ETH zürich

Matlab

Zufall / Programmieren mit Matlab

Dr. Hermann Lehner

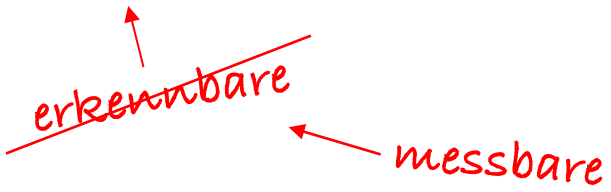
Departement Informatik, ETH Zürich

Zufall ist ein Wort ohne Sinn;
nichts kann ohne Ursache existieren ?

-- Voltaire

Was ist Zufall?

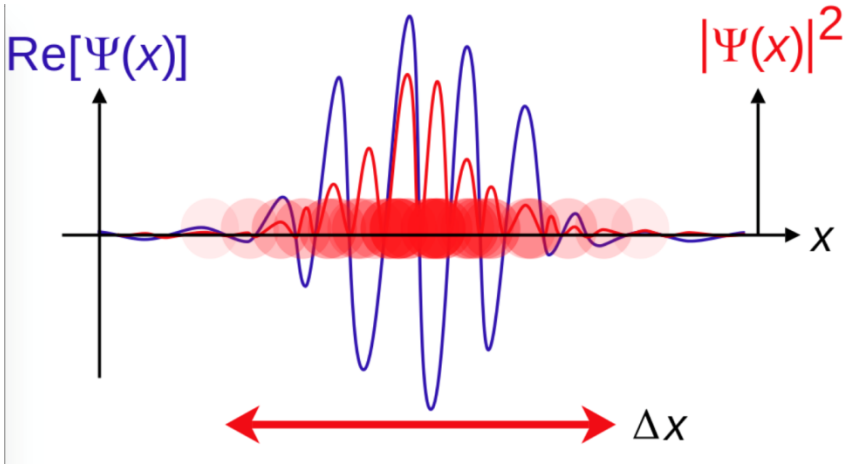
- Zufällige Ereignisse sind unabhängig voneinander
- Zufällige Ereignisse sind gleichmässig verteilt
- Zufällige Ereignisse haben keine Ursache



Quellen zufälliger Ereignisse

Chaotische Systeme

Quantenmechanische Effekte

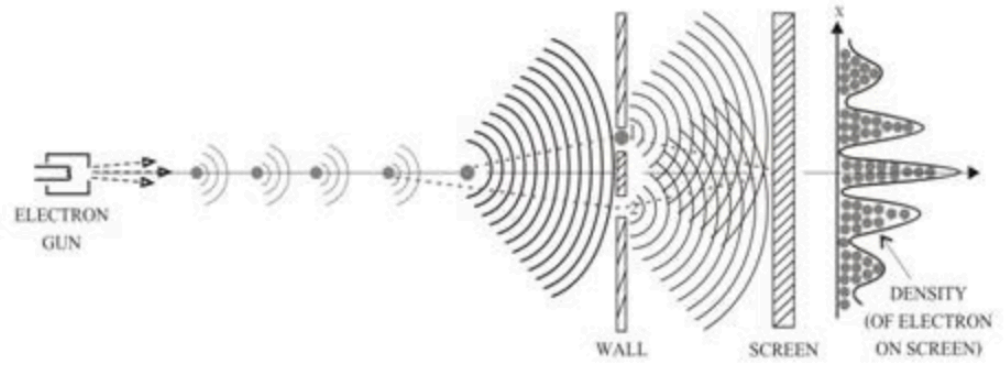


Chaotische Systeme

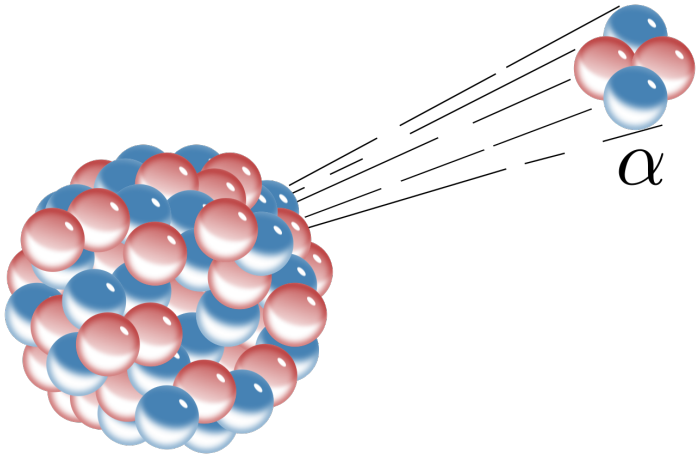
Lotto-Ziehung

Globales Wetter

Quantenmechanische Effekte



Teilchen-Welle Dualismus



Alpha-Zerfall von Atomkernen

Verwendung von Zufallsereignissen

Simulationen komplexer Vorgänge

- Physik (z.B. Molekulardynamik)
- Biochemie (z.B. Falten von Proteinen)
- Meteorologie (z.B. Wettervorhersage)
- Ökonomie (z.B. Aktienkurs-Vorhersagen)
- Politik (Wahl-Prognose Modelle)

Verschlüsselungsverfahren

- Erstellung sicherer Schlüssel
- Sichere Kommunikationsprotokolle

Mathematik

- Integration sehr komplexer Funktionen

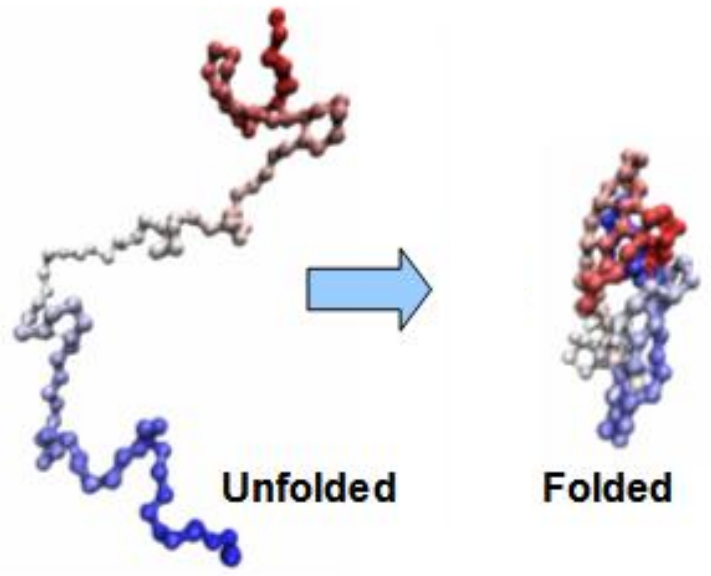
Software-Engineering

- Random testing
- Image rendering

Verwendung von Zufallsereignissen

Simulationen komplexer Vorgänge

- Physik (z.B. Molekulardynamik)
- **Biochemie (z.B. Falten von Proteinen)**
- Meteorologie (z.B. Wettervorhersage)
- Ökonomie (z.B. Aktienkurs-Vorhersagen)
- Politik (Wahl-Prognose Modelle)



Verschlüsselungsverfahren

- Erstellung sicherer Schlüssel
- Sichere Kommunikationsprotokolle

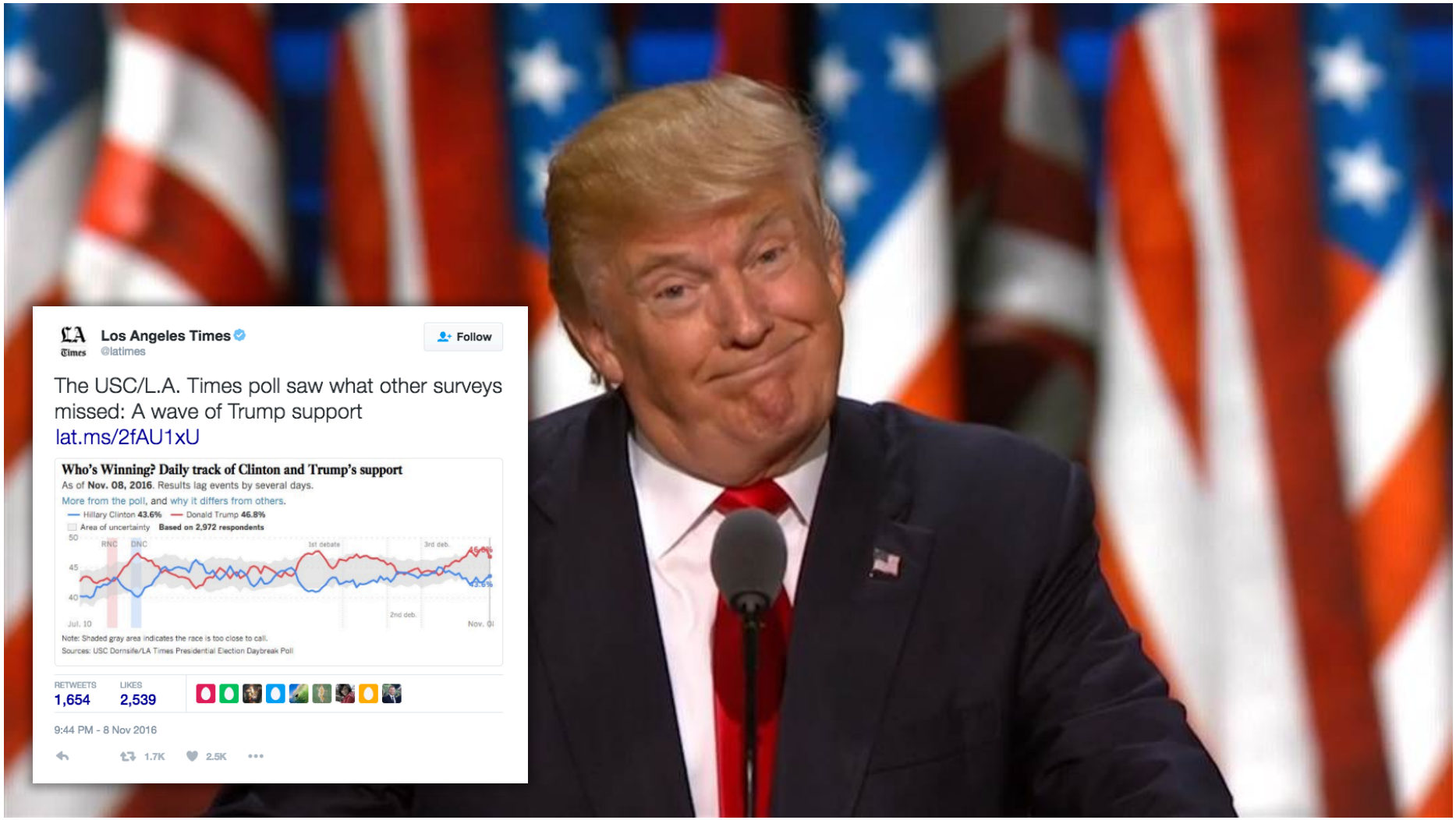
Mathematik

- Integration sehr komplexer Funktionen

Software-Engineering

- Random testing
- Image rendering

Verwendung von Zufallsereignissen



Verwendung von Zufallsereignissen

Simulationen komplexer Vorgänge

- Physik (z.B. Molekulardynamik)
- Biochemie (z.B. Falten von Proteinen)
- Meteorologie (z.B. Wettervorhersage)
- Ökonomie (z.B. Aktienkurs-Vorhersagen)
- Politik (Wahl-Prognose Modelle)

Verschlüsselungsverfahren

- Erstellung sicherer Schlüssel
- Sichere Kommunikationsprotokolle

Mathematik

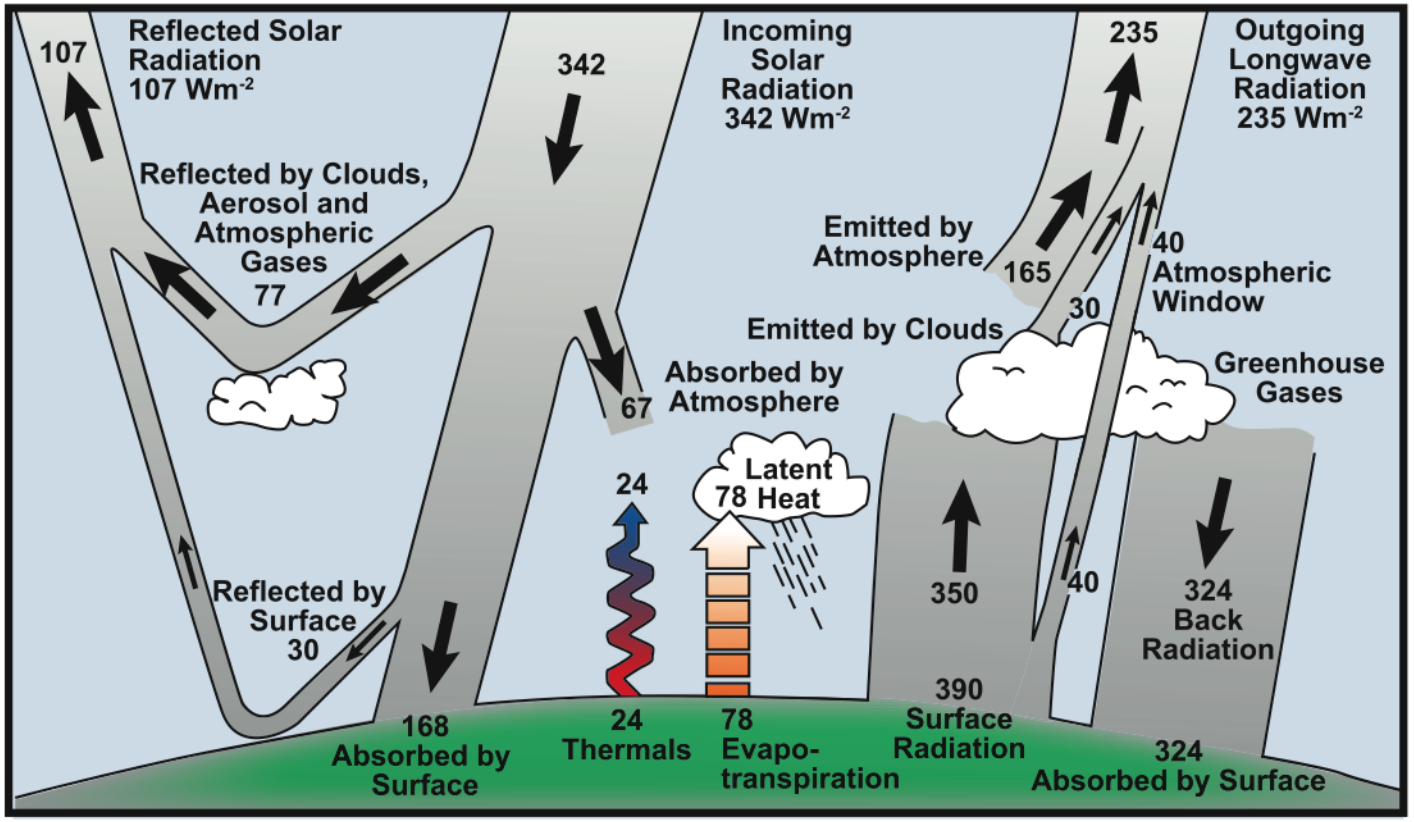
- Integration sehr komplexer Funktionen

Software-Engineering

- **Random testing**
- Image rendering

Simulation komplexer Vorgänge:

Beispiel: Energy Balance Model



Simulation komplexer Vorgänge:
Beispiel: Energy Balance Model

- Nicht analytisch zu lösen
- Simulation mit Hilfe von mathematischen Modellen
- Verbesserung der Modelle aufgrund der Simulation

Methode: Monte Carlo Simulation

$$T_i = \frac{\gamma_i (H_0/4)(1-\alpha_i) + CT_s - A + 5\varphi_i B}{B+C}$$

where:

γ_i is the ratio correction of zone to incoming radiation

H_0 is the solar constant

α_i is the surface albedo for the zone

C is the constant 3.8 watt-m$^{-2}C^{-1}$

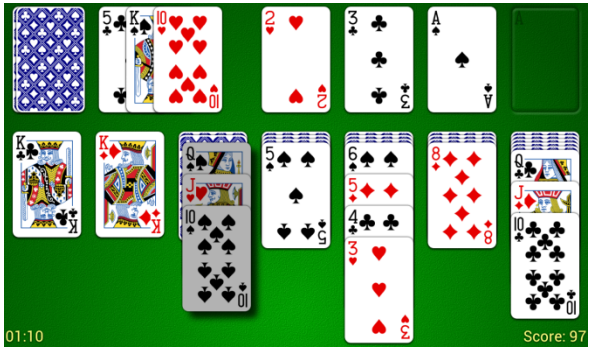
T_s is the average surface temperature

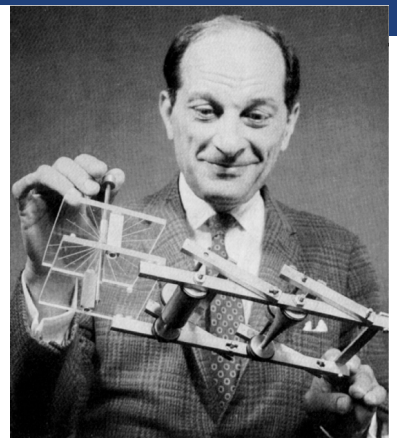
A is a constant 204 watt-m^{-2}

B is a constant 2.17watt-m^{-2}C^{-1}

φ_i is the fractional cloud cover in each zone

Monte Carlo Simulation





Stanislaw Ulam

- Was ist die Wahrscheinlichkeit dass ein lösbares Deck gespielt wird?

→ Zähle Anzahl erfolgreiche/nicht-erfolgreiche Spiele

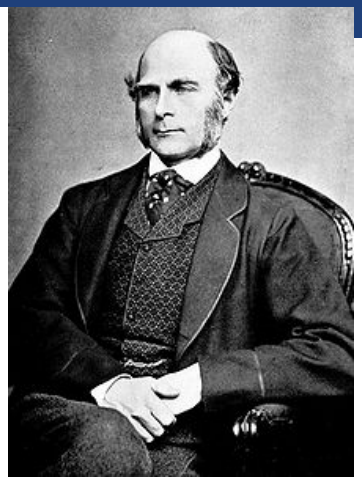
- Je mehr Experimente desto genauer das Resultat

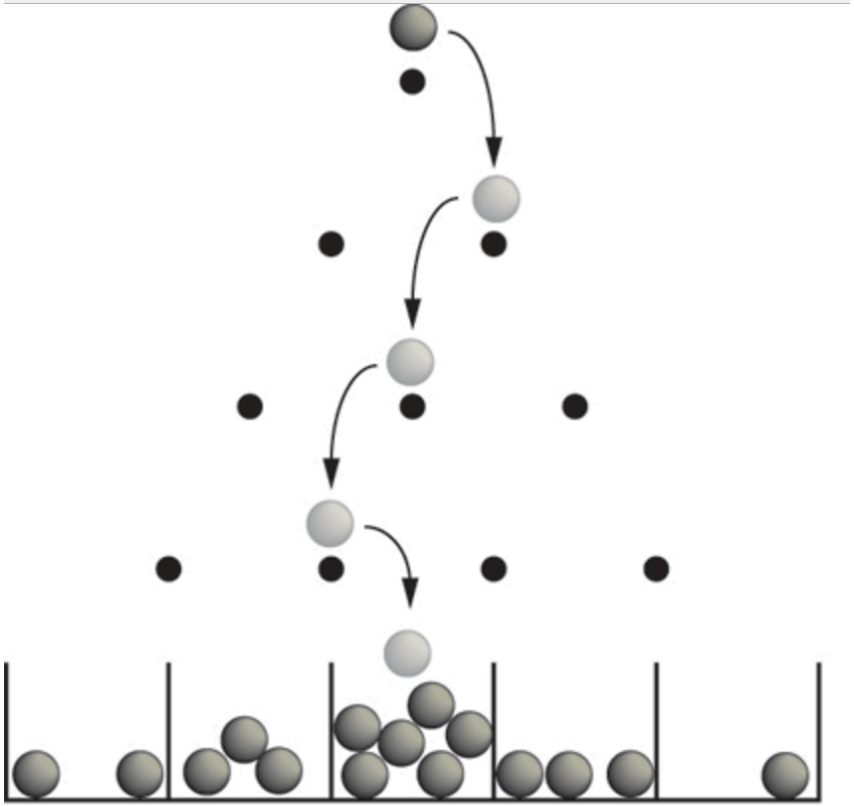
→ Das Gesetz der grossen Zahlen

Wir brauchen sehr, sehr ... sehr viele gute Zufallszahlen

Beispiel einer Monte Carlo Simulation

Das Galton-Brett

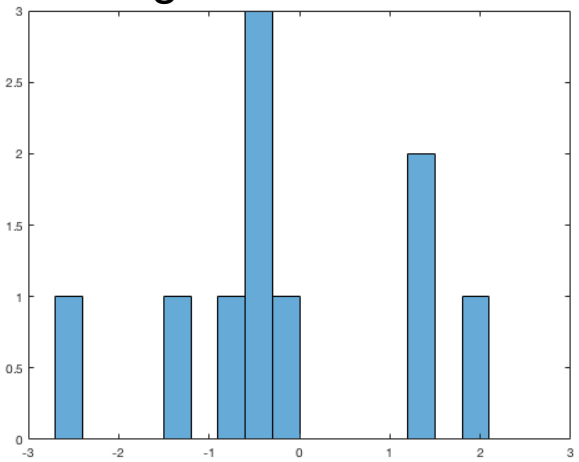




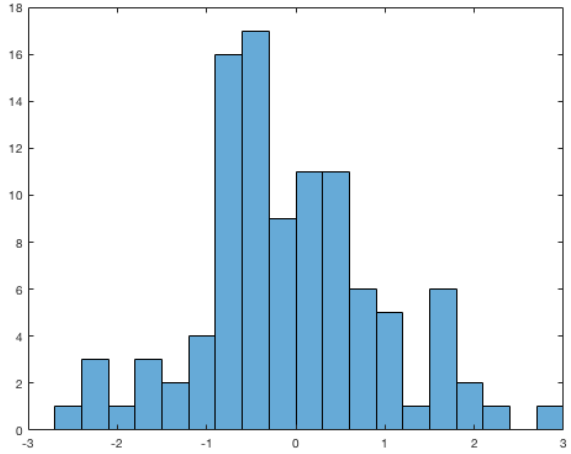
klassisches Galtonbrett

Galton Brett und das Gesetz der grossen Zahlen

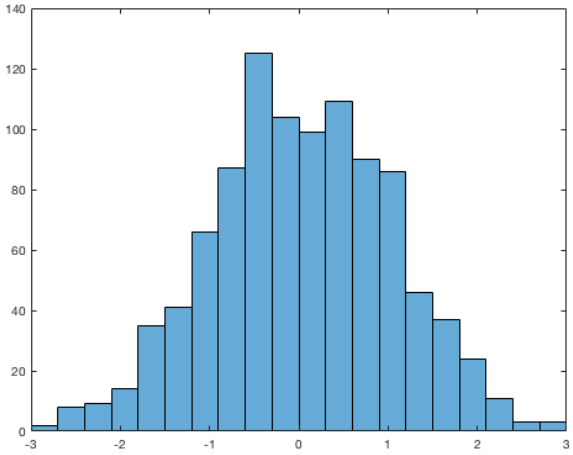
10 Kugeln



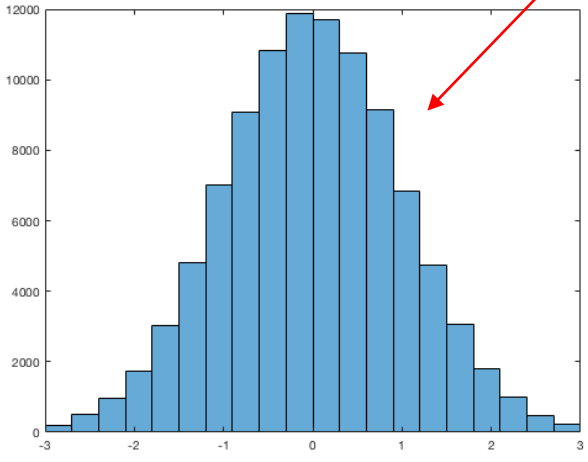
100 Kugeln



1'000 Kugeln



100'000 Kugeln

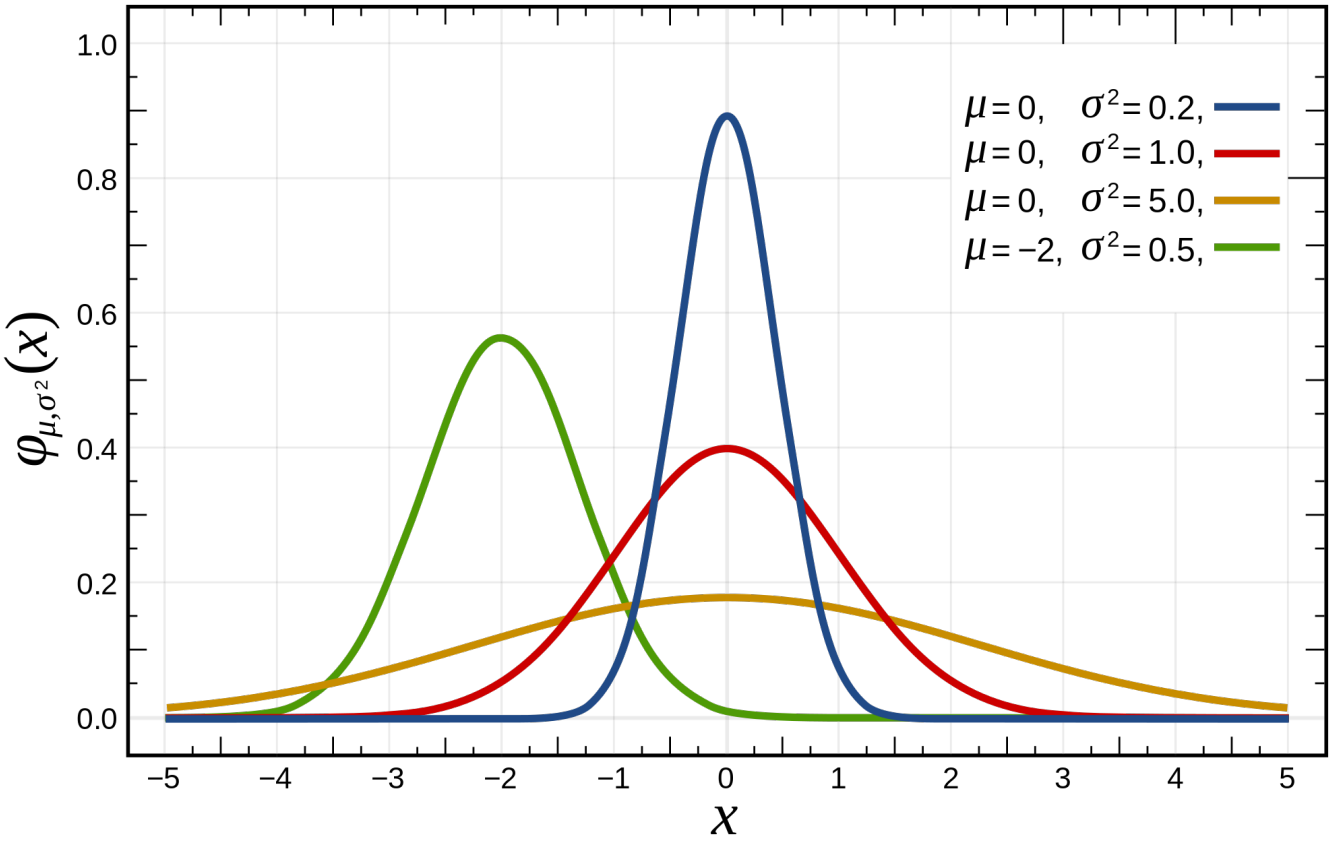


Normalverteilt?

Normalverteilung

$$\varphi_{\mu,\sigma^2}(x) = \frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

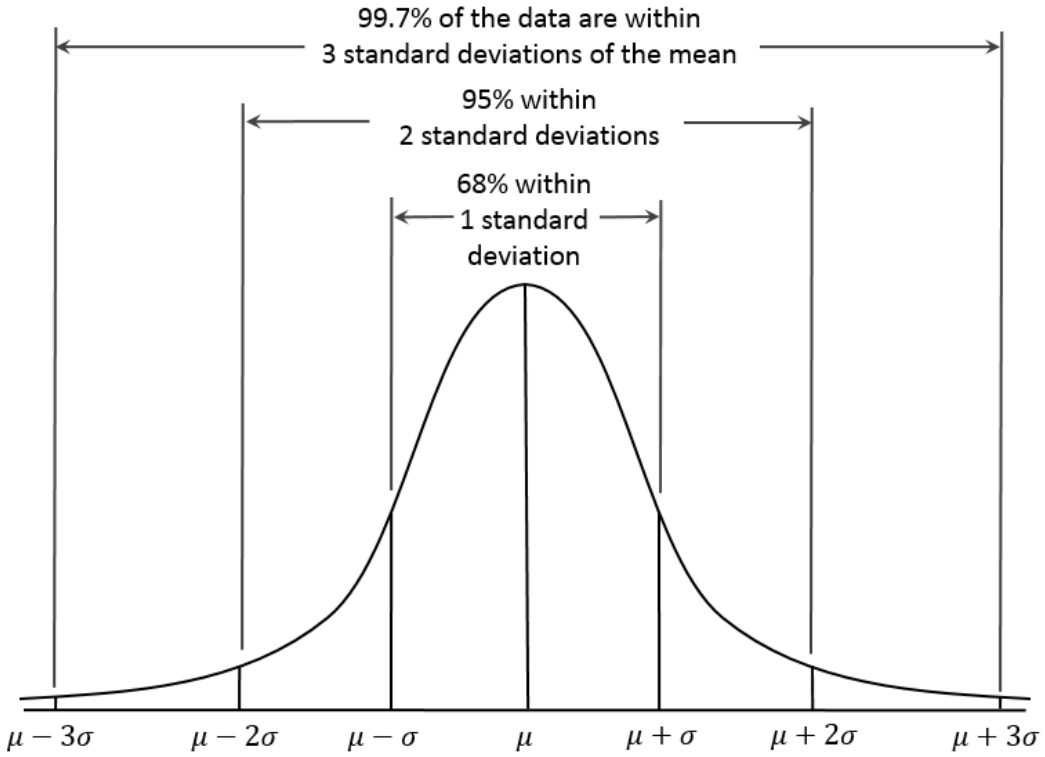
μ	Mittelwert
σ^2	Varianz
σ	Standardabweichung



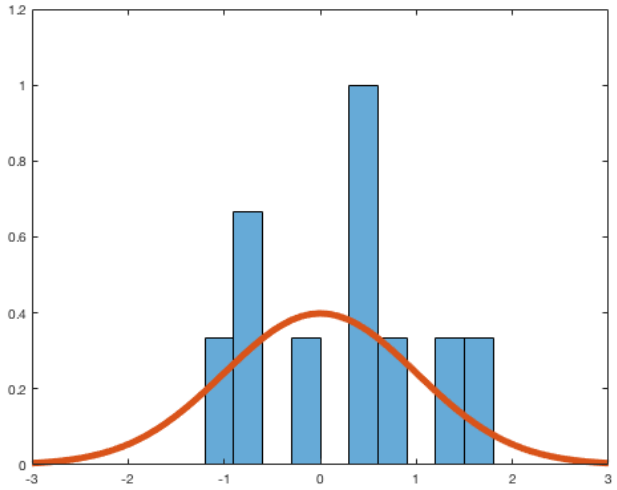
Normalverteilung

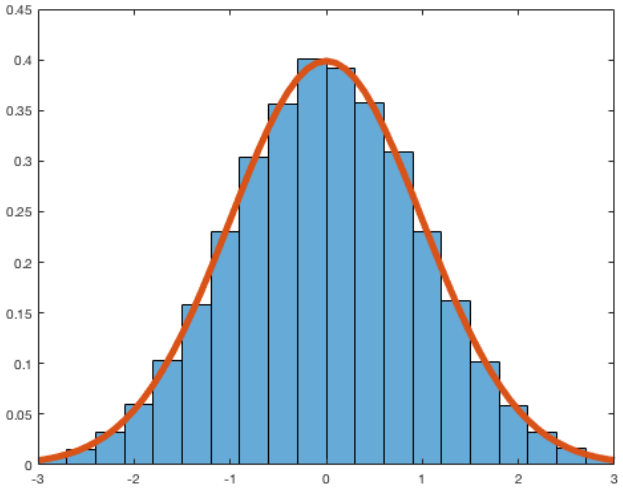
$$\varphi_{\mu,\sigma^2}(x) = \frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

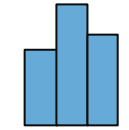
μ	Mittelwert
σ^2	Varianz
σ	Standardabweichung



Galton Brett und das Gesetz der grossen Zahlen



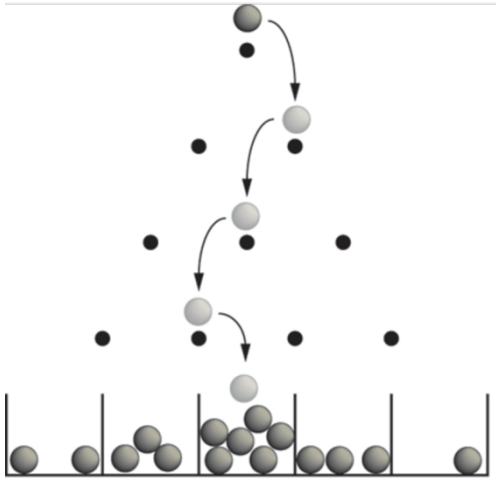




Relative Häufigkeit der Messwerte

Normalverteilung

Relative Häufigkeit:

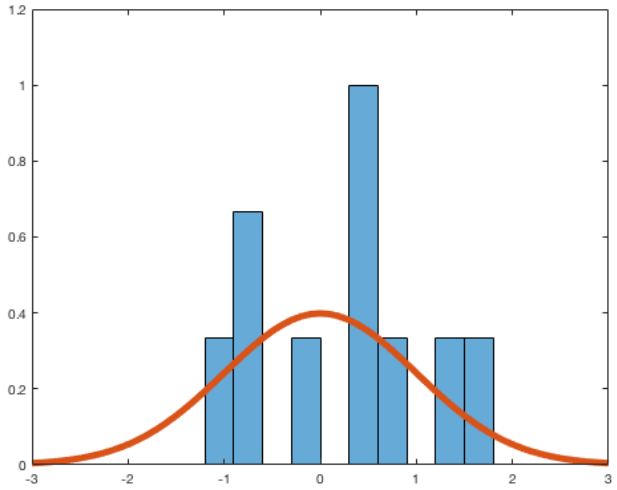


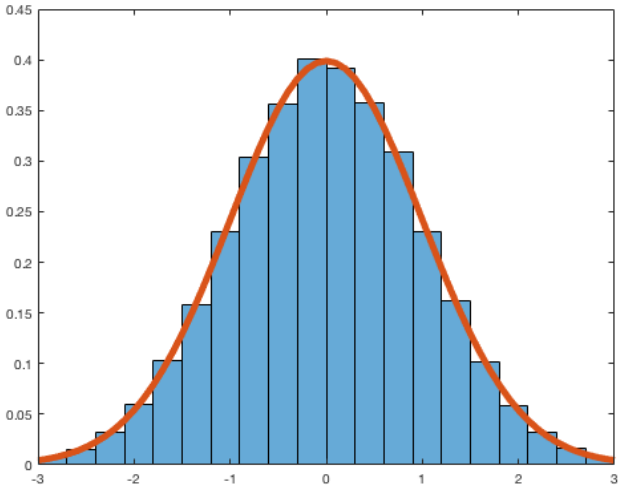
klassisches Galtonbrett

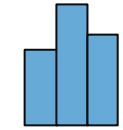
2/15 = 13%

6/15 = 40%

Galton Brett und das Gesetz der grossen Zahlen







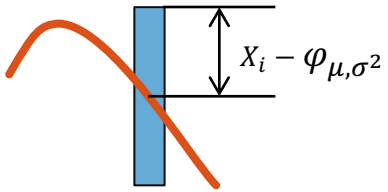
Relative Häufigkeit der Messwerte

Normalverteilung

Gesetz der grossen Zahlen:

*Mit zunehmender Anzahl unabhängiger Experimente unter gleichbleibenden Bedingungen **stabilisiert** sich die relative Häufigkeit der Messwerte um die theoretischen Wahrscheinlichkeit.*

$$\overline{X_n} = \frac{1}{n}\sum_{i=1}^{n}(X_i - \varphi_{\mu,\sigma^2})$$



$$\lim_{n\to\infty} P(|\overline{X_n}| > \varepsilon) = 0 \qquad \varepsilon > 0$$

z.B: 1‰

Echte Zufallszahlen

Generiert aufgrund Beobachtung echt zufälliger Ereignisse:

- Zeitabstände des Alpha-Zerfalls von Atomen
- Varianz im atmosphärisches Rauschen

- Rauschen von verpolten Dioden
- Zeitabstände zwischen Maus-Klicks
- Unregelmässigkeiten beim Tippen
- Zeitliche Abfolge ankommender Datenpackete aus dem Internet
- …

Vorteile:

- Unabhängige Zahlenreihen
- Keine Periodizität
- Selbst mit beliebigem Aufwand nicht vorhersehbar

Nachteile

- Langsam
- Teuer (je nachdem)
- Nicht reproduzierbar

Pseudo Zufallszahlen

Berechnet mittels einem deterministischen Algorithmus:

- Initialisiert mit einem Anfangswert (Seed)
- Jede Zufallszahl in der Reihe wird aufgrund der davor generierten Zufallszahl(en) berechnet

Vorteile:

- Schnell
- Erfordert kein Zugang zu externen Systemen
- Reproduzierbar

Nachteile

- Mit genügend Aufwand vorhersehbar (ungeeignet zum Verschlüsseln)
- Nicht immer genuegend Unabhaengig

Anwendungsbereiche von Zufallszahlen

	Echte Zufallszahlen	Pseudo-Zufallszahlen
Simulationen	✔ (zu langsam)	✔
Mathematische Applikationen	✔ (zu langsam)	✔
Software Engineering	✘ (nicht reproduzierbar)	✔
Verschlüsselung	✔	✘ (vorhersehbar)

Echt zufällig oder generiert?

8, 1, 7, 5, 15 ,15 ,10 ,6 ,12 ,11 ,12 ,2 ,14 ,14 ,4 , ...

Ja, echte Zufallszahlen

RANDOM.ORG

Search RANDOM.ORG
Google™ Custom Search Search

True Random Number Service

Do you own an iOS or Android device? Check out our app!

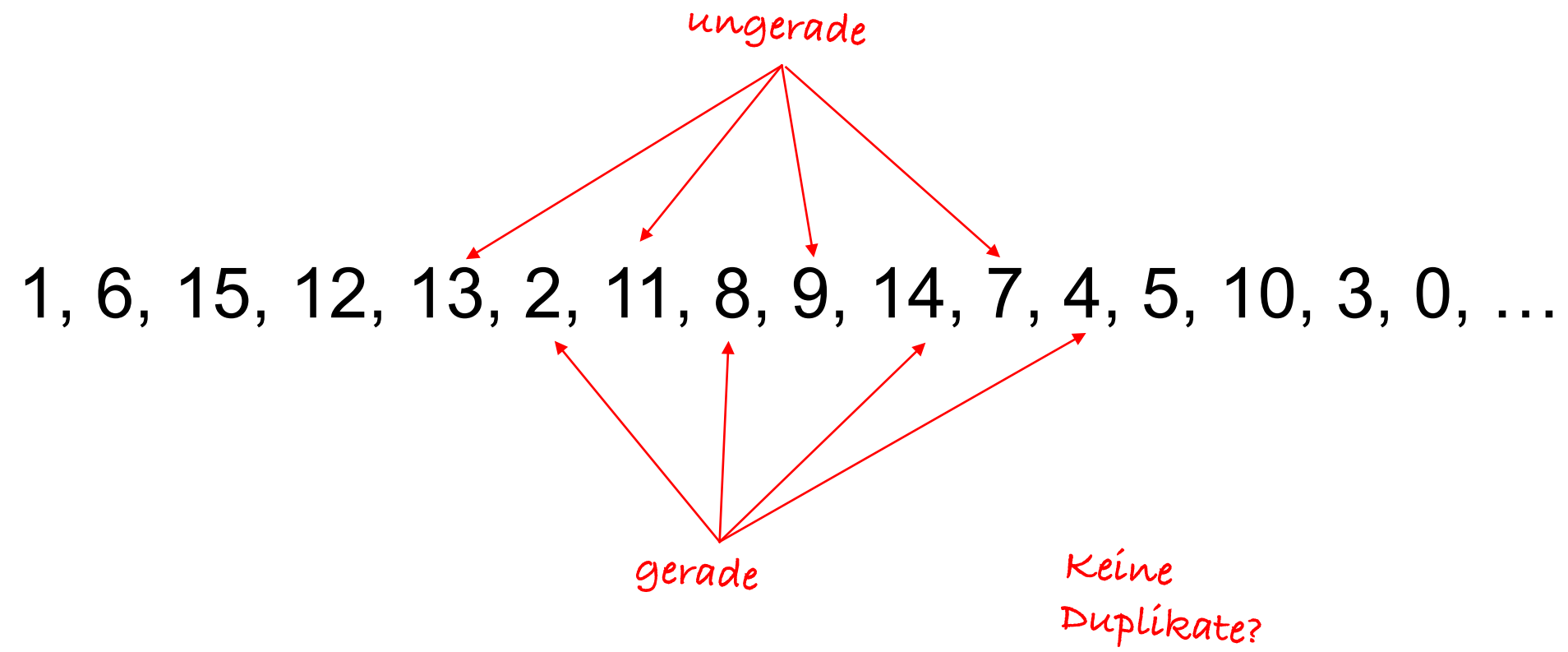
Random Integer Generator

Here are your random numbers:

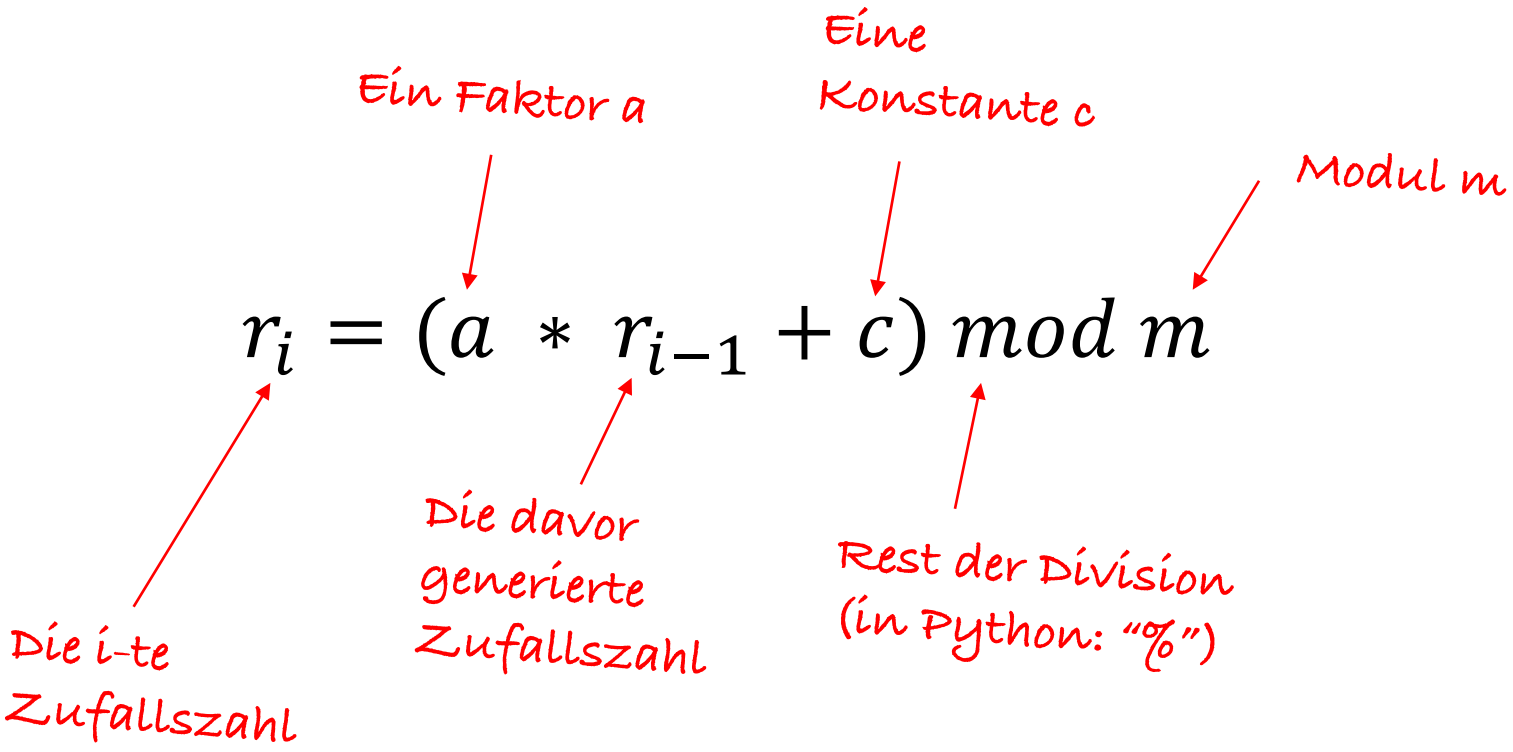
8
1
7
5
15
15
10
6
12
11
12
2
14
14
4

Timestamp: 2016-11-09 23:22:13 UTC

Echt zufällig oder generiert?



Der Lineare Kongruenzgenerator



Der Lineare Kongruenzgenerator

a c m

$$r_i = (\mathbf{5} * r_{i-1} + \mathbf{1}) \; mod \; \mathbf{16}$$

$$1 = (\mathbf{5} * 0 + \mathbf{1}) \; mod \; \mathbf{16}$$
$$6 = (\mathbf{5} * 1 + \mathbf{1}) \; mod \; \mathbf{16}$$
$$15 = (\mathbf{5} * 6 + \mathbf{1}) \; mod \; \mathbf{16}$$
$$12 = (\mathbf{5} * 15 + \mathbf{1}) \; mod \; \mathbf{16}$$

1, 6, 15, 12, 13, 2, 11, 8, 9, 14, 7, 4, 5, 10, 3, 0, 1, 6, 15, ...

Ab hier wiederholt sich die Reihe

Der Lineare Kongruenzgenerator

$$r_i = (\mathbf{65539} * r_{i-1} + \mathbf{0}) \, mod \, \mathbf{2^{32}}$$

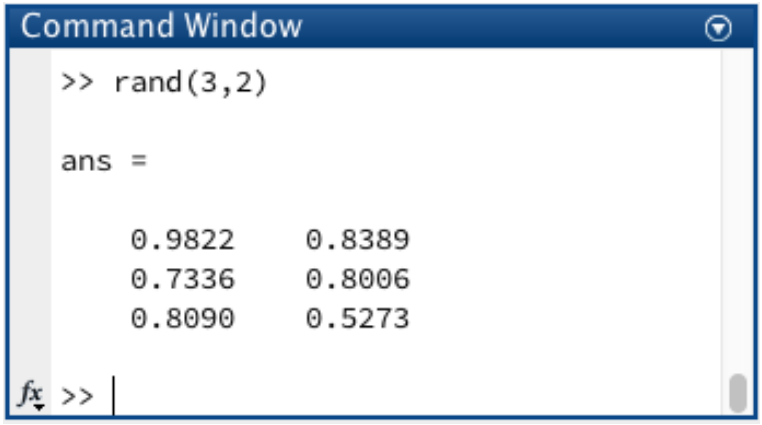
“RANDU”

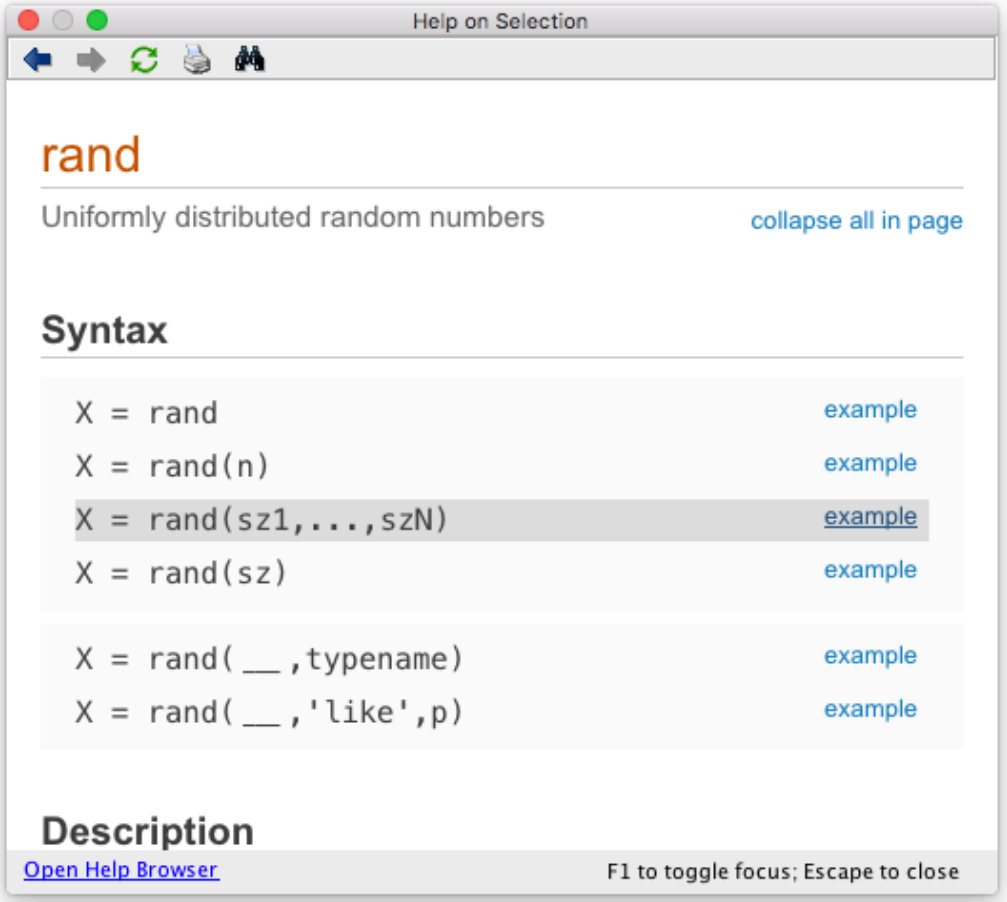
- Extrem schnell
- Früher sehr verbreitet

- Problem:
- Korrelation zwischen drei aufeinanderfolgenden Punkten

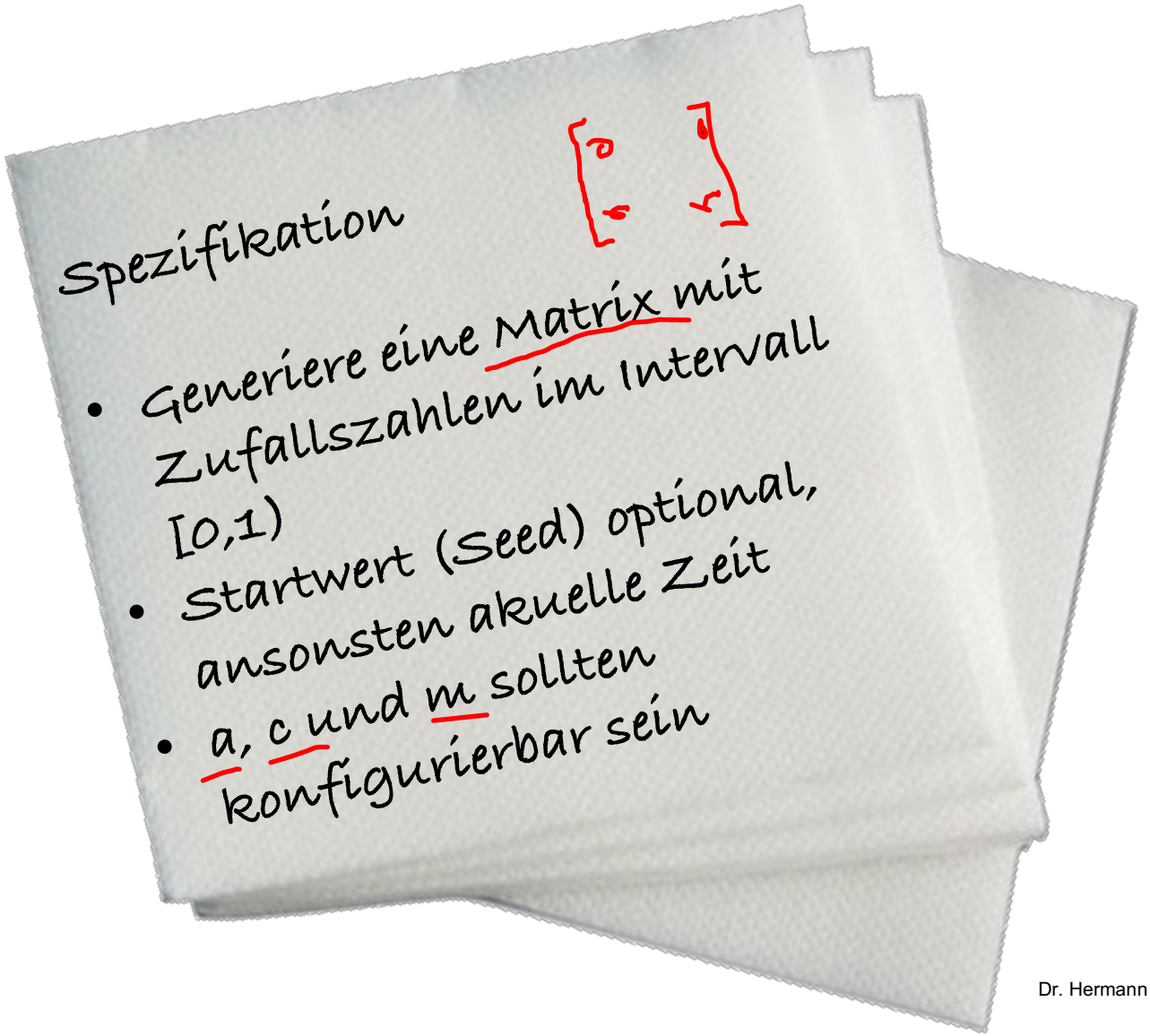
Implementierung eines LCG in Matlab

Erst mal: Wie funktioniert denn der ‘interne’ Random Generator?





Implementierung eines LCG in Matlab



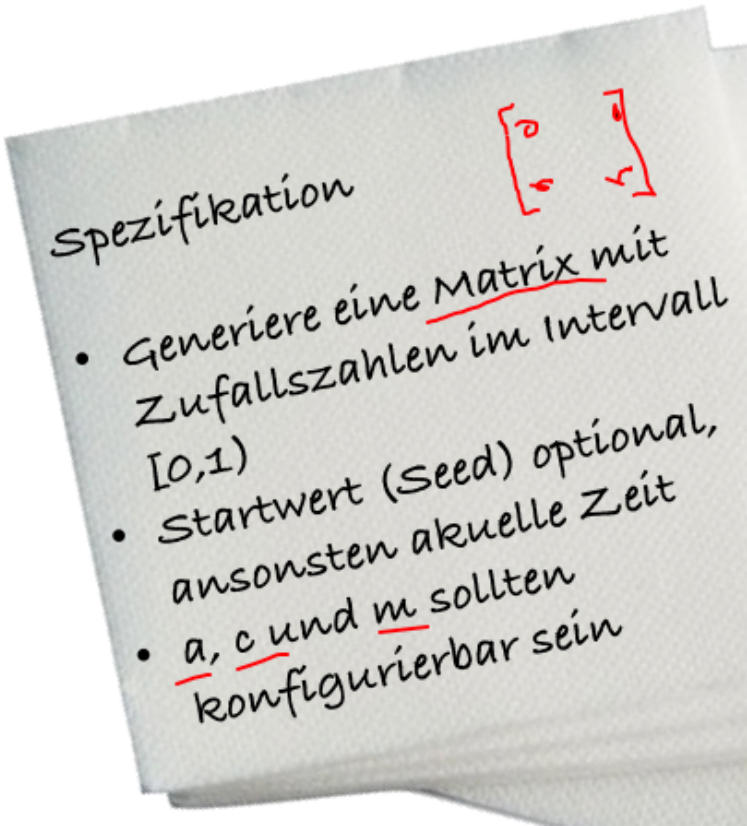
Implementierung eines LCG in Matlab

$$r_i = (a * r_{i-1} + c) \bmod m$$

```
r = mod(seed, m);
```

Startwert: z.B aktuelle Uhrzeit

```
r = mod(a*r + c, m);

disp(r/m)

r = mod(a*r + c, m);

disp(r/m)

...
```

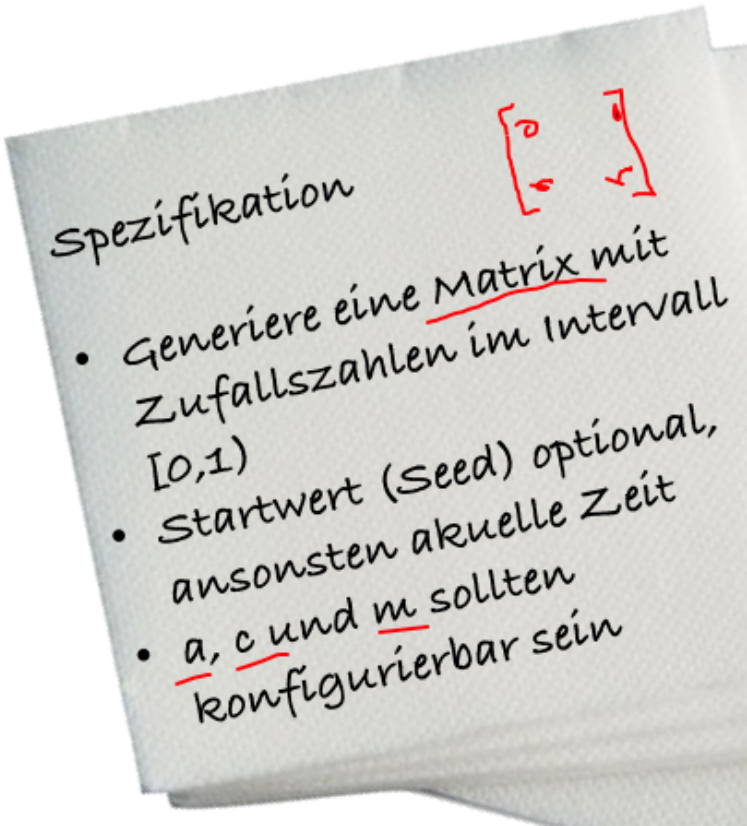


Implementierung eines LCG in Matlab

$$r_i = (a * r_{i-1} + c) \bmod m$$

```
r = mod(seed, m);
```

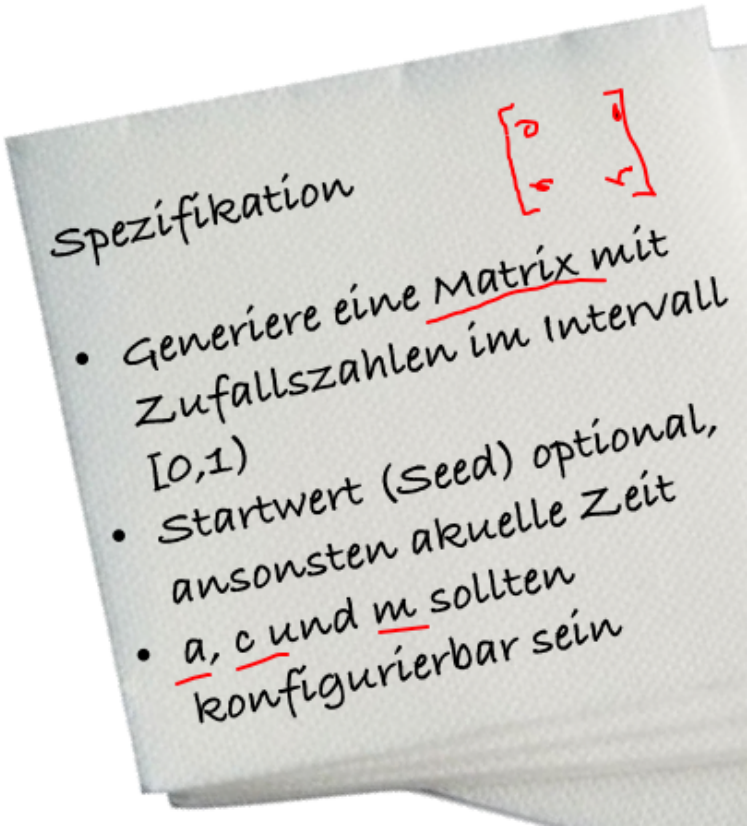
```
for row = 1:rows
    for col = 1:columns
        r = mod(a*r + c, m);
        Result(row, col) = r/m;
    end
end
```



Implementierung eines LCG in Matlab

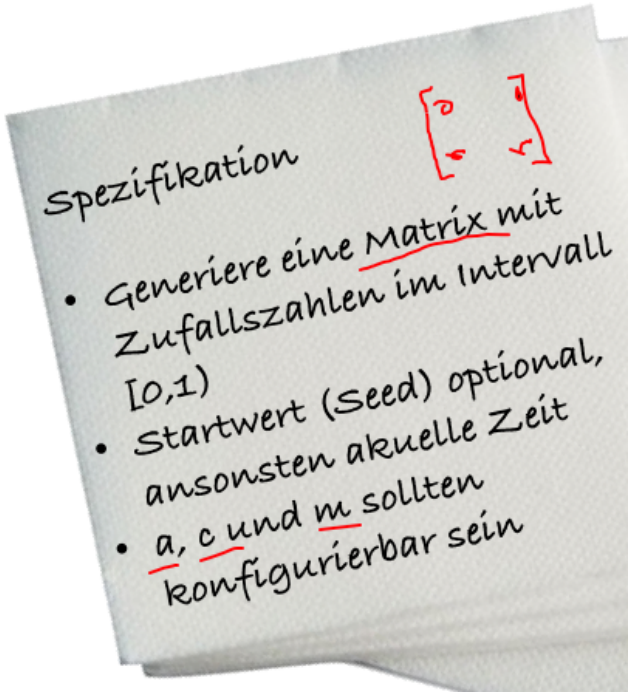
$$r_i = (a * r_{i-1} + c) \bmod m$$

```
r = mod(seed, m);

Result = zeros(rows, columns);

for row = 1:rows
    for col = 1:columns
        r = mod(a*r + c, m);
        Result(row, col) = r/m;
    end
end
```



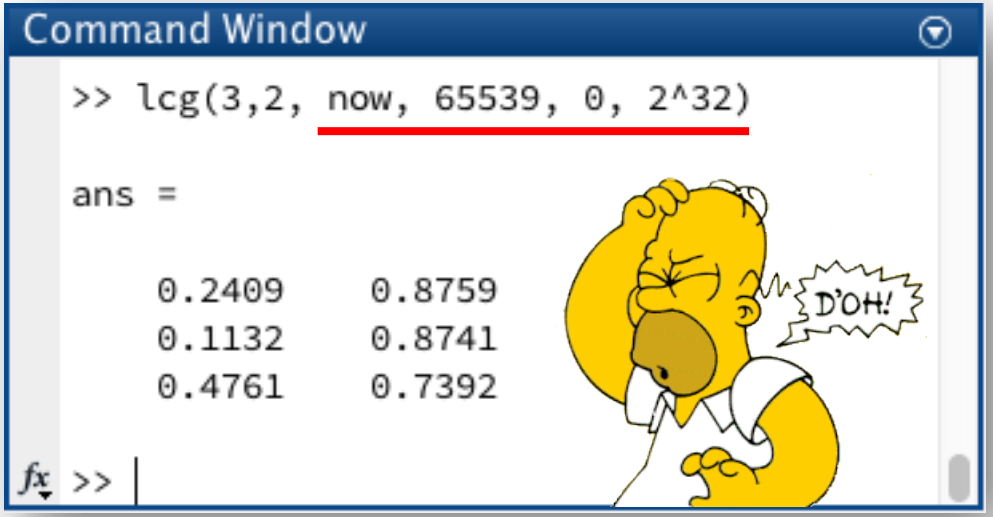
Implementierung eines LCG in Matlab

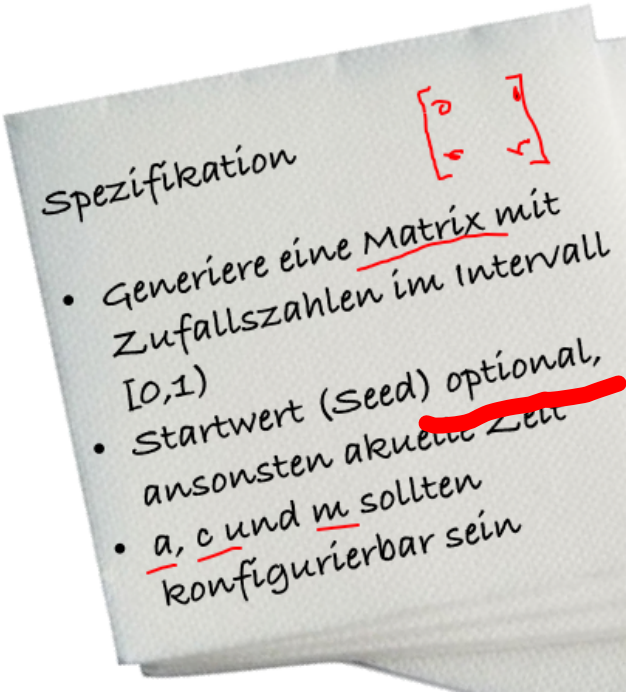
```
function Result = lcg(rows, columns, seed, a, c, m)
    r = mod(seed, m);

    Result = zeros(rows, columns);

    for row = 1:rows
        for col = 1:columns
            r = mod(a*r + c, m);
            Result(row, col) = r/m;
        end
    end

end
```



Implementierung eines LCG in Matlab

```
function Result = lcg(rows, columns, seed, a, c, m)

    r = mod(seed, m);

    Result = zeros(rows, columns);

    for row = 1:rows
        for col = 1:columns
            r = mod(a*r + c, m);
            Result(row, col) = r/m;
        end
    end

end
```





Implementierung eines LCG in Matlab

```
function Result = lcg(rows, columns, seed, a, c, m)
    if ~exist('seed', 'var')
        seed = now;
    end

    r = mod(seed, m);

    Result = zeros(rows, columns);

    for row = 1:rows
        for col = 1:columns
            r = mod(a*r + c, m);
            Result(row, col) = r/m;
        end
    end

end
```

optional

was ist mit diesen drei?

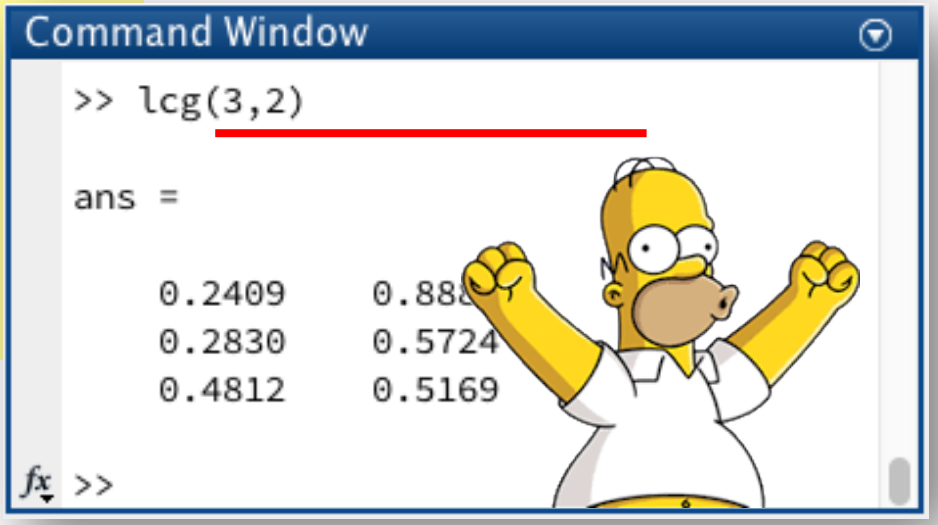
Spezifikation

- Generiere eine Matrix mit Zufallszahlen im Intervall [0,1)
- Startwert (Seed) optional, ansonsten akuelle Zeit
- a, c und m sollten konfigurierbar sein

Implementierung eines LCG in Matlab

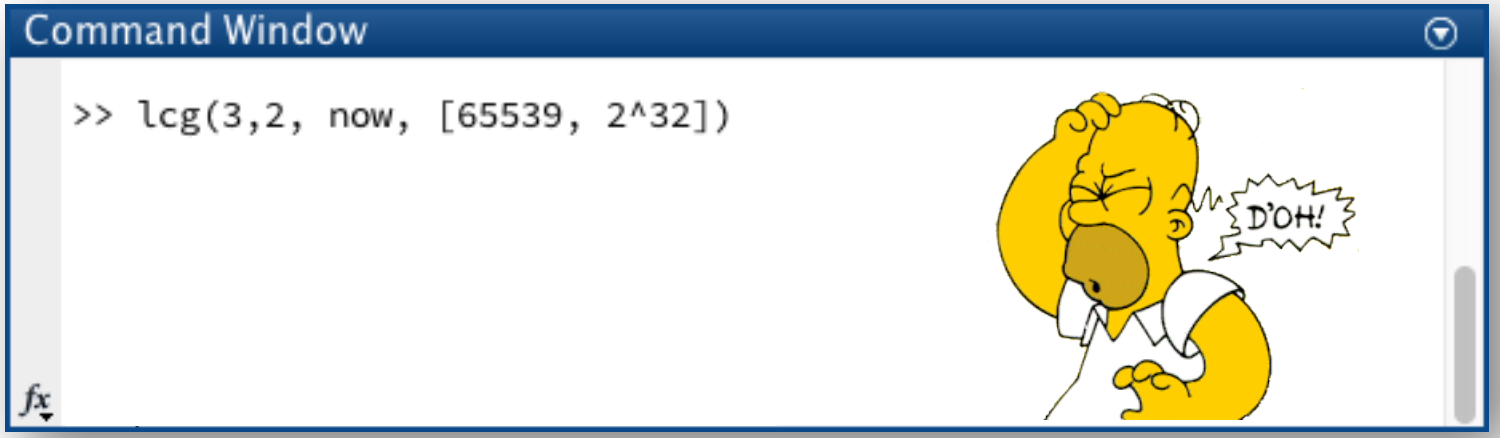
```
function Result = lcg(rows, columns, seed, config)

    if ~exist('seed', 'var')
        seed = now;
    end

    if exist('config','var')
        a = config(1);
        c = config(2);
        m = config(3);
    else
        % Default: RANDU
        a = 65539;
        c = 0;
        m = 2^32;
    end

    ...
```

optional: [a, c, m]



Implementierung eines LCG in Matlab

Wie sieht's aus bei fehlerhaften Eingaben?



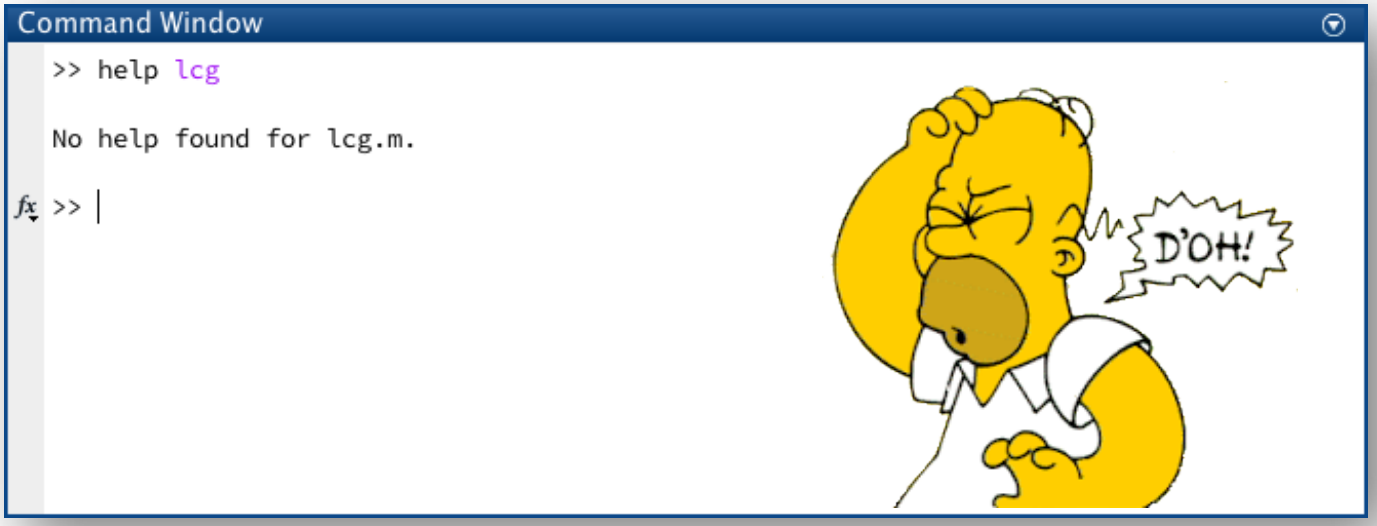
Implementierung eines LCG in Matlab

```
...

    if exist('config','var')
        dim = size(config);
        if (dim(2) ~= 3)
            error('Die Konfiguration muss eine Matrix der Form [ a c m ] sein.')
        end

        a = config(1);
        c = config(2);
        m = config(3);
    else
    ...
```

```
Command Window
>> lcg(3,2, now, [65539, 2^32])
Error using lcg (line 23)
Die Konfiguration muss eine Matrix der Form [ a c m ] sein.

fx >>
```

Implementierung eines LCG in Matlab

Hilfe für den Anwender?



Implementierung eines LCG in Matlab

```
function Result = lcg(rows, columns, seed, config)
% Erstellt eine Matrix mit Zufallszahlen mittels dem LCG Algorithmus.
% Wenn kein seed angegeben wird, wird die aktuelle Uhrzeit benuetzt.
%
% Erforderliche Parameter
% - rows:    Anzahl Reihen in der Matrix
% - columns: Anzahl Kolonnen in der Matrix
%
% Optionale Parameter
% - seed:    Startwert fuer die Zufallszahlen-Reihe.
%            Default: now
% - config:  Konfiguration des LCG in der Form [ a c m ]
%            Default: [ 65339, 0, 2^32 ]
```

```
    if ~exist('seed', 'var')
        seed = now;
    end

    ...
```

Command Window

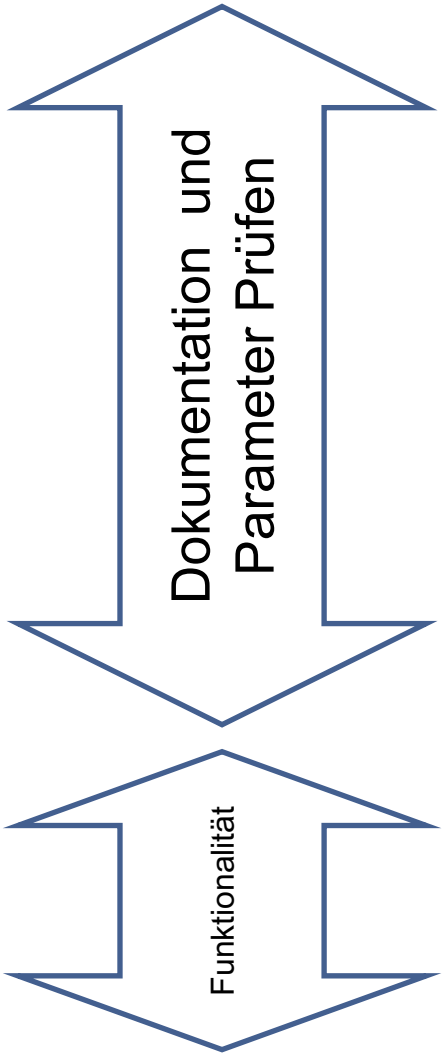
```
>> help lcg
  Erstellt eine Matrix mit Zufallszahlen mittels dem lcg Algorithmus.
  Wenn kein seed angegeben wird, wird die aktuelle Uhrzeit benuetzt.

  Erforderliche Parameter
  - rows:    Anzahl Reihen in der Matrix
  - columns: Anzahl Kolonnen in der Matrix

  Optionale Parameter
  - seed:    Startwert fuer die Zufallszahlen-Reihe.
             Default: now
  - config:  Konfiguration des lcg in der Form [ a c m ]
             Default: [ 65339, 0, 2^32 ]

fx >>
```

Implementierung eines LCG in Matlab

Editor - /Users/Hermann/Documents/MATLAB/lcg.m

script.mlx | lcg.m | birthday_problem.m

```
function Result = lcg(rows,columns,seed, config)
% Erstellt eine Matrix mit Zufallszahlen mittels dem LCG Algorithmus.
% Wenn kein seed angegeben wird, wird die aktuelle Uhrzeit benuetzt.
%
% Erforderliche Parameter
% - rows:    Anzahl Reihen in der Matrix
% - columns: Anzahl Kolonnen in der Matrix
%
% Optionale Parameter
% - seed:    Startwert fuer die Zufallszahlen-Reihe.
%            Default: now
% - config:  Konfiguration des LCG in der Form [ a c m ]
%            Default: [ 65339, 0, 2^32 ]

    %% If there is no given seed, use the current time as seed
    if ~exist('seed','var')
        seed = now;
    end

    %% Konfiguration des LCG mit dem Multiplikator (a), der Konstante (c) und dem Modulus (m)
    if exist('config','var')
        dim = size(config);
        if (dim(2) ~= 3)
            error('Die Konfiguration muss eine Matrix der Form [ a c m ] sein.')
        end
        a = config(1);
        c = config(2);
        m = config(3);
    else
        % Diese Werte stammen vom Zufallsgenerator RANDU (Frueher sehr ueblich).
        a = 65539;
        c = 0;
        m = 2^32;
    end

    %% Generierung der Matrix mit Zufallszahlen
    % Beim ersten Durchgang wird der seed als letzte Zufallszahl
    % angenommen und mittels modulo in das Intervall [0 .. m] gelegt.
    r = mod(seed,m);

    Result = zeros(rows,columns);
    for row = 1:rows
        for col = 1:columns
            %% Berechnung der naechsten Zufallszahl basierend auf der letzten Zufallszahl
            r = mod(a*r + c, m);

            % Normalisiere das Resultat. D.h. das Resultat ist im Intervall [0, 1)
            Result(row,col) = r/m;
        end
    end
end
```



Echt zufällig oder generiert?

1, 6, 15, 12, 13, 2, 11, 8, 9, 14, 7, 4, 5, 10, 3, 0, …

Keine Duplikate?

Das Geburtstags-Problem

Die Klasse meiner Tochter hat 23 Kinder (und zwei Lehrpersonen)

Wie gross ist die Wahrscheinlichkeit, dass zwei oder mehr Kinder am gleichen Tag Geburtstag haben?

0-2%

2-5%

5-10%

10-20%

20-40%

40-60%

> 60%

Ein kurzer Ausflug in die Statistik

Viel einfacher zu berechnen:

Wie gross ist die Wahrscheinlichkeit, dass jedes Kind einen unterschiedlichen Tag Geburtstag hat.

1. Kind: 365 mögliche Geburtstage

$$\frac{365}{365}$$

Ein kurzer Ausflug in die Statistik

Viel einfacher zu berechnen:

Wie gross ist die Wahrscheinlichkeit, dass jedes Kind einen unterschiedlichen Tag Geburtstag hat.

2. Kind: 364 mögliche Geburtstage, einer ist ja schon vom 1. Kind "besetzt"

$$\frac{365}{365} \cdot \frac{364}{365}$$

Ein kurzer Ausflug in die Statistik

Viel einfacher zu berechnen:

Wie gross ist die Wahrscheinlichkeit, dass jedes Kind einen unterschiedlichen Tag Geburtstag hat.

3. Kind: 363 mögliche Geburtstage, zwei sind ja schon von den ersten beiden Kindern besetzt

$$\underbrace{\frac{365}{365} \cdot \frac{364}{365} \cdot \frac{363}{365} \cdot \ldots \cdot \frac{343}{365}}_{23\ Terme}$$

Insgesamt 23 Kinder

Ein kurzer Ausflug in die Statistik

Daraus folgt...

Die Wahrscheinlichkeit für zwei oder mehr Kinder mit dem gleichen Geburtstag ist also 100% minus die berechnete Wahrscheinlichkeit:

$$1 - \left(\frac{365}{365} \cdot \frac{364}{365} \cdot \frac{363}{365} \cdot \ldots \cdot \frac{343}{365}\right) = 50.7\%$$

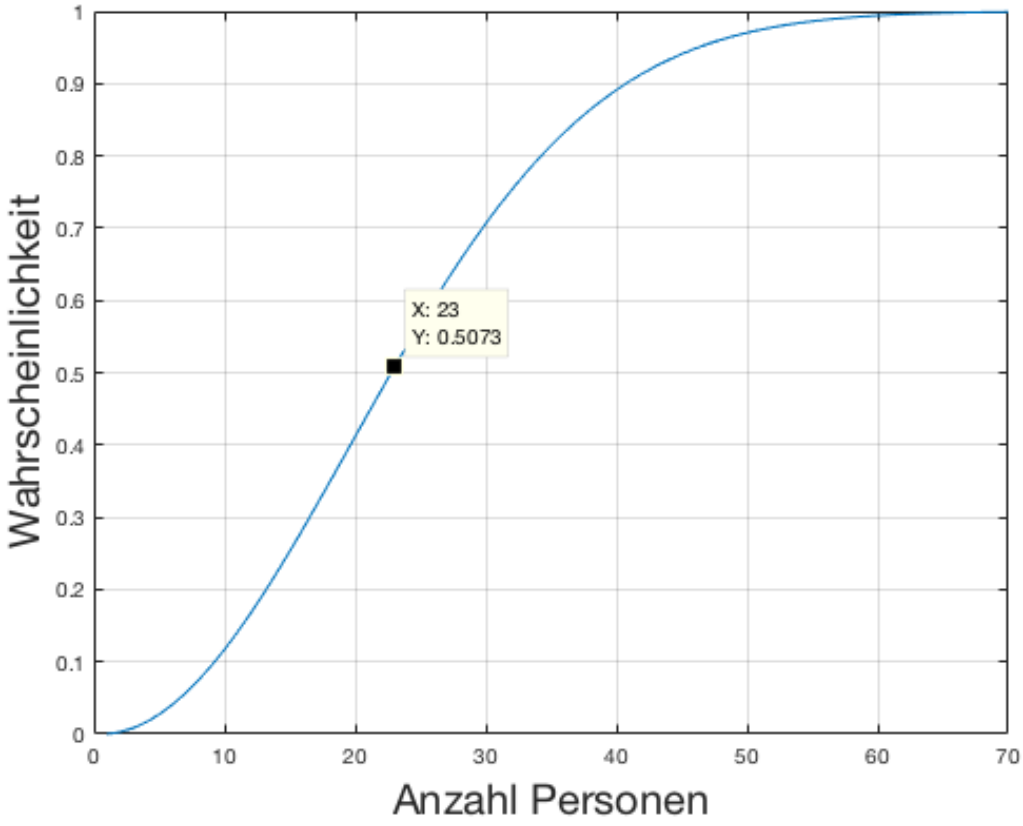
Berechnung der Wahrscheinlichkeit in Matlab

- Wir wollen diese Berechnung in Matlab machen
- Nicht nur für 23 Kinder, sondern für alle Gruppen-Grössen von 1 bis n (z.B. 70)
- Die Wahrscheinlichkeit grafisch darstellen

Berechnung der Wahrscheinlichkeit in Matlab

```
% Anzahl Tage im Jahr
d = 365;

% Was soll die groesste Gruppe sein, welche betrachtet wird
max_people = 70;

N = 1 : max_people;

% Initialisierung aller Wahrscheinlichkeiten auf 1 (neutrales Element der
% Multiplikation)
P = ones(1, max_people);

% Klein n geht also von 1 .. max_people.
% Fuer jedes n berechnen wir die gesuchte Wahrscheinlichkeit.
for n = N
    % Berechnung der Reihe 365/365 * 364/365 * ... * (365 - n + 1)/365
    for i = 1 : n
        P(n) = P(n) * (d - i + 1) / d;
    end
    % Das Resultat ist die inverse Wahrscheinlichkeit
    P(n) = 1 - P(n);
end

plot(P)
grid on
xlabel('Anzahl Personen', 'FontSize', 20)
ylabel('Wahrscheinlichkeit', 'FontSize', 20)
```

Berechnung der Wahrscheinlichkeit in Matlab



Kontrolle ist besser als Vertrauen

Wie können wir kontrollieren, ob das wirklich stimmt?

→ Sehr sehr viele Klassen besuchen und die Resultate auswerten

Also vielleicht 1'000 Klassen, statistisch relevant

Kontrolle ist besser als Vertrauen

Wie können wir kontrollieren, ob das wirklich stimmt?

→ Monte Carlo Simulation!

Monte Carlo Simulation in Matlab

Your Task!

```
d = 365;
max_people = 70;
at_least = 2;
experiments = 100;

% Simuli                                                    eit pro Tag
N = 1 :
P_sim =
for n =
    succ
    for

    end
    P_si
end

% Zeichr                                                    en Wahrscheinlichkeit
plot(P_s
axis([0
grid on
xlabel('Anzahl Personen','FontSize',20)
ylabel('Wahrscheinlichkeit','FontSize',20)
```