



# DER TRINÄRE ZUFALLSGENERATOR

Erfinder: Alexander Kramer

Alexander Kramer  
alexanderkramer@t-online.de

## Inhalt

Der trinäre Zufallsgenerator .....	3
Die Beschreibung des Systems .....	3
Das Schema des Zufallsgenerators .....	3
Die Zustandsmatrix und das Zufallsniveau .....	4
Das konstruktive und destruktive Niveau .....	5
Das konstruktive Niveau $\Delta N$ .....	5
Das destruktive Niveau $\Delta N$ .....	6
Das Bezugsniveau $B$ .....	7
Die komplexe Volatilität $V$ .....	8
Das Erwartungsniveau .....	9
Das konstruktive Erwartungsniveau $F$ .....	9
Das destruktive Erwartungsniveau $F$ .....	9
Die Iterationen .....	10
Die allgemeinen Beschreibungen .....	10
Die periodische Iteration .....	10
Die dynamische Breite .....	10
Die historischen Iterationen .....	11
Die Iteration und die Volatilität .....	11
Der konstruktive Verlauf .....	11
Das destruktive Verlauf .....	11
Die impliziten Iterationen .....	12
Die n-Vektor-Methode .....	12
Die 0-Vektor-Methode .....	14
Die Implementierung des trinären Zufallsgenerators in C++ .....	16
Die Beschreibung .....	16
Der Quellcode .....	16
Das Beispiel: Die Simulation eines Aktienkurses .....	19

## Der trinäre Zufallsgenerator

### Die Beschreibung des Systems

Im Folgenden wird eine kubische  $n \times n \times n$  - Matrix behandelt, welche als Basismatrix bezeichnet wird. Die Summe der Zufallselemente innerhalb der Basismatrix repräsentiert einen absoluten Zustand (Z). Aus dem absoluten Zustand (Z) folgt ein relativer Zustand (z). Die relativen Zustände beschreiben in einer weiteren  $a \times b$  Matrix das relative Zufallsniveau ( $\Delta N$ ) des Zufallsgenerators.

### Das Schema des Zufallsgenerators

Es wird ein Zufallselement X aus drei ganzzahligen Werten (1, 0, -1) bestimmt.

- Die Zahl 1 entspricht dem Zustand der Positivität
- Die Zahl 0 entspricht der Zustand der Neutralität
- Die Zahl -1 entspricht der Zustand der Negativität

Es wird eine Matrix, mit der Ordnung  $n = 10$ , aus  $n \times n \times n$  Zufallselementen gebildet. Die Summe der Matrixelemente definiert das Zustandselement.

(0/0/0)	1	2	3	4	5	6	7	8	9	10
1	1	1	-1	0	1	1	1	0	1	1
2	1	0	1	1	1	0	0	-1	0	-1
3	1	0	-1	0	-1	0	1	1	0	-1
4	1	0	1	-1	1	-1	0	0	1	1
5	-1	1	0	0	0	0	-1	1	0	-1
6	-1	-1	1	0	-1	1	1	-1	-1	0
7	1	1	-1	1	1	-1	0	1	0	-1
8	1	-1	-1	-1	0	1	-1	-1	-1	1
9	0	-1	1	-1	0	0	1	-1	0	1
10	0	0	1	-1	1	-1	-1	-1	1	0

Die zweidimensionale Ansicht des Matrixfeldes (0/0/0)

Die Anzahl  $A(n)$  der Zufallselemente und die Ordnung  $n$  der Basismatrix:

$$A(n) = n^3$$

Der absoluten Zustand Z wird aus der Summe der Zufallselemente X berechnet:

$$Z = \sum_{k=1}^{k=A(n)} X_k$$

Über den absoluten Zustand Z wird der relative Zustand z definiert:

$$z = \frac{Z}{n^3}$$

Die kleinste Einheit k des Zufallsgenerators lautet:

$$k = \pm n^{-3}$$

## Die Zustandsmatrix und das Zufallsniveau

Die Felder einer a x b Matrix enthalten Zustandselemente und definieren die Zustandsmatrix. Die Zustandselemente enthalten den absoluten oder relativen Zustand der Basismatrix.

0	1	2	3	4	5	6	7	8	9	10
1	-0,20%	-0,40%	0,60%	-2,00%	3,80%	-6,80%	0,10%	1,20%	3,20%	-1,30%
2	4,40%	-1,60%	1,00%	0,80%	-0,90%	-4,70%	-2,30%	0,20%	3,10%	-1,70%
3	2,00%	-2,80%	-0,90%	-1,90%	-3,50%	1,30%	2,00%	-3,30%	-2,00%	0,90%
4	0,80%	-1,40%	-0,40%	0,40%	-0,70%	8,60%	-5,10%	-2,40%	1,90%	1,60%
5	-2,00%	-0,20%	-1,40%	3,10%	-2,00%	0,10%	-3,30%	-2,70%	2,30%	1,00%
6	-2,60%	2,10%	1,70%	0,00%	5,90%	2,80%	2,60%	-0,50%	0,30%	-0,90%
7	-1,10%	-2,30%	2,70%	-2,70%	-1,00%	-6,30%	-1,30%	-1,80%	-1,20%	0,70%
8	0,10%	0,80%	-3,00%	-0,30%	-3,00%	4,90%	4,80%	2,00%	1,60%	2,30%
9	2,90%	-0,20%	2,20%	4,40%	0,20%	-0,50%	0,20%	0,10%	-0,10%	-2,00%
10	-0,20%	2,40%	0,90%	1,10%	-0,50%	3,60%	-2,50%	3,00%	1,70%	1,00%

In dieser Zustandsmatrix wird - beispielhaft - der relative Zustand angegeben.

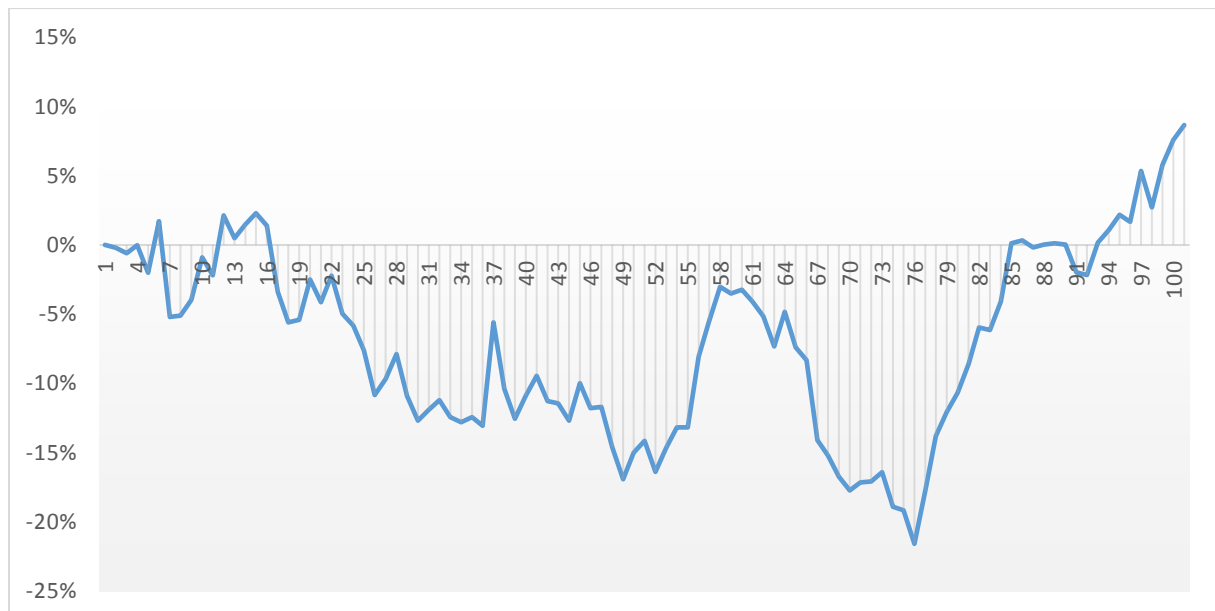
Das relative Zufallsniveau  $\Delta N$  entsteht aus den Zustandselementen der a x b Matrix.

$$\Delta N = \prod_{(i,j)=(1,1)}^{(i,j)=(a,b)} (1 + z_{(i,j)}) - 1$$

Es wird das absolute Zufallsniveau über das relative Zufallsniveau definiert

$$N = \Delta N + 1$$

Die Grafik bildet den Verlauf eines relativen Zufallsniveaus der Zustandsmatrix ab.



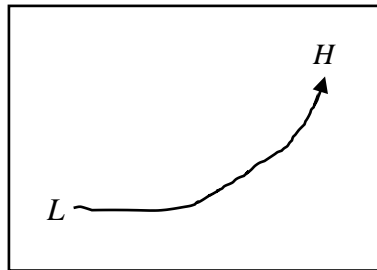
Das relative Zufallsniveau schwankt während den Iterationsschritten 1 - 100. Es setzt sich, in den Iterationsschritten 13 – 76 in einem destruktiven Verlauf und ab den Iterationsschritten 76 – 100 in einem konstruktiven Verlauf fort.

## Das konstruktive und destruktive Niveau

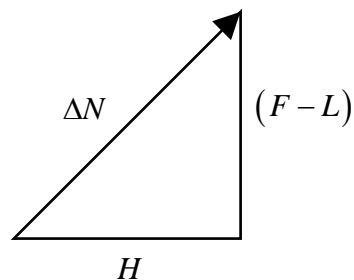
Bei der Berechnung der Erwartungsniveaus  $F$  wird zwischen dem relativen Zufallsniveau in konstruktivem und destruktivem Verlauf unterschieden. Um den Zufallsverlauf berechenbar zu machen, werden im Iterationsverlauf, in Bezug auf die das Erwartungsniveau  $F$ , zwei Parameter ( $L$  und  $H$ , mit  $L < H$ ) definiert, welche sich auf das Nullniveau ( $\Delta N = 0$ ) beziehen.

### Das konstruktive Niveau $\Delta N$

Die Skizze des konstruktiven Niveauperlaufs



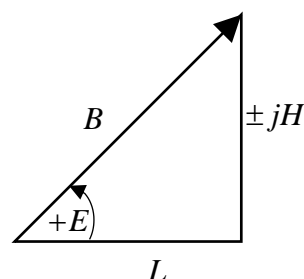
Das relative Zufallsniveau  $\Delta N$  im konstruktiven Verlauf wird aus dem Vektor des H-Niveaus und der Niveaudifferenz des Erwartungsniveaus  $F$  und des L-Niveaus bestimmt:



Es wird der Nullpunkt des relativen Zufallsniveaus  $\Delta N = 0$  bestimmt:

$$\Delta N = \sqrt{H^2 + (F - L)^2} = 0 \quad \rightarrow \quad F = L \pm jH$$

Der komplexe Vektor des konstruktiven Niveaus

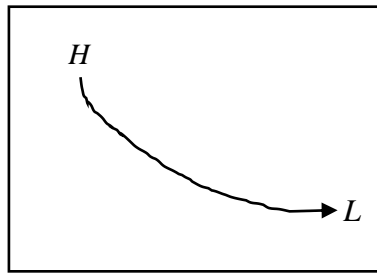


Das komplexe Erwartungsniveau  $F$  des konstruktiven Verlaufs lautet:

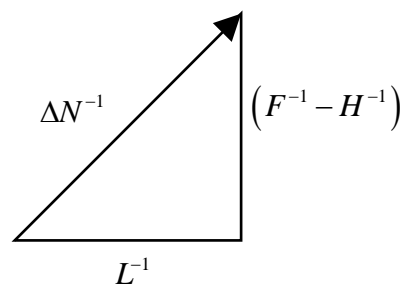
$$F = \sqrt{L^2 + H^2} \cdot e^{\pm j \arctan\left(\frac{H}{L}\right)} = B \cdot e^{\pm jE}$$

## Das destruktive Niveau $\Delta N$

Die Skizze des destruktiven Niveauverlaufs



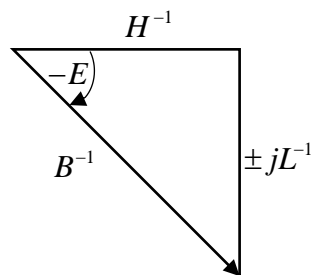
Das relative Zufallsniveau  $\Delta N$  im destruktiven Verlauf wird aus dem Vektor des L-Niveaus und der Niveaudifferenz des H-Niveaus und des Erwartungsniveaus F bestimmt:



Es wird der Nullpunkt des relativen Zufallsniveau  $\Delta N^{-1} = 0$  bestimmt:

$$\Delta N^{-1} = \sqrt{L^{-2} + (F^{-1} - H^{-1})^{-2}} = 0 \quad \rightarrow \quad F^{-1} = H^{-1} \pm jL^{-1}$$

Der komplexe Vektor des destruktiven Niveaus



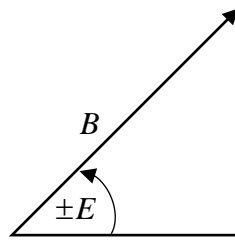
Die komplexe Erwartungsniveau F des destruktiven Verlaufs lautet:

$$F^{-1} = \sqrt{L^{-2} + H^{-2}} \cdot e^{\pm j \arctan\left(\frac{L^{-1}}{H^{-1}}\right)}$$

$$\rightarrow F = \sqrt{L^{-2} + H^{-2}}^{-1} \cdot e^{-\left(\pm j \arctan\left(\frac{L^{-1}}{H^{-1}}\right)\right)} = B \cdot e^{-jE}$$

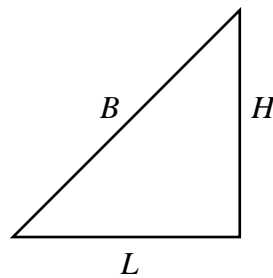
## Das Bezugsniveau B

Das komplexe Erwartungsniveau wird durch das Parameterpaar L und H über das Bezugsniveau B und die Eigendynamik E beschrieben. Das Bezugsniveau B ist Element des Erwartungsniveaus F.



$$F = B \cdot e^{\pm jE}$$

Das konstruktive Bezugsniveau

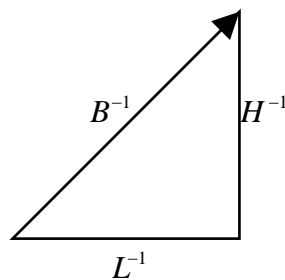


$$B = \sqrt{L^2 + H^2}$$

Die Eigendynamik E

$$E = \pm \arctan\left(\frac{H}{L}\right)$$

Das destruktive Bezugsniveau



$$B = \frac{1}{\sqrt{L^{-2} + H^{-2}}}$$

Die Eigendynamik E

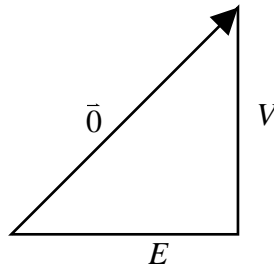
$$E = \pm \arctan\left(\frac{H}{L}\right)$$

## Die komplexe Volatilität V

Die komplexe Volatilität wird durch den Nullvektor von Eigendynamik E und Volatilität zum Ausdruck gebracht.

$$\vec{0} = \begin{pmatrix} V \\ E \end{pmatrix}$$

Es wird der Nullvektor der Eigendynamik E und der Volatilität V definiert.



$$0 = \sqrt{E^2 + V^2} \quad \rightarrow \quad E = \pm jV$$

Es wird der Imaginärteil des konstruktiven Verlaufs bestimmt.

$$\begin{aligned} E &= \pm jV \\ \pm \arctan\left(\frac{H}{L}\right) &= \pm jV \\ \rightarrow \quad jH &= L \cdot \tanh(2V) \end{aligned}$$

Es wird die Volatilität des konstruktiven K-Niveaus definiert.

$$\begin{aligned} K &= L \pm jH = L \pm L \cdot \tanh(2V) \\ \rightarrow \quad V &= \mp \frac{1}{2} \operatorname{arctanh}\left(\frac{K}{L} - 1\right) \end{aligned}$$

Es wird der Imaginärteil des destruktiven Verlaufs bestimmt.

$$\begin{aligned} -E &= \pm jV \\ -\left(\pm \arctan\left(\frac{L^{-1}}{H^{-1}}\right)\right) &= \pm jV \\ \rightarrow \quad jL^{-1} &= -H^{-1} \cdot \tanh(2V) \end{aligned}$$

Es wird die Volatilität des destruktiven K-Niveaus definiert.

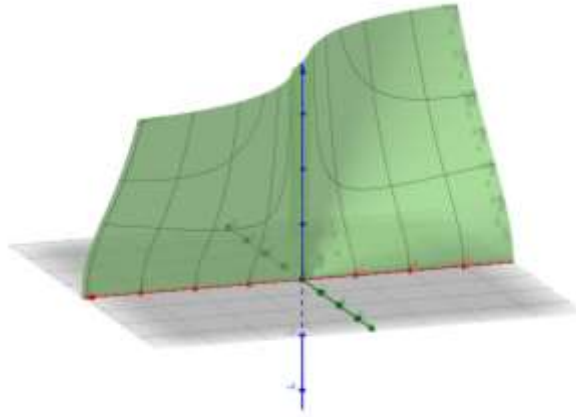
$$\begin{aligned} K^{-1} &= H^{-1} \pm jL^{-1} = H^{-1} \mp H^{-1} \cdot \tanh(2V) \\ V &= \pm \frac{1}{2} \operatorname{arctanh}\left(\frac{H}{K} - 1\right) \end{aligned}$$



## Das Erwartungsniveau

### Das konstruktive Erwartungsniveau F

Das konstruktive Erwartungsniveau F wird über die implizite Volatilität V des konstruktiven Niveauverlaufs bestimmt.

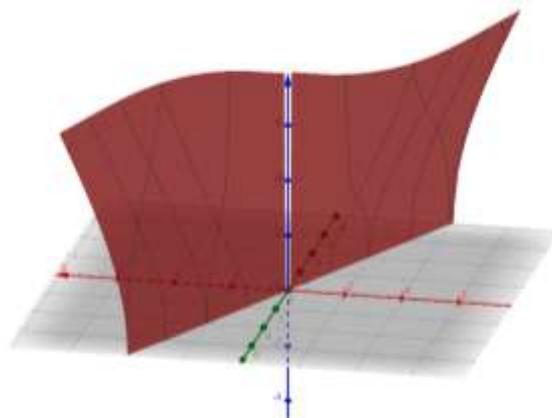


$$V = \mp \frac{1}{2} \operatorname{arctanh} \left( \frac{K}{L} - 1 \right)$$

$$\rightarrow F_{(K)} = B \cdot e^{\pm jE} = B \cdot e^{-V} = B \cdot e^{\pm \frac{1}{2} \operatorname{arctanh} \left( \frac{K}{L} - 1 \right)}$$

### Das destruktive Erwartungsniveau F

Das destruktive Erwartungsniveau F wird über die implizite Volatilität V des destruktiven Niveauverlaufs bestimmt.



$$V = \pm \frac{1}{2} \operatorname{arctanh} \left( \frac{H}{K} - 1 \right)$$

$$\rightarrow F_{(K)} = B \cdot e^{-V} = B \cdot e^{\mp \frac{1}{2} \operatorname{arctanh} \left( \frac{H}{K} - 1 \right)}$$

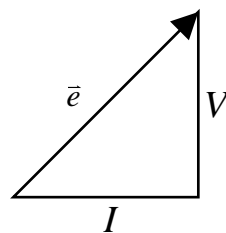
## Die Iterationen

### Die allgemeinen Beschreibungen

Die Zahl der Iterationsschritte wird als „Iteration“ definiert. Die Methode zur Berechnung der Iterationsdifferenz hypothetisiert die Annahme, dass die historische Eigendynamik einen weiteren Verlauf des relativen Zufallsniveaus impliziert. Es wird zwischen historischen und impliziten Iterationen unterschieden.

- Die historischen Iterationen definieren die Anzahl der Iterationen der Parameter L und H in einer Periode
- Die impliziten Iterationen definieren die Anzahl der maximalen Iterationsschritte in n Perioden, bei welchen das Erwartungsniveau F mindestens erreicht ist.

Es werden die historischen Iterationen bestimmt, wozu der Einheitsvektor der Iterationen I und der Volatilitäten V definiert wird.



$$\bar{e} = \begin{pmatrix} V \\ I \end{pmatrix} \quad \rightarrow \quad 1 = \pm \sqrt{I^2 + V^2}$$
$$\quad \quad \quad \rightarrow \quad I = \pm \sqrt{1 - V^2}$$

### Die periodische Iteration

Die Iterationen werden über  $t = t + 1$  mit dem Startwert  $t = 0$  definiert. Die natürliche Zahl der mittleren Iterationsschritte definiert die Anzahl der Perioden.

$$I = \frac{t}{\tau} = n$$

Die Anzahl der mittleren Iterationsschritte, wird aus der Differenz der historischen Iterationen I des L- und H-Niveaus bestimmt.

$$I_H - I_L = \frac{t_H}{\tau} - \frac{t_L}{\tau} \quad \rightarrow \quad \tau = \frac{t_H - t_L}{I_H - I_L}$$

### Die dynamische Breite

Die impliziten Iterationen lassen sich über die dynamische Breite darstellen. Die dynamische Breite ist durch den Spitze-Spitze-Wert aus der konstruktiven und destruktiven Eigendynamik definiert.

$$\Lambda = (+E) - (-E) = \pm jV - (\mp jV) = \pm j \cdot 2V \quad \rightarrow \quad \pm j\Lambda = 2V$$

## Die historischen Iterationen

### Die Iteration und die Volatilität

Die Iteration I ist abhängig von der Volatilität

$$K = B \cdot e^{-V_K} \quad \rightarrow \quad V_K = \ln\left(\frac{B}{K}\right)$$
$$\rightarrow \quad I_K = +\sqrt{1-V_K^2} = +\sqrt{1-\ln^2\left(\frac{B}{K}\right)}$$

### Der konstruktive Verlauf

Die Konstruktionsiteration des L-Niveaus:

$$K = L$$
$$I_L = \sqrt{1-\ln^2\left(\frac{B}{K}\right)} = \sqrt{1-\ln^2\left(\frac{\sqrt{L^2+H^2}}{L}\right)}$$

Die Konstruktionsiteration des H-Niveaus:

$$K = H$$
$$I_H = \sqrt{1-\ln^2\left(\frac{B}{K}\right)} = \sqrt{1-\ln^2\left(\frac{\sqrt{L^2+H^2}}{H}\right)}$$

### Das destruktive Verlauf

Die Destruktionsiteration des L-Niveaus:

$$K = L$$
$$I_L = \sqrt{1-\ln^2\left(\frac{B}{K}\right)} = \sqrt{1-\ln^2\left(\frac{\sqrt{L^{-2}+H^{-2}}}{L}\right)}$$

Die Destruktionsiteration des H-Niveaus:

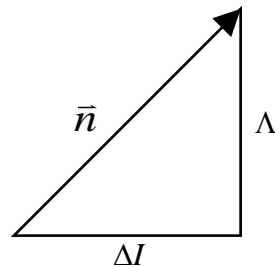
$$K = H$$
$$I_H = \sqrt{1-\ln^2\left(\frac{B}{K}\right)} = \sqrt{1-\ln^2\left(\frac{\sqrt{H^{-2}+L^{-2}}}{H}\right)}$$

## Die impliziten Iterationen

### Die n-Vektor-Methode

#### Der dynamische Iterationsvektor

Der n-Vektor zur Ermittlung der Erwartungspunkte F ist durch die implizite Iterationsdifferenz und die dynamischen Breite definiert. Der Betrag des n-Vektors gibt die Periode an.



$$\vec{n} = \begin{pmatrix} \Lambda \\ \Delta I \end{pmatrix} : \quad \rightarrow \quad n = \sqrt{(\Delta I)^2 + \Lambda^2}$$

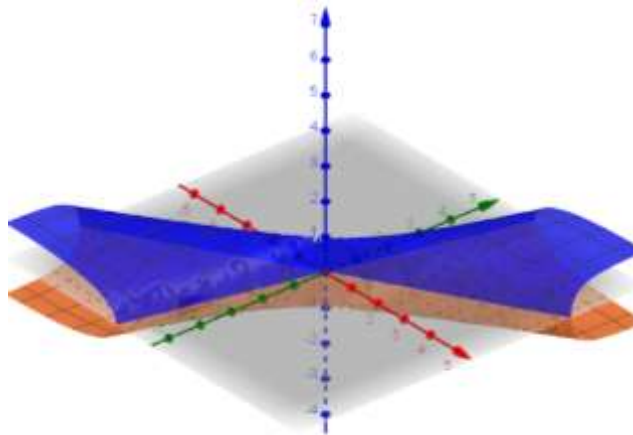
$$\rightarrow \quad \Delta I = \pm \sqrt{n^2 - \Lambda^2}$$

#### Die Erwartungspunkte des konstruktiven Verlaufs

Das konstruktive Erwartungsniveau F des iterativen Erwartungspunkts t(F) des K-Niveaus

$$F = B \cdot e^{\pm \frac{1}{2} \operatorname{arctanh}\left(\frac{K}{L}-1\right)}$$

Die implizite Iteration des konstruktiven Verlaufs



$$\Delta I = +\sqrt{n^2 - \Lambda^2} = +\sqrt{n^2 - \operatorname{arctanh}^2\left(\frac{K}{L}-1\right)}$$

Der iterative Erwartungspunkt des K-Niveaus:

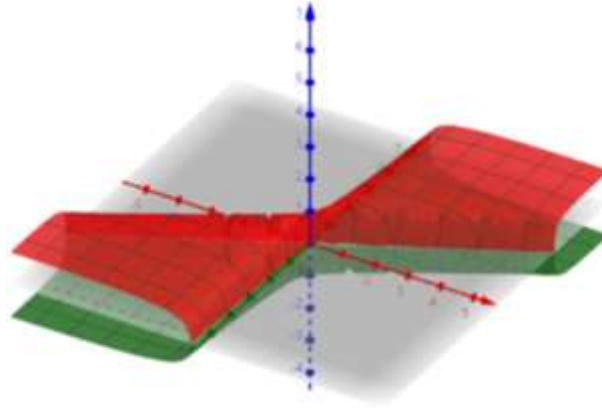
$$t_F = t_K + I_K \cdot \Delta t = t_K + I_K \cdot \tau \cdot \Delta I = t_K + \frac{t_H - t_L}{I_H - I_L} \cdot I_K \cdot \sqrt{n^2 - \operatorname{arctanh}^2\left(\frac{K}{L}-1\right)}$$

*Die Erwartungspunkte des destruktiven Verlaufs*

Das destruktive Erwartungsniveau F des iterativen Erwartungspunkts t(F) des K-Niveaus

$$F = B \cdot e^{\mp \frac{1}{2} \operatorname{arctanh}\left(\frac{H}{K}-1\right)}$$

Die implizite Iteration des destruktiven Verlaufs



$$\Delta I = -\sqrt{n^2 - \Lambda^2} = -\sqrt{n^2 - \operatorname{arctanh}^2\left(\frac{H}{K}-1\right)}$$

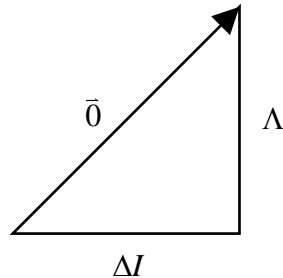
Der iterative Erwartungspunkt des K-Niveaus:

$$t_F = t_K + I_K \cdot \Delta t = t_K + I_K \cdot \tau \cdot \Delta I = t_K - \frac{t_H - t_L}{I_H - I_L} \cdot I_K \cdot \sqrt{n^2 - \operatorname{arctanh}^2\left(\frac{H}{K}-1\right)}$$

## Die 0-Vektor-Methode

### Der dynamische Iterationsvektor

Der Nullvektor zur Ermittlung der Erwartungspunkte F ist durch die implizite Iterationsdifferenz und die dynamischen Breite definiert. Der Betrag des Nullvektors gibt die Periode an.



$$\bar{0} = \begin{pmatrix} \Lambda \\ \Delta I \end{pmatrix} : \quad \rightarrow \quad 0 = \sqrt{(\Delta I)^2 + \Lambda^2}$$

$$\rightarrow \quad \Delta I = \pm j\Lambda = 2V$$

### Die Erwartungspunkte des konstruktiven Verlaufs

Das Erwartungsniveau F des konstruktiven Verlaufs

$$F = \sqrt{L^2 + H^2} \cdot e^{\pm \arctanh\left(\frac{K}{L}-1\right)}$$

Die implizite Iteration des konstruktiven Verlaufs besitzt ein positives Vorzeichen



$$\Delta I = +j\Lambda = 2V_+ = +\arctanh\left(\frac{K}{L}-1\right)$$

Der iterative Erwartungspunkt t(F) des K-Niveaus

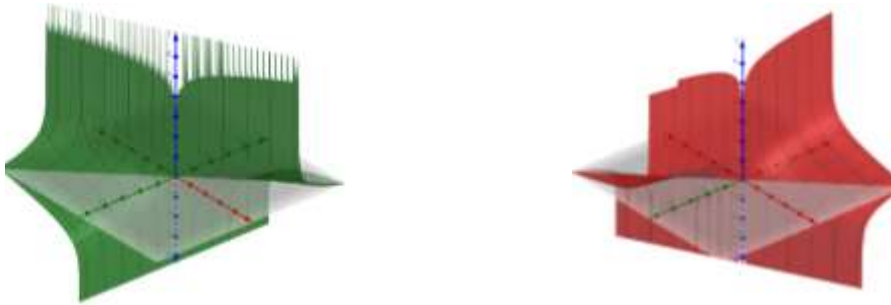
$$t_F = t_K + I_K \cdot t\Delta = t_K + I_K \cdot \tau \cdot \Delta I = t_K + \frac{t_H - t_L}{I_H - I_L} \cdot I_K \cdot \arctanh\left(\frac{K}{L}-1\right)$$

Die Erwartungspunkte des destruktiven Verlaufs

Das Erwartungsniveau F des destruktiven Verlaufs

$$F = \sqrt{L^{-2} + H^{-2}}^{-1} \cdot e^{\mp \operatorname{arctanh}\left(\frac{H}{K}-1\right)}$$

Die implizite Iteration des destruktiven Verlaufs besitzt ein negatives Vorzeichen



$$\Delta I = -j\Lambda = 2V_- = -\operatorname{arctanh}\left(\frac{H}{K}-1\right)$$

Der iterative Erwartungspunkt t(F) des K-Niveaus

$$t_F = t_K + I_K \cdot \Delta t = t_K + I_K \cdot \tau \cdot \Delta I = t_K - \frac{t_H - t_L}{I_H - I_L} \cdot I_K \cdot \operatorname{arctanh}\left(\frac{H}{K}-1\right)$$

# Die Implementierung des trinären Zufallsgenerators in C++

## Die Beschreibung

Im Quellcode werden vier Dateien angelegt. „\Ergebnisse\Zufallselemente\ Zufallselemente.txt“, „\Ergebnisse\Zustandselemente\ Zustandselemente.txt“, „\Ergebnisse\Zufallselemente\ Erwartungsniveau.txt“ und „\Ergebnisse\Niveauverlauf\ Niveauverlauf.txt“. Der Algorithmus schreibt die Basismatrizen in die Datei „\Zufallselemente.txt“. Die Zustandsmatrix wird in der Datei „\Zustandselemente.txt“, die Erwartungsniveaus in die Datei „\Erwartungsniveaus.txt“ und die relativen Zufallsniveaus in der Datei „\Niveauverlauf.txt“ geschrieben.

## Der Quellcode

```
/*
Der Entwurf des trinären Zufallsgenerators
*/
#include <iostream>
#include <fstream>
#include <time.h>
#include <stdlib.h>
#include <iomanip>
#include <string>
#include <sstream>
#include <unistd.h>
#include <cmath>
// Zeilen, Spalten und Ordnung über Präprozessor festlegen
#define rel_abs 0.0 // '0.0' für absolutes Zufallsniveau; '1.0' für relatives Zufallsniveau
#define H_L_MAX 2.0 // Der Maximalwert (H=2L) der Erwartungsniveaus soll 2.0 sein
#define Ordnung 10 // Gibt die kubische Größe der Basismatrix an. Bei n = 10 enthält die Basismatrix
10^3 Elemente
#define Zeilen 12 // Anzahl der Zeilen (Bei senkrechter Ausgabe z.B. '31*12' eingeben.)
#define Spalten 31 // Anzahl der Spalten (Bei waagerechter Ausgabe z.B. '31*12' eingeben.)
//Namespace std als Standard
using namespace std;
//Funktionen definieren
double Zufallsgenerator(string, string, string, string, int, int, int, double);
int Zustandselement(string, string, string, int);
//main-Funktion
int main(int argc, char *argv[]) {
// C++ Zufallsgenerator einstellen
srand(unsigned(time(NULL)));
// Dateinamen definieren
string Zufallselemente = "Zufallselemente";
string Zustandselemente = "Zustandselemente";
string Niveauverlauf = "Niveauverlauf";
string Erwartungsniveau = "Erwartungsniveau";
//Niveau einstellen
double Niveau = 1.0;
//Zufallsgenerator starten
Zufallsgenerator(Zufallselemente, Zustandselemente, Niveauverlauf, Erwartungsniveau, Spalten,
Zeilen, Ordnung, Niveau);
//Rückgabe 0
return 0;
}

double Zufallsgenerator(string Zufallselemente_str, string Zustandselemente_str, string Niveauverlauf_str,
string Erwartungsniveau_str, int Spalten_0, int Zeilen_0, int Ordnung_0, double Niveau)
{
//Verzeichnisse konfigurieren
string Zufallselemente_dir = "Ergebnisse\\" + Zufallselemente_str;
string Zustandselemente_dir = "Ergebnisse\\" + Zustandselemente_str;
string Niveauverlauf_dir = "Ergebnisse\\" + Niveauverlauf_str;
string Erwartungsniveau_dir = "Ergebnisse\\" + Erwartungsniveau_str;
//Verzeichnisse anlegen (Ergebnisse\\...)
mkdir("Ergebnisse");
mkdir(Zufallselemente_dir.c_str());
mkdir(Zustandselemente_dir.c_str());
mkdir(Niveauverlauf_dir.c_str());
mkdir(Erwartungsniveau_dir.c_str());
//Dateinamen erstellen
string Zufallselemente_dat = Zufallselemente_dir + "\\" + Zufallselemente_str;
string Zustandselemente_dat = Zustandselemente_dir + "\\" + Zustandselemente_str + ".txt";
string Niveauverlauf_dat = Niveauverlauf_dir + "\\" + Niveauverlauf_str + ".txt";
string Erwartungsniveau_dat = Erwartungsniveau_dir + "\\" + Erwartungsniveau_str + ".txt";
// Dateien für Schreib-Zugriff öffnen
ofstream Niveaus(Niveauverlauf_dat.c_str(), ios::out);
ofstream Zustandselemente(Zustandselemente_dat.c_str(), ios::out);
ofstream Erwartungsniveaus(Erwartungsniveau_dat.c_str(), ios::out);
//Die Initialisierungsvariablen des Zufallsgenerators
double L = Niveau;
double Erwartungsniveau = L;
// Schreibdateien zum Eintragen der Werte konfigurieren
```



```

Niveaus << "0";
Zustandselemente << "0";
//Schreibdateien konfigurieren
for(int a=0; a < Spalten_0; a++)
{
    Niveaus.width(11);
    Niveaus << a + 1;
    Zustandselemente.width(11);
    Zustandselemente << a + 1;
    Erwartungselemente.width(11);
    Erwartungselemente << a + 1;
}
// Mit der ersten und zweiten For-Schleife wird die 12 x 31 Statusmatrix berechnet
// Erste For-Schleife für Monate, welche in den Schreibdateien als Zeilen eingetragen werden
for(int b=0; b < Zeilen_0 ; b++)
{
    //Schreibdateien konfigurieren
    Niveaus << endl << b + 1;
    Erwartungselemente << endl << b + 1;
    Zustandselemente << endl << b + 1;
    //Zweite For-Schleife für Tage, welche in den Schreibdateien als Spalten eingetragen werden
    for(int c=0; c < Spalten_0 ; c++)
    {
        //Kalender konfigurieren
        if( (b%12+1) == 2 && ( c%31 + 1 > 28 ) ) //Februar
        {
            continue;
        }
        else if( (b%12+1) < 8 && (b%12+1)%2 == 0 && c%31 == 30-1 ) //August
        {
            continue;
        }
        else if( (b%12+1)%2 == 1 && (b%12+1) >= 9 && c%31 == 31-1) //September - Dezember
        {
            continue;
        }
        //Index der Dateien konfigurieren
        stringstream b_0;        b_0 << b + 1;
        stringstream c_0;        c_0 << c + 1;
        //absolutes und relatives Zustandsfeld berechnen
        int Zustand_abs = Zustandselement(b_0.str(), c_0.str(), Zufallselemente_dat,
        Ordnung_0);
        double Zustand_rel = double(Zustand_abs)/double(Ordnung_0 * Ordnung_0 *
        Ordnung_0);
        //relativen Zustand dem relativen Niveau anfügen
        Niveau *= 1.0 + Zustand_rel;
        // Erwartungsniveau berechnen
        if ( Niveau > L && Niveau < L * (H_L_MAX))
        {
            Erwartungsniveau = sqrt((L * L) + (Niveau * Niveau) * exp ( + 0.5 * atanh( Niveau
            / L - 1 )));
        }
        else if ( Niveau < L && Niveau > L/(H_L_MAX) )
        {
            Erwartungsniveau = 1.0/sqrt(1.0/(L * L) + 1.0/(Niveau * Niveau) * exp ( - 0.5 *
            atanh( L / Niveau - 1 )));
        }
        else if ( ( Niveau > L && Niveau < L * (H_L_MAX) && Niveau < L * (H_L_MAX) &&
            Erwartungsniveau > L * (H_L_MAX) ) ^ ( Niveau < L && Niveau > L/(H_L_MAX) &&
            Erwartungsniveau < L / (H_L_MAX) ) )
        {
            L = Niveau;
        }
        //relatives Niveau in Datei schreiben
        Niveaus.width(10);
        Niveaus << setprecision(3) << (Niveau - (rel_abs)) * 100.0 << "%";
        //Erwartungsniveau in Datei schreiben
        Erwartungselemente.width(10);
        Erwartungselemente << setprecision(3) << (Erwartungsniveau - (rel_abs)) * 100.0
        << "%";
        //relativen Zustand in Datei schreiben
        Zustandselemente.width(10);
        Zustandselemente << setprecision(3) << Zustand_rel * 100.0 << "%";
        //Bildschirmausgabe
        cout << "Spalte: " << (c + 1) << "           Zeile: " << (b + 1) << " Niveau: " <<
        (Niveau - (rel_abs)) * 100.0 << "%" << endl;
    }
}
//Dateien schließen
Niveaus.close();
Zustandselemente.close();
Erwartungselemente.close();
//Niveau zurückgeben
return Niveau;
}

int Zustandselement(string a, string b, string Zufallselemente_str, int Ordnung_0)
{
    int Zufallselement = 0;

```

```

int Zustand_abs = 0;
//Dateinamen zuschneiden
string Dateien = Zufallselemente_str + "_" + a + "_" + b + ".txt";
//Datei zum schreiben öffnen
ofstream Zufallselemente(Dateien.c_str(), ios::out);
//Basismatrizen erstellen und schreiben
for(int a=0; a < (Ordnung_0); a++) // Zeilen
{
    //Beginnende Null
    Zufallselemente << "0";
    //Matrizenkopf der Basismatrix
    for(int Spalten_Basismatrix = 1; Spalten_Basismatrix <= (Ordnung_0); Spalten_Basismatrix++)
    {
        Zufallselemente.width(4);
        Zufallselemente << Spalten_Basismatrix;
    }
    // Spalten
    for(int b=0; b < (Ordnung_0); b++)
    {
        //Zeilenindex Basismatrix
        Zufallselemente << endl;
        Zufallselemente << b + 1;
        Zufallselemente.width(4);

        //Zeilen
        for(int c = 0; c < (Ordnung_0); c++)
        {
            // Zufallselement berechnen
            Zufallselement = int( rand() + rand() + rand() )%3 - 1;
            // absoluten Zustand berechnen
            Zustand_abs += Zufallselement;
            // Zufallselemente schreiben
            Zufallselemente.width(4);
            Zufallselemente << Zufallselement;
        }
    }
    //Zwei Zeilen frei
    Zufallselemente << endl << endl;
}
//Datei schließen
Zufallselemente.close();
//relatives Zustandsniveau zurückgeben
return Zustand_abs;
}

```

## Das Beispiel: Die Simulation eines Aktienkurses

Über den trinären Zufallsgenerator wird der Chart eines Aktienkurses simuliert. In der Basismatrix werden die täglichen Entscheidungen von 1000 Händlern simuliert. Die Handelsmöglichkeiten (Entweder ‚Kauf‘ oder ‚Handel‘ oder ‚Verkauf‘) definieren die Zufallselemente  $X = (1, 0, -1)$ . Nach dem vordefinierten Handelsprinzip (Handel = Kauf + Verkauf) entsteht die Bewertung von Angebot und Nachfrage. Die relativen Zustandselemente der  $12 \times 31$  Zustandsmatrix beschreiben dadurch den Markt der Aktie und definieren die Tagesrenditen.

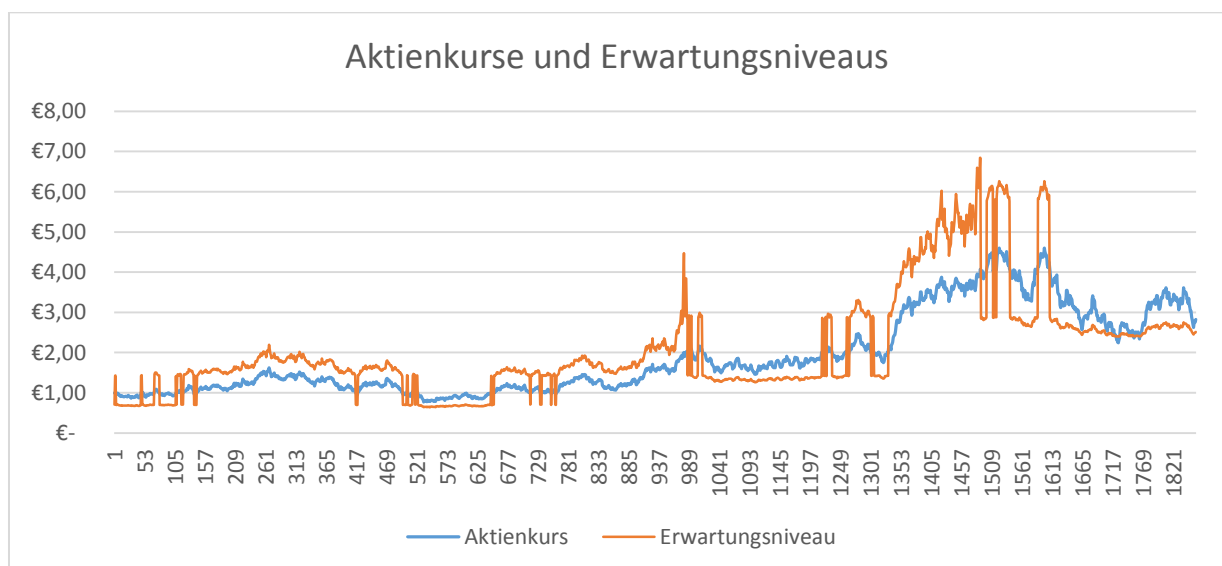
	0	1	2	3	4	5	6	...
1	-2,60%	-4,70%	-2,50%	-1,50%	0,80%	-1,40%	0,60%	
2	-0,10%	0%	-1,10%	-3,20%	-4,60%	-1,60%	4,30%	
3	-4,60%	0,70%	-4,10%	4,50%	0,70%	1,50%	-0,80%	
4	4,50%	0%	0,80%	0,30%	3,50%	1,70%	4,40%	
5	-1,10%	-2,70%	0,80%	0,50%	2,60%	-4,20%	2,40%	
6	-0,80%	2,10%	0,60%	-4,80%	1%	-2,20%	4,80%	
7	0,20%	3,40%	2%	-5,70%	1,30%	3,80%	-1,90%	
8	-8%	-2,30%	1,30%	0,40%	3,70%	2,80%	-1,50%	
9	-5,10%	-4,20%	-2%	-1,90%	1,30%	2,20%	-1,40%	
10	-0,30%	0%	2,80%	2,30%	-2,70%	0,90%	5,80%	
11	0,30%	-0,70%	-2,20%	-3,20%	0,80%	1,30%	7,60%	
12	0,10%	0,30%	1,10%	-4%	-0,90%	-2,30%	-3,10%	

Ein Beispiel für eine Zustandsmatrix der Tagesrenditen. Die Tage sind senkrecht und die Monate waagrecht angetragen...

Die Berechnung des Kursniveaus  $K(N)$  über das absolute Zufallsniveau  $N$  und den Startkurs  $K(0)$ :

$$K_{(N)} = K_{(0)} \cdot N = K_{(0)} \cdot \prod_{(i,j)=(1,1)}^{(i,j)=(a,b)} (1 + z_{(i,j)})$$

Das absolute Zufallsniveau  $N$  bildet bei Multiplikation mit dem Startkurs  $K(0)$  den Aktienkurs ab. Die simulierten Aktienkurse bestimmen die Erwartungsniveaus.



Der Aktienkurs beträgt als Startkurs das Niveau von 1,00€.