Über die Erzeugung gleichverteilter Pseudozufallszahlen

Bernd Paul Jäger
Inst. f. Biometrie u. Med. Informatik
Ernst-Moritz-Arndt-Universität
Walther-Rathenau-Str. 48
17487 Greifswald
bjaeger@biometrie.uni-greifswald.de

Paul Eberhard Rudolph
Forschungsinstitut für die Biologie
landwirtschaftlicher Nutztiere (FBN)
Wilhelm-Stahl-Allee 2
18196 Dummerstorf
pe.rudolph@fbn-dummerstorf.de

Karl-Ernst Biebler
Inst. f. Biometrie u. Med. Informatik
Ernst-Moritz-Arndt-Universität
Walther-Rathenau-Str. 48
17487 Greifswald
biebler@biometrie.uni-greifswald.de

Armin Tuchscherer
Forschungsinstitut für die Biologie
landwirtschaftlicher Nutztiere (FBN)
Wilhelm-Stahl-Allee 2
18196 Dummerstorf
atuchsch@fbn-dummerstorf.de

Zusammenfassung

Mit der Entwicklung geeigneter Rechentechnik gewannen auch die Anwendungen sogenannter Monte-Carlo-Methoden schnell an Bedeutung. Wesentlicher Bestandteil einer Monte-Carlo-Methode ist die Erzeugung von Zufallszahlen einer interessierenden Verteilung. Die Erzeugung von Zufallszahlen einer bestimmten Verteilung erfolgt dabei fast immer über die vorhergehende Erzeugung gleichverteilter Zufallszahlen.
Bereits vom Pionier der Computertechnik, John von Neumann, gab es ein erstes Verfahren, das auf Rechnern realisiert wurde, die so genannte Methode der mittleren Ziffern von Quadraten, die allerdings denkbar schlecht ist und heute keine Rolle mehr spielt.
Die Mehrheit der heute - auch im Programmpaket SAS - verwendeten Zufallszahlengeneratoren sind multiplikative, additive oder gemischte Generatoren, die auf der Restklassenarithmetik ganzer Zahlen basieren.
Die Verwendung von Zufallszahlen in Verschlüsselungstechniken hat ebenfalls stark zur Entwicklung von Zufallszahlengeneratoren beigetragen.
Es wird in diesem Beitrag der Versuch unternommen, einige Entwicklungsrichtungen anzugeben. Insbesondere wird auf Generatoren eingegangen, die auf einem Satz von Weyl basieren. Für derartige Generatoren gilt theoretisch, dass die mit ihnen erzielten Zufallszahlen im betrachteten Intervall dicht liegen und damit Folgen von Zufallszahlen nicht zyklisch werden.

Schlüsselwörter: Zufallszahlengenerator, Gleichverteilung, Satz von Weyl

1 Einleitung

Heute steht dem Programmierer eine Vielzahl von Pseudozufallszahlengeneratoren zur Erzeugung gleichverteilter Zufallszahlen zur Verfügung, deren exaktes Arbeiten mit einer großen Anzahl statistischer Tests überprüft werden kann.

Den Beginn der Erzeugung von Zufallsgeneratoren stellt sicher die Quadrat-Mitten-Methode (mid-square-method) nach John von Neumann dar, die für die ENIAC, einen der ersten Computer der Welt, implementiert wurde.
Die meisten Generatoren beruhen auf der Restklassenarithmetik ganzer Zahlen. Kurz eingegangen wird auf die folgenden Zufallszahlengeneratoren:

- Multiplikative Generatoren, z.B. von Coveyou und MacPherson [3]
- Additiver Zufallszahlengenerator (Fibonacci-Generator)
- Gemischte Generatoren, z.B. nach Knuth [5]
- Quadratischer Generator von Blum-Blum-Shub [1], [2]
- Der Mersenne-Twister von Matsumoto/Nishimura [6], der beim Zufallszahlengenerator RAND(.) in SAS realisiert ist, ist ein moderner und zu Recht weit verbreiteter Generator mit sehr guten Eigenschaften. Für viele Programmiersprachen findet man im Internet Quellprogramme für den Mersenne-Twister.

Die beiden nicht zu behebenden Nachteile von Zufallszahlengeneratoren, die auf der Restklassenarithmetik beruhen, sind

- die Periodizität (auch wenn die Periodenlänge sehr groß ist) und
- die Eigenschaft, dass es einen kleinsten, nicht zu unterbietenden Abstand zwischen zwei erzeugten Zufallszahlen gibt, nämlich 1/M, wobei M die Zahl ist, bezüglich der die Restklassen bestimmt werden.

Irrationale Zahlen sind Dezimalzahlen mit unendlich vielen Nachkommastellen ohne Periode. Sie können als Grundlage zahlreicher Zufallszahlengeneratoren dienen. Die historisch sicher erste bewiesene Methode beruht auf einem Satz von Weyl [11] über Zahlenfolgen, die gleichverteilt modulo 1 sind. Auf einen auf dieser Methode beruhenden Zufallszahlengenerator wird besonders eingegangen. Er liefert theoretisch nichtperiodische Zufallszahlen, die darüber hinaus im Intervall [0, 1) dicht liegen. Wem an diesen Eigenschaften gelegen ist, wird den notwendigen Programmieraufwand in Kauf nehmen, um diesen Zufallsgenerator nach Weyl zu implementieren.

2 Der Quadrat-Mitten-Generator (mid-square-method) nach John von Neumann

John von Neumann (* 28. Dezember 1903, Budapest; † 8. Februar 1957 in Washington, (DC)) gilt als ein Pionier der Computertechnik. Neben seinen herausragenden Leistungen in der Mathematik und Computer-Architektur gehen auf John von Neumann aber auch wichtige Beiträge zur Physik (Quantenmechanik), zur Meteorologie (Strömungsdynamik) und den Wirtschaftswissenschaften (Spieltheorie) zurück.
Schon in der Anfangszeit der Programmierung war Bedarf an der Erzeugung von Zufallszahlen vorhanden. Man hoffte, dass bei der Multiplikation von vierstelligen ganzen Zahlen, deren Produkt eine maximal achtstellige ganze Zahl ergibt, die dritte bis sechste

Ziffer von rechts (vom Einer aus) gezählt eine nicht vorhersagbare neue maximal vierstellige Zahl und damit zufällig wäre. Für die Startzahl 2008, die man „seed“ (engl. Saat) nennt, wird im folgenden Rechenbeispiel die Folge x_n der vierstelligen Zufallszahlen erzeugt. Die fett markierten Ziffern der dritten bis sechsten Stelle (von rechts beginnend) ergeben die folgende generierte Zufallszahl.

$x_1 = 2008$	$x_1^2 = 4\mathbf{0320}64$	$x_6 = 4056$	$x_6^2 = 16\mathbf{4511}36$
$x_2 = 320$	$x_2^2 = \mathbf{1024}00$	$x_7 = 4511$	$x_7^2 = 20\mathbf{3491}21$
$x_3 = 1024$	$x_3^2 = 1\mathbf{0485}76$	$x_8 = 3491$	$x_8^2 = 12\mathbf{1870}81$
$x_4 = 485$	$x_4^2 = \mathbf{2352}25$	$x_9 = 1870$	$x_9^2 = 3\mathbf{4969}00$
$x_5 = 2325$	$x_5^2 = 5\mathbf{4056}25$	$x_{10} = 4969$	…….

Dieser Zufallszahlengenerator ist offensichtlich zyklisch. Immer dann, wenn die neu generierte Zahl x_n mit einer der Zahlen x_i übereinstimmt, i = 1, …, n-1, kann keine neue Zufallszahl entstehen, weil $x_{n+1} = x_{i+1}$, $x_{n+2} = x_{i+2}$, ….
Man hoffte, dass die Zyklen möglichst lang würden. Leider sind lange Zyklen eher die Ausnahme. Die folgende Abbildung 1 gibt die Zyklenlänge z in Abhängigkeit von der seed-Zahl an, wobei exemplarisch 0 < seed < 2000 gewählt wurde. Die Ergebnisse für alle weiteren Startzahlen sind ähnlich.

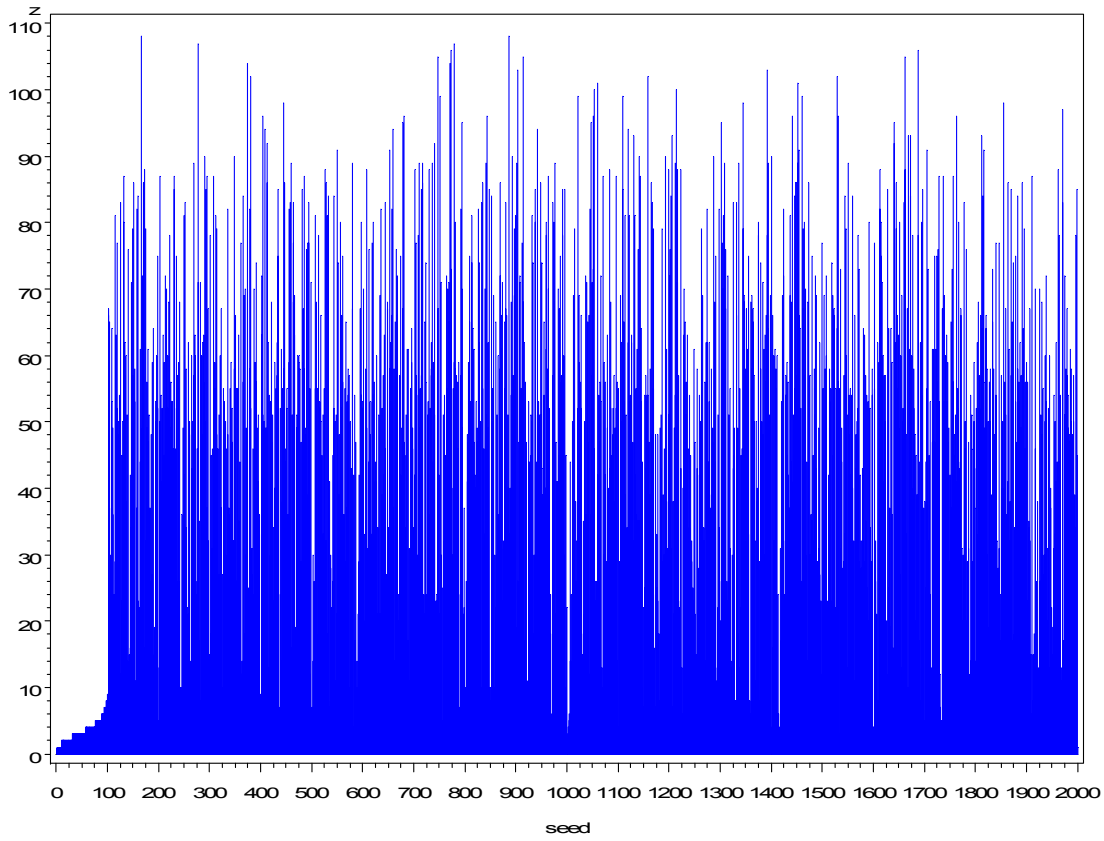

Abbildung 1: Zyklenlängen z bei der Quadrat-Mitten-Methode nach von Neumann in Abhängigkeit von der Startzahl (seed).

Man erkennt deutlich, dass die Zyklenlängen kleiner als 110 sind, dass sehr kleine Zyklenlängen vorkommen und große Zyklenlängen eher die Ausnahme darstellen.
In der folgenden Abbildung 2 sind für jede Zyklenlänge die relativen Häufigkeiten angetragen. Die mittlere Zykluslänge liegt zwischen 40 und 45.

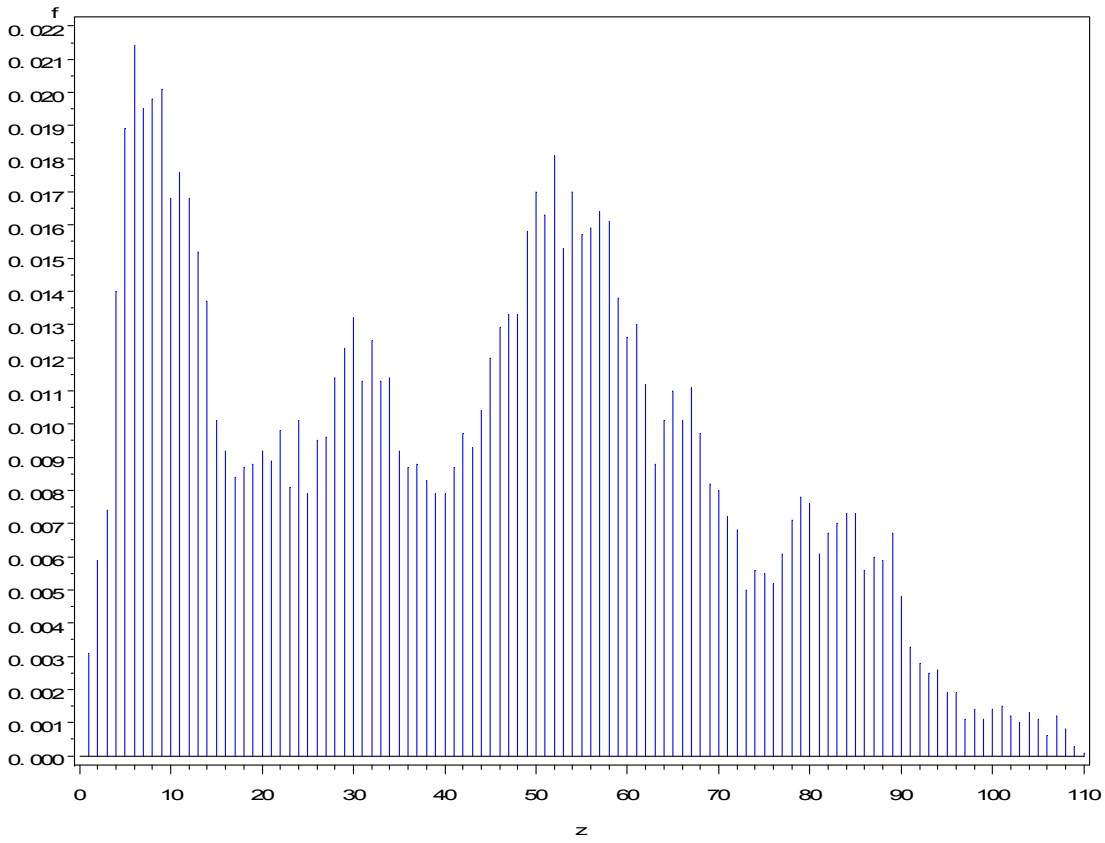

Abbildung 2: Häufigkeitsfunktion f der Längen z der Zufallszahlenfolgen, erzeugt nach der Quadrat-Mitten-Methode nach von Neumann bis zum Zyklischwerden.

Der nach der Quadrat-Mitten-Methode nach von Neumann arbeitende Pseudozufallszahlengenerator besitzt denkbar schlechte Eigenschaften. Er ist nur von historischem Interesse und wird heute nicht mehr angewandt. Auch seine Verallgemeinerung, die Verwendung sechs-, acht- oder gar zehnstelliger Zahlen anstelle der vierstelligen ganzen Zahlen, bringt keine wesentliche Verbesserung der oben angezeigten Probleme.

3 Generatoren, die auf der Restklassenarithmetik ganzer Zahlen beruhen

3.1 Multiplikative Generatoren

Die Arbeitsweise eines so genannten multiplikativen Zufallszahlengenerators versteht man am besten an einem Beispiel:

Als Parameter des multiplikativen Generators werden a = 7 und M = 11 festgelegt. Ausgehend von einer Startzahl $x_0 = 3$ (seed) erhält man die Folge der vom Rechner erzeugten Pseudozufallszahlen durch die Iteration

$$\mathbf{x_{n+1} = MOD(a \cdot x_n , M).}$$

Mittels $\mathbf{y_n = x_n / M}$ transformiert man die ganze Zahl x_{n+1} in das Intervall [0,1). Das gilt auch für die weiteren besprochenen Generatoren.

Man erhält für das Beispiel

$x_0 = 3$	$y_0 = 3/11 = 0.2727$
$x_1 = MOD(7 \cdot 3 , 11) = 10$	$y_1 = 10/11 = 0.9091$
$x_2 = MOD(7 \cdot 10 , 11) = 4$	$y_2 = 4/11 = 0.3636$
$x_3 = MOD(7 \cdot 4 , 11) = 6$	$y_3 = 6/11 = 0.5455$
$x_4 = MOD(7 \cdot 6 , 11) = 9$	$y_4 = 9/11 = 0.8182$
$x_5 = MOD(7 \cdot 9 , 11) = 8$	$y_5 = 8/11 = 0.7273$
$x_6 = MOD(7 \cdot 8 , 11) = 1$	$y_6 = 1/11 = 0.0909$
$x_7 = MOD(7 \cdot 1 , 11) = 7$	$y_7 = 7/11 = 0.6363$
$x_8 = MOD(7 \cdot 7 , 11) = 5$	$y_8 = 5/11 = 0.4545$
$x_9 = MOD(7 \cdot 5 , 11) = 2$	$y_9 = 2/11 = 0.1819$
$x_{10} = MOD(7 \cdot 2 , 11) = 3$	$y_{10} = 3/11 = 0.2727$

Man bemerkt, dass $x_{10} = 3$ mit dem Startwert $x_0 = 3$ übereinstimmt und die weiteren Zahlen der Iterationsfolge sich in der gleichen Reihenfolge wiederholen. Man sagt, der Zufallszahlengenerator ist zyklisch geworden.
Ein Zufallszahlengenerator bringt also nichts anderes zu Stande als die Restklassen von 0 bis M-1 gehörig zu durchmischen und diese Durchmischung von verschiedenen Startwerten aus zyklisch zu durchlaufen. Da bei der Division durch M nur die Reste von 0 bis M-1 vorkommen können, muss ein multiplikativer Zufallszahlengenerator spätestens nach M Iterationsschritten zyklisch geworden sein. Durch sehr große Zahlen M lässt sich das Zyklischwerden hinausschieben, nicht aber verhindern.

Bemerkungen

- Ein multiplikativer Zufallszahlengenerator bringt nichts anderes zu Stande als die Restklassen von 0 bis M-1 gehörig zu durchmischen und diese Durchmischung von verschiedenen Startwerten aus zyklisch zu durchlaufen. Dadurch ist die Gleichverteilung offensichtlich.
- Es gibt Sätze über die Parameter multiplikativer Generatoren, die möglichst große Zyklen sichern.
- Ein Beispiel für einen guten multiplikativen Zufallszahlengenerator ist der nach Coveyou und MacPherson: $\mathbf{x_{n+1} = MOD(a \cdot x_n , M)}$ mit $a = 3^{17}$, $M = 10^{10}$.

3.2 Additive Zufallszahlengeneratoren

Einen etwas anderen Weg gehen additive Zufallszahlengeneratoren, um größere Zyklenlängen zu erreichen. Beim so genannten Fibonacci-Generator wird die Iteration durch

$$\mathbf{x_{n+2} = MOD(x_{n+1} + x_n, M)}$$

beschrieben, wobei zwei Startzahlen x_0 und x_1 benötigt werden.
Als Beispiel seien M = 13 als Parameter des Fibonacci-Generators und x_0 =1 und x_1 = 1 als Startwerte festgelegt. Die folgende Tabelle 1 enthält die Iterationen, bis erstmals alle Zahlen mit den Startzahlen übereinstimmen. Ab hier beginnt der Zyklus erneut.

Tabelle 1: Berechnung der Fibonacci-Zufallszahlen für M = 13 und die Startwerte x_0 =1 und x_1 = 1 bis zum Zyklischwerden (Länge des Zyklus 28)

Iteration	x_{n+2}	x_{n+1}	x_n	Iteration	x_{n+2}	x_{n+1}	x_n
1	2	1	1	16	10	11	12
2	3	2	1	17	8	10	11
3	5	3	2	18	5	8	10
4	8	5	3	19	0	5	8
5	0	8	5	20	5	0	5
6	8	0	8	21	5	5	0
7	8	8	0	22	10	5	5
8	3	8	8	23	2	10	5
9	11	3	8	24	12	2	10
10	1	11	3	25	1	12	2
11	12	1	11	26	0	1	12
12	0	12	1	27	1	0	1
13	12	0	12	28	1	1	0
14	12	12	0	29	2	1	1
15	11	12	12				

Obwohl nur 13 Restklassen möglich sind, wiederholen sich die Zufallszahlen erst ab der 29. Iteration. In der 6. Iteration kommt zum zweiten Mal eine 8. Während die erste 8 in der 4. Iteration von einer 0 gefolgt wird, ist die Folgezahl der 8 in der 8. Iteration wiederum eine 8. Der Zyklus wird erst beendet, wenn die Startwerte x_1 = 1 und x_0 = 1 als Kombination erneut auftreten. Das ist in der 29. Zeile der Fall. Die Zykluslänge ist 28. Man erkennt, dass die Periodenlänge bei Fibonacci-Generatoren größer als M werden kann.

Bemerkungen

- Es gibt Sätze über die Periodenlänge von Zufallsgeneratoren. Für $M = 2^n$ beträgt die Periode eines Fibonacci-Generators unabhängig von der Wahl der Startwerte x_0 und x_1 stets $3 \cdot 2^{n-1}$.
- Beispiel für einen guten Fibonacci-Generator ist $\mathbf{x_{n+2} = MOD(x_{n+1} + x_n, M)}$ mit dem Parameter $M = 2^{35}$.

3.3 Gemischte Generatoren

Die gemischten Zufallszahlengeneratoren verbinden die guten Eigenschaften von additiven und multiplikativen Generatoren. Die Iteration wird realisiert durch

$$\mathbf{x_{n+1} = MOD(ax_n + b , M),}$$

einer Multiplikation des Startwertes x_n mit a und einer zusätzlichen Addition einer ganzen Zahl b, wobei a, b und x_n ganze Zahlen aus [0,M) sind.
Ein gutes Beispiel für einen gemischten Zufallszahlengenerator ist der nach Knuth [5] mit den Parametern $a = 5^{13}, b = 1, x_0 = 37$ und $M = 2^{35}$.

3.4 Quadratischer Blum-Blum-Shub-Zufallszahlengenerator

Der Modulo-Parameter M des Blum-Blum-Shub-Generators [2] ist das Produkt $M = p \cdot q$ zweier sehr großer Primzahlen p und q mit jeweils ca. 100 Stellen. Die Bedingungen 3 = MOD(p,4) und 3 = MOD(q,4) sorgen für große Zyklenlängen. Die Iteration erfolgt nach

$$\mathbf{x_{n+1} = MOD((x_n)^2 , M).}$$

Die Iterationszufallszahlenfolge des Blum-Blum-Shub-Generators wird weniger zu Simulationszwecken als zu modernen Codierungsverfahren benutzt. Die statistischen Eigenschaften sind weniger von Interesse.

3.5 Der Mersenne-Twister

Verallgemeinerungen der bisher aufgezeigten Iterationen der Zufallszahlengeneratoren sind durch Ausdehnung auf mehr als zwei Startwerte und mehr als zwei Summanden in der Iterationsgleichung möglich.
Der Mersenne-Twister von Matsumoto, M. und Nishimura, T. [6], der gegenwärtig am häufigsten für Simulationszwecke verwendete Zufallszahlengenerator, hat Startwerte von x_1 bis x_{624}. Er kann aus dem Internet als Quelltext für die meisten gängigen Programmiersprachen kostenlos bezogen werden. Seine Periodenlänge ist für praktische Belange mehr als ausreichend.

Bemerkungen

- Er besitzt eine extrem lange Periode von $2^{19937}-1$ ($\approx 4{,}3\cdot 10^{6001}$). Die Periodenlänge ist eine Mersenne-Primzahl (Nachweis 1971 von Tuckerman).
- In der SAS-Funktion Rand(.) ist der Mersenne-Twister realisiert.
- Der besondere Vorzug besteht darin, dass er für mehrdimensional gleichverteilte Zufallszahlen geeignet ist. Der Nachweis ist bis zur Dimension 623 geführt.

4 Zufallszahlengeneratoren, die auf einem Satz von Weyl basieren

Definition 1 (Bruchteilfolge)
Sei $x_1, x_2, \ldots$ eine Folge reeller Zahlen. Die zugehörige Folge b_1, b_2, … mit $b_i = x_i - [x_i]$ für alle i, wobei [z] die größte ganze Zahl kleiner als z bedeutet (SAS-Funktion FLOOR), heißt Bruchteilfolge.

Für die Folgenglieder der Bruchteilfolge jeder reellen Zahlenfolge gilt stets $0 \le b_i \le 1$.
Die Folgenglieder der Bruchteilfolge werden bei bestimmten Folgen, den so genannten gleichverteilten modulo 1, als Zufallszahlenfolge aufgefasst. Ihre Gleichverteilung wird in der Definition gefordert.

Definition 2 (gleichverteilte Folge modulo 1)
Die Folge $x_1, x_{2,} \ldots$ heißt gleichverteilt modulo 1, wenn für jedes Intervall $[a,b) \subset [0,1)$ die relative Anzahl der Folgenglieder b_i ($i \le N$) aus diesem Intervall gegen die Länge des Intervalls strebt, d.h. wenn $A\left([a,b),N\right) := \#\left\{n \le N \mid \{b_n\} \in [a,b)\right\}$ die Anzahl der Bruchteilfolgenglieder ist mit einem Index kleiner als N, so ist der Grenzwert

$$\lim_{N\to\infty} \frac{A\left([a,b),N\right)}{N} = b - a\,.$$

Ob überhaupt Folgen existieren, die gleichverteilt modulo 1 sind, weiß man zu diesem Zeitpunkt nicht. Ein Satz von Weyl [11] charakterisiert aber solche Folgen, die gleichverteilt modulo 1 sind. Der Nachweis ist aufwändig und soll hier nicht geführt werden. Es sollen nur einige Beispiele für Folgen genannt werden, die gleichverteilt modulo 1 sind. Man weiß beispielsweise, dass die reellen Zahlenfolgen

- $(n\cdot\alpha)_{n\ge 1}$, wenn α irrational,
- $\left(n^{\alpha}\log_{\beta} n\right)_{n\ge 1}$, wenn $0 < \alpha < 1$ und β eine beliebige reelle Zahl und

- $\left(P(n)\right)_{n\geq 1}$, wenn P(n) ein Polynom in n vom Grade k≥1 bezeichnet, welches mindestens einen irrationalen Koeffizienten besitzt,

gleichverteilt modulo 1 sind. Das dritte Beispiel schließt das erste ein, denn $n \cdot \alpha$ ist ein Polynom vom Grade 1 in n mit einem irrationalen Koeffizienten α. Mit den folgenden Argumentationsschritten wird erschlossen, dass die Folgenglieder der Bruchteilfolge dicht im Intervall [0, 1) liegen:

- Wenn eine Folge gleichverteilt modulo 1 ist, muss in jedem Intervall $[a,b) \subset [0,1)$ laut Definition asymptotisch etwa der Anteil $N\cdot(b-a)$ der Glieder der Bruchteilfolge liegen.
- Wenn eine Folge gleichverteilt modulo 1 ist, muss daher jedes Intervall $[a,b)\subset[0,1)$ unendlich viele Glieder der Bruchteilfolge enthalten.
- Zwei Folgenglieder der Bruchteilfolge bilden o.B.d.A. ebenfalls ein solches Intervall $[b_i, b_{i+1}) \subset [0,1)$. In diesem müssen dann ebenfalls unendlich viele weitere Folgenglieder liegen, d.h. die Bruchteilfolge liegt dicht im Intervall [0,1). (Das ist der so genannte Approximationssatz von Kronecker.)

Beispiel:
Das Polynom $P(n) = \sqrt{5} \cdot n + \sqrt{17} \cdot n^2$ in n vom Grade 2 mit zwei irrationalen Koeffizienten definiert eine Folge, die gleichverteilt modulo 1 ist.
SAS-Programm zur Erzeugung von gleichverteilten Zufallszahlen nach dem Satz von Weyl:

```
data weyl;
format z1 z 20.18;
seed=7;
/* Startwert Zufallszahlengenerators zwischen 0 und 1000 wählen */
sim=10000;/* Simulationsumfang */
do n=seed to sim+seed;
z1=SQRT(5)*n+SQRT(17)*n**2;  /* Folge */
z=z1-FLOOR(z1);              /* Bruchfolge */
output;
end;
run;
```

Mit Hilfe dieses SAS-Programms wurden 10 000 Zufallszahlen generiert. Ihre empirische Verteilung ist in Abbildung 3 dargestellt. Die Treppenfunktion ist nicht mehr als solche erkennbar. Die exakte Verteilungsfunktion der Gleichverteilung auf dem Einheitsintervall ist die Gerade y = z. Unterschiede sind nicht darstellbar.

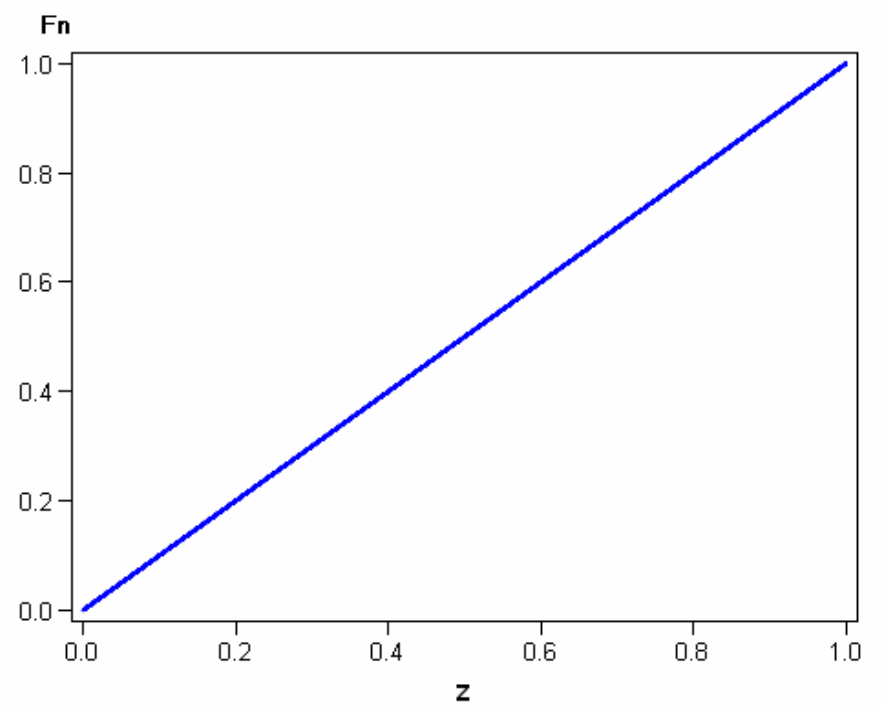

Abbildung 3: Empirische Verteilungsfunktion Fn für die nach dem Satz von Weyl mittels der gleichverteilten Folge $P(n) = \sqrt{5} \cdot n + \sqrt{17} \cdot n^2$ modulo 1 erzeugten gleichverteilten 10000 Pseudozufallszahlen z.

Literatur

[1] Blum,L.; Blum, M.; Shub, M. (1986): A Simple Unpredictable Pseudo-Random Number Generator. SIAM Journal on Computing, 15, 364–383.

[2] Blum,L.; Blum, M.; Shub, M. (2004): Comparison of Two Pseudo-Random Number Generators. Advances in Cryptology: Proceedings of Crypto '82.

[3] Coveyou, R. R.; MacPherson, R. D. (1967): Fourier Analysis of Uniform Random Number Generators. Journal of the ACM, 14,100-119.

[4] Jäger, B. P.; Philipp, T.; Rudolph, P. E.; Biebler, K.-E. (2008): Über Tests von Zufallszahlengeneratoren. in: Hilgers, R.-D.; Heussen, N.; Herff, W.; Ortseifen, C. (2008):KSFE 2008, Proceedings der 12. Konferenz der SAS-Anwender in Forschung und Entwicklung (KSFE), 105-123.

[5] Knuth, D.E. (1969): The Art of Computerprogramming. Volume 2, Addison-Wesley Publishing Company

[6] Matsumoto, M.; Nishimura, T. (1998): Mersenne Twister: A 623-dimensionally Equidistributed Uniform Pseudorandom Number Generator. ACM Trans. on Modeling and Computer Simulations, 8, 3-30

[7] Matsumoto, M.; Saito, M.; Haramoto, H.; Nishimura, T. (2006): Pseudorandom Number Generation: Impossibility and Compromise. Journal of Universal Computer Science, vol. 12, no. 6, 672-690

[8] Niederreiter, H.: Random Number Generationand Quasi-Monte Carlo Methods, SIAM,1992

[9] SAS Institute Inc. (2004). SAS/STAT 9.1 User's Guide. Cary, NC: SAS Institute Inc.

[10] SAS Institute Inc. (2004): SAS 9.1 Macro Language Reference. Cary, NC: SAS Institute Inc.

[11] Weyl, H. K. H. (1916): Über die Gleichverteilung der Zahlen mod 1. Math.Annalen, LXXVII, 313-352.