauert es, bis der **DERIVE**-Generator die Periodenlänge durchlaufen hat? Gehe von der in Aufgabe 4 ermittelten Zeit aus.

Eine mögliche Bearbeitung der Aufgaben 1-5:

zu 1 und 2:

```

1:  MOD(3141592653 x + 1, 232)
2:  ITERATES(MOD(3141592653 x + 1, 232), x, 100000, 5)
3:  [100000, 3882434081, 3417921006, 527608983, 4014838636, 1808293757]

COMMAND: Author Build Calculus Declare Expand Factor Help Jump solve Manage
Options Plot Quit Remove Simplify Transfer move Window approx

```

zu 3: Die von Knuth angegebenen Bedingungen sind alle erfüllt. Der **DERIVE**-Generator besitzt deshalb die Periodenlänge  $2^{32} = 4294967296$

zu 4:

Die Erzeugung von 1000 Zufallszahlen mit dem Befehl `vector(random(1),i,1000)` braucht etwa 3 Sekunden. (Die Angabe hängt von der verwendeten Hardware ab.)

```

1:  VECTOR(RANDOM(1), i, 1000)
2:  [0.993211, 0.476578, 0.438999, 0.993288, 0.602238, 0.235918, 0.

COMMAND: Author Build Calculus Declare Expand Factor Help Jump solve
Options Plot Quit Remove Simplify Transfer move Window approx
Compute time: 3.2 seconds
Approx(1) Free:54% Ins De

```

zu 5:  $2^{32} \cdot 0,003 \text{ s} = 1,288 \cdot 10^7 \text{ s} = 3579,13 \text{ h} = 149,13 \text{ d.}$

## Nachbildung der Wirkung von RANDOM(m)

Der **DERIVE**-Generator liefert eine Zahl aus dem Intervall  $[0 ; 2^{32}[$ . Dividiert man diese Zahl durch  $2^{32}$ , so erhält man eine Zahl zwischen 0 und 1. Multiplikation mit  $m$  und Bildung des ganzzahligen Anteils liefert eine ganze Zahl aus  $[0 ; m[$ . Es sind demnach die folgenden Schritte durchzuführen:

1. Erzeugung einer ganzen Zahl aus  $[0 ; 2^{32}[$  mit dem Generator.
2. Division durch  $2^{32}$ .
3. Multiplikation mit  $m$  (z.B.  $m=10$ ) und Bildung des ganzzahligen Anteils.

Die Bildung des ganzzahligen Anteils leistet die **DERIVE**-Funktion **FLOOR**.

Der folgende **DERIVE**-Bildschirm zeigt die Durchführung der beschriebenen Schritte, wenn der Generator mit 100000 initialisiert wird.

```

1:  MOD(3141592653 100000 + 1, 232 )
2:  "Simplify liefert:"
3:  3882434081
   3882434081
   -----
4:  32
   2
5:  "approx auf 4 anwenden:"
6:  0.903949
7:  FLOOR(0.903949 10)
8:  "Simplify:"
9:  9

COMMAND: Author Build Calculus Declare Expand Factor Help Jump solve Manage
          Options Plot Quit Remove Simplify Transfer move Window approx
Compute time: 0.0 seconds
Simp(?)                               Free:100% Ins                Derive Algebra

```

Wiederholt man das eben beschriebene Verfahren mehrmals, so erhält man der Reihe nach die "Zufallszahlen" 9, 7, 1, 9, 4..

Dieselbe Folge von Zufallszahlen erhält man mit den Standard-**DERIVE**-Funktionen, wenn auch hier der Zufallszahlengenerator mit 100000 initialisiert wurde:

```
1:  RANDOM(-100000)
2:  100000
3:  RANDOM(10)
4:  9
5:  RANDOM(10)
6:  7
7:  RANDOM(10)
8:  1
```