



# Vektorfolgen



## *I. Autor*

Diese Abhandlung wurde von mir, Dieter Käppel wohnhaft im Lanzenweg 11 in 90455 Nürnberg verfaßt. Meine Telefonnummer ist 0911-884857, meine Emailadresse tires@gmx.de. Ich studiere derzeit Informatik an der Georg Simon Ohm Fachhochschule, nachdem ich freiberuflich in der Branche der Informationstechnologie beschäftigt war.

## *II. Rechte*

Ich behalte mir das Eigentum am geistigen Inhalt dieses Dokumentes nach dem Urhaberschutzgesetz vor. Ich erlaube die Vervielfältigung dieses Schriftstücks, sowie die nichtkommerzielle Verwertung zu Lern- und Versuchszwecken. Wer gewinnbringenden Nutzen aus dem Inhalt dieses Werkes ziehen möchte, muß dies von mir unter obiger Adresse genehmigen lassen. Ich übernehme keine Verantwortung für Schäden, welche direkt oder indirekt durch den Inhalt dieses Schriftstücks verursacht werden könnten, auch nicht wenn es sich um fehlerhaften Inhalt oder unsachgemäße Benutzung handelt. Die Benutzung der vorgestellten Algorithmen und mathematischen Arbeitsweisen erfolgt auf eigene Gefahr.

## *III. Thema*

Gegenstand der Betrachtung sollen Folgen über der Menge der Vektoren sein.



#### *IV. Motivation*

Im Bereich der Computeranwendungen ist es zwar möglich, die vielfältigen mathematischen Funktionen zu verwenden. In der Synthese und Simulation können Programme eingegeben werden, welche Funktionen entweder aus der Fließkommaeinheit des Systems oder aus Funktionsbibliotheken aufrufen. Für die Simulation sind diese Funktionsaufrufe zum Teil gar nicht notwendig und auch zu langsam. Aber gerade in der Analyse, wo es darum geht, für einen unbekanntem Sachverhalt passende Funktionen zu finden, ist es ein Problem mit Funktionsaufrufen zu arbeiten. Es wäre viel eleganter, wenn ein Spektrum von unterschiedlichen Funktionen durch eine numerische Konfiguration dargestellt werden könnte. Damit könnte man mathematische Sachverhalte kürzer und effizienter handhaben.

Praktisch bedeutet dies, komplexe Systeme wie aero-, flüssigkeits-, feld-dynamische und andere nichtlineare Vorgänge vorauszusagen, sowie die Kommunikation zwischen Mensch und Maschine zu unterstützen. Ziel ist es, dem Menschen automatisierbare Arbeit bei Recherchen und Analysen abzunehmen und durch ein System zu ersetzen, welches mehr Fakten einbeziehen kann, als einfache statistische Betrachtungen.

#### *V. Einleitung*

In der folgenden Abhandlung soll Schritt für Schritt, begleitet durch Beweise, ein Verfahren entwickelt werden, daß eine große Zahl von darstellbaren Funktionen abdeckt. Erreicht wird das, indem ein Vektor sukzessive<sup>(3)</sup> mit einer Matrix multipliziert wird. Es wird erklärt, auf welchen Überlegungen dieses Verfahren basiert und wieso es erfolgreich arbeitet. Es wurde darauf geachtet, daß der Schreibstil wissenschaftlich ist, ohne den Text jedoch mit Fremdwörtern, selten benutzten Symbolen oder vielen Formeln zu überladen. Die Abfolge des Textes ist linear und es wird zum Verständnis eine Bildung vorausgesetzt, die in etwa dem Grundstudium der Informatik entspricht.



## VI. *Inhalt*

1. *Folgen einfacher Zahlen*
2. *Binäre Folgen*
3. *Entropie*
4. *Vektorfolgen*



## VII. Hauptteil

### 1. Folgen einfacher Zahlen

Folgen über der Menge der natürlichen oder reellen Zahlen sind ein ausgiebig diskutierte Themen<sup>(1)</sup>. Eine Folgevariable wird mit dem Startwert initialisiert. Aus dem Startwert wird mit Hilfe der Folgefunktion das nächste Glied berechnet. Aus diesem wiederum das nächste und so fort. Es sind damit einige Funktionen darstellbar:

Startwert	Folge	Ersatzfunktion	Werte
$x_0 = 0$	$x_n = x_{n-1}$	$x_n = 0$	0, 0, 0, 0, 0, 0, 0, 0, 0, ...
$x_0 = 0$	$x_n = x_{n-1} + 1$	$x_n = n$	0, 1, 2, 3, 4, 5, 6, 7, 8, ...
$x_0 = 0$	$x_n = x_{n-1} + 2n - 1$	$x_n = n^2$	0, 1, 4, 9, 16, 25, 36, ...
$x_0 = 0$	$x_n = x_{n-1} + 3n^2 + 3n + 1$	$x_n = n^3$	0, 1, 8, 27, 64, 125, 216, ...
$x_0 = 1$	$x_n = 2x_{n-1}$	$x_n = 2^n$	1, 2, 4, 8, 16, 32, 64, 128, ...
$x_0 = 1$	$x_n = -x_{n-1}$	$x_n = (-1)^n$	1, -1, 1, -1, 1, -1, 1, -1, ...
$x_0 = 1$	$x_n = nx_{n-1}$	$x_n = n!$	1, 1, 2, 6, 24, 120, 720, 5040, ...
$x_0 = 2$	$x_n = (x_{n-1})^2$	$x_n = 2^n$	2, 4, 16, 256, 65536, ...
$x_0 = 0$	$x_n = 2^{x_{n-1}}$	$x_n = ?$	0, 1, 2, 4, 16, 65536, ...

Abb. 1a

Im Allgemeinen können Folgen Summen, Produkte und höhere Funktionen darstellen. Interessant sind dabei Summen, weil bei der Summe das Integral d.h. die Stammfunktion der Folgefunktion berechnet wird. Die Folgefunktion gibt bei jedem Folgeschritt die Differenz zwischen zwei Werten der Ersatzfunktion an. Dies wird klar, wenn man die Folgefunktion umstellt:

$$x_n = x_{n-1} + 2n - 1 \Leftrightarrow s_n = x_n - x_{n-1} = 2n - 1$$

Abb. 1b

Umgekehrt erhält man für eine gegebene Ersatzfunktion die gewünschte Folgefunktion. Beispielsweise errechnet man für die Funktion  $x_n = n^3$ :

$$s_n = (n+1)^3 - n^3 = n^3 + 3n^2 + 3n + 1 - n^3 = 3n^2 + 3n + 1$$

Abb. 1c

Das Produkt kann analog zur Summe betrachtet werden, wenn man statt der Differenz den Quotienten zweier aufeinanderfolgender Glieder betrachtet:

$$x_n = 2x_{n-1} \Leftrightarrow q_n = \frac{x_n}{x_{n-1}} = 2$$

Abb. 1d

Auch hier kann man den Vorgang umkehren, damit man die Folgefunktion erhält:

$$q_n = \frac{(n+1)!}{n!} = n$$

Abb. 1e

Man stößt dabei unter anderem auf Potenzierung und Fakultät. Benutzt man als Folgeglieder noch höhere Funktionen als Produktbildung, so entstehen wiederum noch höhere Funktionen, wie z.B. die Tetration<sup>(2)</sup>. Es ist zu beachten, daß nicht alle Folgen notwendigerweise divergieren müssen. Die Folgen können auch konvergieren oder alternieren. Jedoch macht es wenig Sinn, sich in der Folgehierarchie der immer höheren Funktionen weiter nach oben zu arbeiten, denn die Häufigkeit des Vorkommens in der Natur ist umgekehrt zu ihrer Ordnung.

## 2. Binäre Folgen

Ist die Grundmenge, über der die Folgen arbeiten, die Menge der natürlichen Zahlen, sind eine unendliche Menge von Funktionen darstellbar. Da es aber nicht so sehr um die qualitative Vielfalt der möglichen Funktionen geht, sondern eher um die kombinatorische Vielfalt, sollen vorübergehend die Funktionen über dem binären Raum betrachtet werden. Die Menge der Funktionen über dem binären Raum haben den Vorteil, daß ihre Anzahl endlich ist und damit eine vollständige Betrachtung möglich ist.

Geht man vom Konstruktionsprinzip der Folgen aus Kapitel 1 aus, nämlich daß aus dem vorherigen Glied der Folge das nächste Glied entsteht, ergibt sich diese Reihe binärer Folgen:

Startwert	Folge	Ersatzfunktion	Werte
$x_0 := 0$	$x_n := 0$	$x_n = 0$	0000000000000000...
$x_0 := 1$	$x_n := 0$	$x_n = 1$	1000000000000000...
$x_0 := 0$	$x_n := x_{n-1}$	$x_n = (n \bmod 2)$	0101010101010101...
$x_0 := 1$	$x_n := x_{n-1}$	$x_n = \neg(n \bmod 2)$	1010101010101010...
$x_0 := 0$	$x_n := \neg x_{n-1}$	$x_n = 0$	0000000000000000...
$x_0 := 1$	$x_n := \neg x_{n-1}$	$x_n = 1$	1111111111111111...
$x_0 := 0$	$x_n := 1$	$x_n = 1$	0111111111111111...
$x_0 := 1$	$x_n := 1$	$x_n = 1$	1111111111111111...

Abb. 2a



Sachverhalt eines abklingenden Impulses, also eines Ereignisses, dessen Wirkung im Verlauf der Zeit nachläßt.

### 3. Entropie

Die Betrachtung der Entropie im Zusammenhang mit den binären Folgen führt vor allem bei den Folgen höherer Ordnung zu dem Schluß, daß quasineichtdeterministische Reihen sehr wenig Sinn machen. Diese fluktuierenden Reihen mit großen Perioden wirken sich in der Analyse so aus, daß sie für die Identifikation komplexer Zusammenhänge sinnvoll sind, sich jedoch störend auf die Identifikation einfacher Zusammenhänge auswirken.

Die entropische Betrachtung führt zu einer weiteren Erkenntnis. Mißt man die Vollständigkeit eines Systems an dem Maße, indem die Energie in ihm konstant bleibt, motiviert sich die Forderung mehrere Folgen parallel laufen zu lassen und deren Energiehaushalt konstant zu halten. Somit geht keine Energie bzw. Information verloren, die einmal in das System eingegeben wurde. Sind Informationen inhärent, löschen sie sich im Mittel aus. Sind die Informationen kohärent, verstärken sie sich. Neu an dieser Betrachtung ist, daß nicht nur Kohärenz im Sinne der Wellen- oder Impulstheorie<sup>(5)</sup> verstanden wird, sondern im Sinne jeder Induktion, welche von der Folge abgebildet werden kann. Es sind vielfältige Arten von Induktionen bekannt, doch indem gerade das System die Induktion vollzieht erleichtert es dem Benutzer die Arbeit der Analyse.

Mit Hilfe der Energieerhaltung löst man darüber hinaus ein Problem, welches die statistische Analyse aufwirft, nämlich daß die gewonnenen Informationen unvollständig sind. Dies führt dazu, daß die Folgen divergieren weil die Energie im System über alle Maße wächst, oder daß die Folgen konvergieren und dann keine neuen Informationen mehr liefern weil die Energie aus dem System entwichen ist. Durch einen stetigen Energiehaushalt wird dieses Problem beseitigt, ohne jedoch die Reihe in ihrer Funktionalität zu beschneiden. Es ist immer noch möglich, daß sich Energie in einer Variable konzentriert oder daß sich eine Energiekonzentration wieder auflöst.

Die Energie des Vektors ist konstant, wenn gilt:

$$E^2 = x_{0,n}^2 + x_{1,n}^2 + \dots + x_{k-1,n}^2 + x_{k,n}^2$$

Die Praktische Forderung der Energieerhaltung

#### 4. Vektorfolgen

Im Folgenden soll die Summen- und Produktbildung als ausreichend gelten, in erster Linie um die Problematik zu vereinfachen. Aus bereits genannten Gründen ist dies für die meisten praktischen Fälle auch ausreichen. Vielmehr soll sich das Interesse darauf richten, mehr als nur eine einzige Folgevariable zu betrachten. Es sollen Folgen eines Vektors anstatt Folgen einer Variablen betrachtet werden. Die Summenbetrachtung weist keinen Unterschied zur Variablenfolge auf, denn jede Komponente des Vektors wird wieder auf sich selbst abgebildet:

$$\vec{v} := \begin{pmatrix} x_{0,n} \\ x_{1,n} \\ \vdots \\ x_{k-1,n} \\ x_{k,n} \end{pmatrix} := \begin{pmatrix} x_{0,n-1} \\ x_{1,n-1} \\ \vdots \\ x_{k-1,n-1} \\ x_{k,n-1} \end{pmatrix} + \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{k-1} \\ a_k \end{pmatrix}$$

Abb. 4a

Die Vektorkomponenten blieben getrennt und können somit keine neuen Funktionen erzeugen. Das System ist zerlegbar wie folgt:

$$x_{0,n} = x_{0,n-1} + a_1 \wedge x_{1,n} = x_{1,n-1} + a_1 \wedge \dots \wedge x_{k-1,n} = x_{k-1,n-1} + a_{k-1} \wedge x_{k,n} = x_{k,n-1} + a_k$$

Abb. 4b

Somit ist bewiesen, daß die Vektorsummenfolge durch die Variablensummenfolge darstellbar ist und damit ist auch keine weitere Betrachtung mehr notwendig. Anders hingegen verhält sich das Produkt. Um ein Produkt zu bilden, muß eine Matrix eingeführt werden. Diese sorgt für eine Vermischung der Vektorkomponenten und damit sind die KomponentenvARIABLEN nicht mehr trennbar:

$$\vec{v}_n = \vec{v}_{n-1}^T \times M$$

Abb. 4c

Bei erneutem Einsetzen, ist der Vektor immer zu transponieren. Dadurch erreicht man, daß der Vektor immer die gleiche Form beibehält. Dies steht in keinem Zusammenhang mit der Funktionalität, sondern ergibt sich lediglich aus der Definition der Matrizenmultiplikation. Bei der Implementation durch ein Computerprogramm fällt diese Transformation durch geeignete Arrays weg.

$$\begin{pmatrix} x_{0,n} \\ x_{1,n} \\ \vdots \\ x_{k-1,n} \\ x_{k,n} \end{pmatrix} = \begin{pmatrix} x_{0,n-1} & x_{1,n-1} & \cdots & x_{k-1,n-1} & x_{k,n-1} \end{pmatrix} \times \begin{pmatrix} m_{0,0} & m_{1,0} & \cdots & m_{k-1,0} & m_{k,0} \\ m_{0,1} & m_{1,1} & \cdots & m_{k-1,1} & m_{k,1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ m_{0,k-1} & m_{1,k-1} & \cdots & m_{k-1,k-1} & m_{k,k-1} \\ m_{0,k} & m_{1,k} & \cdots & m_{k-1,k} & m_{k,k} \end{pmatrix}$$

Abb. 4d

Mit Hilfe dieser Form sind noch weitere Funktionen als Folgen darstellbar. Durch Matrizenmultiplikation sind aber auch Funktionen darstellbar, welche lediglich auf einer Summe basieren. Dies wird erreicht, indem eine Komponente des Vektors konstant gehalten wird.

Startwert	Folge	Ersatzfunktion
$\vec{v}_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\vec{v}_n = \vec{v}_{n-1}^T \times \begin{pmatrix} \cos \mathbf{j} & -\sin \mathbf{j} \\ \sin \mathbf{j} & \cos \mathbf{j} \end{pmatrix}$	$\vec{v}_n = \begin{pmatrix} \cos \mathbf{j} \\ \sin \mathbf{j} \end{pmatrix}$
$\vec{v}_0 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\vec{v}_n = \vec{v}_{n-1}^T \times \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$	$\vec{v}_n = \begin{pmatrix} n \\ 1 \end{pmatrix}$

Abb. 4e

## VIII. *Legende*

- (1) Heyberg Vachenauer "Höhere Mathematik 1", 4. Auflage Springer Verlag S.88ff ISBN 3-540-63688-9
- (2) Tetration ist die n-fache Wiederholung der Potenzierung  $\underbrace{a^{a \cdot \dots \cdot a}}_{n\text{-mal}}$ .
- (3) sukzessiv: [lat.], allmählich, nach und nach.
- (4) Der Sourcecode des Programms, welcher die Folgen auf ihre Unterschiedlichkeit untersucht:

```
#define BITS 2
#define FILTER

#include <iostream.h>

template <int bits> class BinaryFunction {
private:
    unsigned long int signature;

public:
    BinaryFunction(int s) :signature(s) {}

    bool operator () (int number) const {return
        (signature>>number)&1;}
};

unsigned long int render(
    const BinaryFunction<BITS> &bf,
    unsigned long int x,
    unsigned int offset,
    bool visible)
{
    unsigned int i,y=0;
    for (i=0;i<offset;++i) {
#ifdef FILTER
        cout << (x&1);
#endif
        x=(x|(bf(x)<<BITS))>>1;
    }
    for (i=0;i<2<<BITS;++i) {
        if (visible) cout << (x&1);
        y=(y<<1)|(x&1);
        x=(x|(bf(x)<<BITS))>>1;
    }
    if (visible) {
        for (i=0;i<2<<BITS;++i) {
            cout << (x&1);
            x=(x|(bf(x)<<BITS))>>1;
        }
        cout << endl;
    }
    return y;
}
```

```

int main() {
#ifdef FILTER
    unsigned long int memory[1024];
    unsigned long int j,v,y;
#endif
    unsigned long int upper=0,i,s;

    for (i=0;i<1<<(1<<BITS);++i) {                // Alle Binären
Funktionen
        BinaryFunction<BITS> bf(i);

        for (s=0;s<1<<BITS;++s) {
#ifdef FILTER
            for (v=0;v<BITS<<2;++v) {
                y=render(bf,s,((BITS+1)<<1)+v,false);
                for (j=0;j<upper;++j)
                    if (memory[j]==y ||
memory[j]==(y^((1<<(2<<BITS))-1))) break;
                    if (j<upper) break;
                }
                if (j==upper) {
                    memory[upper++]=y;
#else
                    ++upper;
#endif
                cout << upper << ".\t";
                render(bf,s,(BITS+1)<<1,true);
#ifdef FILTER
            }
#endif
        }
    }
    return 0;
}

```

- (5) Wellen- und Impulstheoretische Betrachtungen werden durch die Fourier- und Laplacetransformation berechnet. Heyberg Vachenaer "Höhere Mathematik 2", 2. Auflage Springer Verlag S.285ff, ISBN 3-540-62398-1



## *IX. Reflexion*

Bestimmte Probleme werden sich wohl für immer unserer Erkenntnis entziehen. Trotzdem erleichtert die Darstellungsweise eines bestimmten Funktionsspektrums als numerische Konfiguration sicher in vielen Anwendungsgebieten die Arbeit. Vor allem im Zusammenhang mit Computer, welche in sehr kurzer Zeit mit einer großen Menge von Zahlen umgehen können bietet sich an, einem derartigen System Induktionen übernehmen zu lassen.

# X. Index

<b>A</b>	
Abfolge des Textes .....	3
abklingenden Impulses .....	9
alternieren .....	7
Arten von Induktionen.....	9
automatisierbare Arbeit .....	3
<i>Autor</i> .....	1

<b>B</b>	
Bildung.....	3
<i>Binäre Folgen</i> .....	5, 7
binären Raum.....	7

<b>C</b>	
Computeranwendungen.....	3
Computerprogramm.....	8, 10, 12

<b>D</b>	
darstellbaren Funktionen.....	3
Darstellbarkeit .....	6
Daseinsberechtigung.....	8
Differenz .....	6
divergieren .....	7, 9

<b>E</b>	
Eigentum.....	1
einfacher Zusammenhänge.....	9
<i>Einleitung</i> .....	3
Emailadresse.....	1
endlich .....	7
Energiehaushalt.....	9
Energiekonzentration .....	9
Entropie .....	9
Ereignisses .....	9
Erkenntnis .....	9, 14
Ersatzfunktion .....	6, 7, 11

<b>F</b>	
Fakultät.....	7
fehlerhaften Inhalt .....	1
fluktuierenden Reihen.....	9
Folge .....	6, 7, 11
Folgefunktion .....	6, 7
Folgen einer Variablen .....	10

Folgen eines Vektors.....	10
<i>Folgen einfacher Zahlen</i> .....	5, 6
Folgen natürlicher und reeller Zahlen.....	9
Folgevariable .....	6, 10
Fouriertransformation .....	13
Funktionsaufrufe.....	3
Funktionsspektrums .....	14

<b>G</b>	
Gefahr.....	1
Gegenstand der Betrachtung.....	1
genehmigen.....	1
Georg Simon Ohm Fachhochschule.....	1
gewinnbringenden Nutzen .....	1
Glied .....	6, 7, 8
Grundmenge .....	7

<b>H</b>	
Häufigkeit .....	7
<i>Hauptteil</i> .....	6
Heyberg .....	12, 13
höhere Funktionen.....	6, 7
Höhere Mathematik 1 .....	12, 13

<b>I</b>	
Identifikation .....	9
<i>Index</i> .....	15
Induktion.....	9
Informatik .....	1, 3
Informationstechnologie.....	1
<i>Inhalt</i> .....	1, 5
inhärent.....	9
initialisiert .....	6
Integral.....	6
Inversion.....	8

<b>K</b>	
kohärent.....	9
kombinatorische Vielfalt .....	7
komplexe Systeme .....	3
komplexer Zusammenhänge.....	9
konstant .....	9, 11
Konstruktionsprinzip der Folgen .....	7
konvergieren .....	7, 9

<b>L</b>	Springer Verlag.....	12, 13
Laplacetransformation.....	Stammfunktion.....	6
<i>Legende</i> .....	Startwert.....	6, 7, 11
<b>M</b>	statistische Analyse.....	9
mathematischen Funktionen.....	statistische Betrachtungen.....	3
Matrix.....	störend.....	9
Matrizenmultiplikation.....	Summen.....	6, 10
Menge der Funktionen.....	Synthese.....	3
möglichen Funktionen.....	<b>T</b>	
<i>Motivation</i> .....	Telefonnummer.....	1
<b>N</b>	Tetration.....	7, 12
natürlichen Zahlen.....	<i>Thema</i> .....	1
nichttrivialen Folgen.....	transponieren.....	10
numerische Konfiguration.....	<b>U</b>	
<b>O</b>	unbekannten Sachverhalt.....	3
Ordnung.....	unendliche Menge.....	7
<b>P</b>	unsachgemäße Benutzung.....	1
parallel.....	unvollständig.....	9
passende Funktionen.....	Urheberrechtsgesetz.....	1
Perioden.....	<b>V</b>	
Potenzierung.....	Vachenauer.....	12, 13
Problematik zu vereinfachen.....	Variablensummenfolge.....	10
Produkte.....	Varianten.....	8
Programme.....	Vektor.....	3, 10
<b>Q</b>	Vektorfolgen.....	2, 5, 10
qualitative Vielfalt.....	Vektorsummenfolge.....	10
quasinichtdeterministische Reihen.....	Verantwortung für Schäden.....	1
Quotienten.....	Vergangenheit der Reihe.....	8
<b>R</b>	Vermischung.....	10
Recherchen.....	Verschiebung.....	8
<i>Rechte</i> .....	Vervielfältigung.....	1
<i>Reflexion</i> .....	Verwertung.....	1
<b>S</b>	vollständige Betrachtung.....	7
sehr hoher Ordnung.....	vollständige Reihe.....	8
Selbstabbildung.....	Vollständigkeit.....	8, 9
Simulation.....	Vorgang umkehren.....	7
Sinn.....	<b>W</b>	
Sourcecode.....	welche k-stellige Binärfunktionen.....	8
	Wirkung.....	9
	<b>Z</b>	
	zerlegbar.....	10

