

2 Digitale Darstellung von Mediendaten

2.1 Gesetze von Fechner und Weber

2.2 ASCII, DIN 66003, ISO 646 und ISO 8859

2.3 Unicode

2.4 UTF-16, UTF-8 und UTF-7

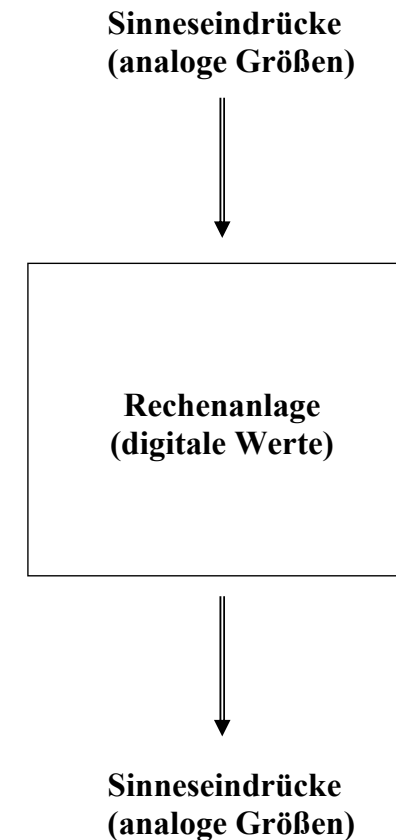
2.5 Digitalisierung

2.6 Fourier-Darstellung

2.7 Nyquist-Theorem

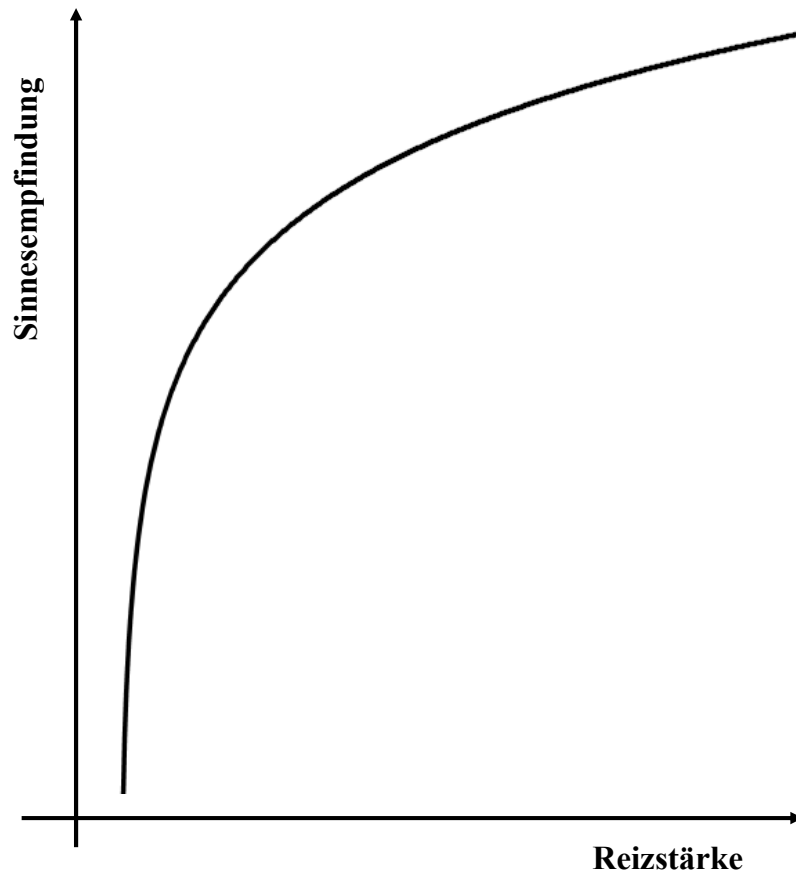
2.8 Pulscodierung

Allgemeine Situation:



Fragen: Wie wandelt man analoge Werte in digitale und zurück?
Wie viele unterschiedliche Werte sollte man berücksichtigen?

Allgemeines Schaubild zum Zusammenhang von Reizstärke und Wahrnehmung:



Beobachtungen:

Es existieren ein unterer Schwellwert und ein oberer Schwellwert der Wahrnehmung. Änderungen des Reizes im unteren Wahrnehmungsbereich werden relativ stärker empfunden als Reizänderungen im oberen Wahrnehmungsbereich.

Fechner (1850):

Die Stärke einer Sinnesempfindung ist proportional dem Logarithmus der Reizstärke.

$$S = \text{Konst} * 10 \log (R)$$

(für $R > R_0$, $R_0 = \text{Reizschwelle}$)

Weber (1834):

Das Auflösungsvermögen für Reizunterschiede ist proportional der Reizstärke.

$$\delta R = k * R$$

Die Werte für den Proportionalitätsfaktor k streuen über einen weiten Bereich. Als typische Werte können angesehen werden für

die Helligkeit	$k = 0,016,$
die Lautstärke	$k = 0,088,$
die Tonhöhe	$k = 0,003.$

Bereiche zwischen Reizschwelle und Schmerzgrenze sind

für die Helligkeit $1 : 10^{10}$,
für die Lautstärke $1 : 10^{12}$,
für die Tonhöhe $1 : 10^3$.

Für zwei gerade noch unterscheidbare Reizstärken Ra und Rb gilt:

$$R_b = (1 + k) * R_a.$$

Gemäß der Formel

$$\text{Schmerzgrenze} = (1 + k)^{\text{Stufen}} * \text{Reizschwelle}$$

berechnet man etwa

1450 unterscheidbare Helligkeitsstufen,
330 unterscheidbare Lautstärken,
2300 unterscheidbare Tonhöhen.

Bemerkung: Ein Klavier verfügt im allgemeinen über 88 Tasten und deckt einen Tonumfang von $7\frac{1}{4}$ Oktaven ab. Der niedrigste Ton liegt bei 27,5 Hz und der höchste bei 4186 Hz.

Bemerkungen zum Weber-Fechnerschen Gesetz:

- (i) Die algorithmische Beziehung des Weber-Fechnerschen Gesetzes gilt nur für den Mittelteil des Beobachtungsintervalls.
- (ii) Ursprünglich formulierte Ernst Heinrich Weber seine Beobachtungen für Gewichtsempfindungen.

Gewicht P in Gramm	Gewichtsdifferenz = $0,04 * P$	Gewichtsdifferenz = $0,08 * P$
300	612	714
500	586	701
1000	629	747
1500	638	811
2000	661	828
3000	685	839

Zahl der richtigen Fälle bei zweihändigen Gewichtsvergleichen, jeweils 1024 Versuche.

Zahlen aus G. Th. Fechner: Elemente der Psychophysik, Leipzig (1889).

- (iii) S. S. Stevens formulierte 1957 ein Potenzgesetz: $S = \text{const} * R^x$; seiner Meinung nach beschreibt es die psychophysikalischen Gegebenheiten besser als das Weber-Fechnersche Gesetz.

Codierungen:

Als Code betrachtet man eine Abbildung von Zahlen in einen Zeichenraum.

Beispiel: Auszug aus Microsoft Windows Codepage 1250

33	EXCLAMATION MARK	!
34	QUOTATION MARK	"
35	NUMBER SIGN	#
⋮		
133	HORIZONTAL ELLIPSIS	⋯
134	DAGGER	†
135	DOUBLE DAGGER	‡
137	PER MILLE SIGN	‰
138	LATIN CAPITAL LETTER S WITH CARON	Š
⋮		
233	LATIN SMALL LETTER E WITH ACUTE	é
234	LATIN SMALL LETTER E WITH OGONEK	ė
235	LATIN SMALL LETTER E WITH DIAERESIS	ë
236	LATIN SMALL LETTER E WITH CARON	ě
237	LATIN SMALL LETTER I WITH ACUTE	í

Tabelle des American Standard Code for Information Interchange (ASCII):

$b_3b_2b_1b_0$	$b_6b_5b_4$	000	001	010	011	100	101	110	111
0000		NUL	DLE	SP	0	@	P	'	p
0001		SOH	DC1	!	1	A	Q	a	q
0010		STX	DC2	"	2	B	R	b	r
0011		ETX	DC3	#	3	C	S	c	s
0100		EOT	DC4	\$	4	D	T	d	t
0101		ENQ	NAK	%	5	E	U	e	u
0110		ACK	SYN	&	6	F	V	f	v
0111		BEL	ETB	'	7	G	W	g	w
1000		BS	CAN	(8	H	X	h	x
1001		HT	EM)	9	I	Y	i	y
1010		LF	SUB	*	:	J	Z	j	z
1011		VT	ESC	+	;	K	[\	{
1100		FF	FS	,	<	L]	^	
1101		CR	GS	-	=	M	^	_	~
1110		SO	RS	.	>	N	~	DEL	
1111		SI	US	/	?	O	_		

Zeichentabelle nach DIN 66003:

$b_3b_2b_1b_0$	$b_6b_5b_4$	000	001	010	011	100	101	110	111
0 0 0 0		NUL	(TC ₇) DLE	SP	0	S	P	'	p
0 0 0 1		(TC ₁) SOH	DC1	!	1	A	Q	a	q
0 0 1 0		(TC ₂) STX	DC2	"	2	B	R	b	r
0 0 1 1		(TC ₃) ETX	DC3	#	3	C	S	c	s
0 1 0 0		(TC ₄) EOT	DC4	\$	4	D	T	d	t
0 1 0 1		(TC ₅) ENQ	(TC ₈) NAK	%	5	E	U	e	u
0 1 1 0		(TC ₆) ACK	(TC ₉) SYN	&	6	F	V	f	v
0 1 1 1		BEL	(TC ₁₀) ETB	'	7	G	W	g	w
1 0 0 0		FE ₀ (BS)	CAN	(8	H	X	h	x
1 0 0 1		FE ₁ (HT)	EM)	9	I	Y	i	y
1 0 1 0		FE ₂ (LF)	SUB	*	:	J	Z	j	z
1 0 1 1		FE ₃ (VT)	ESC	+	;	K	Ä	k	ä
1 1 0 0		FE ₄ (FF)	IS ₄ (FS)	,	<	L	Ö	l	ö
1 1 0 1		FE ₅ (CR)	IS ₃ (GS)	-	=	M	Ü	m	ü
1 1 1 0		SO	IS ₂ (RS)	.	>	N	<	n	&
1 1 1 1		SI	IS ₁ (US)	/	?	O	_	o	DEL

Abkürzungen:

SOH	Start Of Heading
STX	Start Of Text
ETX	End Of Text
EOT	End Of Transmission
ENQ	Enquiry
ACK	Acknowledge
BS	Backspace
HT	Horizontal Tab
LF	Line Feed
VT	Vertical Tab
FF	Form Feed
CR	Carriage Return
SO	Shift Out
SI	Shift In
DLE	Data Link Escape
DCi	Device Control i (1 ≤ i ≤ 4)
NAK	Negative Acknowledge
SYN	Synchronous Idle (SYNC)
ETB	End Of Transmission Block
CAN	Cancel
EM	End Of Medium
SUB	Substitute
ESC	Escape
FS	File separator
GS	Group separator
RS	Record separator
US	Unit separator
DEL	Delete
TC	Transmission Control
IS	Information Separator
FE	Format Effector
SP	Space

Internationales Referenzalphabet:

Das internationale Referenzalphabet ist unter mehreren Namen bekannt, z. B. unter
 ISO 646-1972,
 ECMA-6,
 IA5 (Internationales Telegraphie Alphabet No. 5).

Bis auf die Codepositionen 35, 36, 64, 91, 92, 93, 94, 96, 123, 124, 125, 126 ist das internationale Referenzalphabet gleichwertig zu ASCII, wobei der semantische Inhalt einiger Kontrollcodes durchaus geändert wurde.

Den Positionen 35 und 36 können jeweils zwei unterschiedliche Zeichen zugeordnet werden, £ oder # bzw. \$ oder □.

Die zehn Positionen 64, 91, 92, 93, 94, 96, 123, 124, 125 und 126 sind für nationale oder anwendungsspezifische Belegungen reserviert.

Beispiele:

	64	91	92	93	94	96	123	124	125	126
US	@	[\		^	`	{		}	~
DE	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
DK	@	Æ	Ø	Å	^	`	æ	ø	å	~
ES	§	ı	Ñ	ı	^	`	°	ñ	ç	~
FR	à	°	ç	§	^	μ	é	ù	è	¨

Tabelle von ISO/IEC 8859-1, genannt Latin 1 (erster Teil):

b ₇ b ₆ b ₅ b ₄	b ₃ b ₂ b ₁ b ₀	
0000	0000	SP
0000	0001	!
0000	0010	"
0000	0011	#
0000	0100	\$
0000	0101	%
0000	0110	&
0000	0111	'
0000	1000	(
0000	1001)
0000	1010	*
0000	1011	+
0000	1100	,
0000	1101	-
0000	1110	.
0000	1111	/
0001	0000	SP
0001	0001	!
0001	0010	"
0001	0011	#
0001	0100	\$
0001	0101	%
0001	0110	&
0001	0111	'
0001	1000	(
0001	1001)
0001	1010	*
0001	1011	+
0001	1100	,
0001	1101	-
0001	1110	.
0001	1111	/
0010	0000	0
0010	0001	1
0010	0010	2
0010	0011	3
0010	0100	4
0010	0101	5
0010	0110	6
0010	0111	7
0010	1000	8
0010	1001	9
0010	1010	:
0010	1011	;
0010	1100	<
0010	1101	=
0010	1110	>
0010	1111	?
0100	0000	@
0100	0001	A
0100	0010	B
0100	0011	C
0100	0100	D
0100	0101	E
0100	0110	F
0100	0111	G
0100	1000	H
0100	1001	I
0100	1010	J
0100	1011	K
0100	1100	L
0100	1101	M
0100	1110	N
0100	1111	O
0101	0000	P
0101	0001	Q
0101	0010	R
0101	0011	S
0101	0100	T
0101	0101	U
0101	0110	V
0101	0111	W
0101	1000	X
0101	1001	Y
0101	1010	Z
0101	1011	[
0101	1100	\
0101	1101]
0101	1110	^
0101	1111	_
0110	0000	0
0110	0001	1
0110	0010	2
0110	0011	3
0110	0100	4
0110	0101	5
0110	0110	6
0110	0111	7
0110	1000	8
0110	1001	9
0110	1010	:
0110	1011	;
0110	1100	<
0110	1101	=
0110	1110	>
0110	1111	?
0111	0000	@
0111	0001	A
0111	0010	B
0111	0011	C
0111	0100	D
0111	0101	E
0111	0110	F
0111	0111	G
0111	1000	H
0111	1001	I
0111	1010	J
0111	1011	K
0111	1100	L
0111	1101	M
0111	1110	N
0111	1111	O
0111	0000	p
0111	0001	q
0111	0010	r
0111	0011	s
0111	0100	t
0111	0101	u
0111	0110	v
0111	0111	w
0111	1000	x
0111	1001	y
0111	1010	z
0111	1011	{
0111	1100	
0111	1101	}
0111	1110	~
0111	1111	~

UNICODE:

Zuweisung von Zeichen zu Codepunkten aus dem Raum

0x000000 .. 0x10FFFF.

Der Coderaum 0x0000 .. 0xFFFF, genannt Basic Multilingual Plane, umfaßt die gängigsten Zeichen.

Teilmengen:

0x0020 .. 0x007E : Druckbare ASCII-Zeichen

0x0020 .. 0x024F : Lateinische Zeichen

0x20A0 .. 0x20CF : Währungssymbole

0x2700 .. 0x27BF : Dingbats

0x4E00 .. 0x9FFF : CJK vereinheitlichte Ideogramme

0xD800 .. 0xDFFF : Reserviert für UTF-16

0xE000 .. 0xF8FF : Privater Bereich

Bemerkungen:

- (i) CJK ist Abkürzung für Chinese, Japanese, and Korean.
- (ii) Die Norm ISO-10646, Universal Multiple-Octet Coded Character Set, und Unicode werden gemeinsam weiterentwickelt. Im USC stehen 31 Bit für die Zeichenzuweisung zur Verfügung.
- (ii) Unter der Internetadresse <http://www.unicode.org> findet man die aktuelle Version des Unicode.

Beispiele aus Unicode:

Nummer	Zeichename	Zeichenbild
192	LATIN CAPITAL LETTER A WITH GRAVE	À
193	LATIN CAPITAL LETTER A WITH ACUTE	Á
913	GREEK CAPITAL LETTER ALPHA	Α
914	GREEK CAPITAL LETTER BETA	Β
8473	DOUBLE-STRUCK CAPITAL P	ℙ
8474	DOUBLE-STRUCK CAPITAL Q	ℚ
8475	SCRIPT CAPITAL R	℞
12353	HIRAGANA LETTER SMALL A	ぁ
12354	HIRAGANA LETTER A	あ
12355	HIRAGANA LETTER SMALL I	ぃ
12356	HIRAGANA LETTER I	い
19993	"namenloser Ideograph"	丙

UTF-16:

UTF-16 dient der ökonomischen Speicherung von Unicode-Text. Jedes Unicode-Zeichen lässt sich in 4 Oktetten speichern, die meisten benötigen aber höchstens zwei Oktette.

Codierregel: Sei U ein Unicode-Symbol:

Ist $U < 0x10000$, dann codiere U in zwei Oktetten und beende Codierung.

Bilde $V = U - 0x10000$.

$V \leq 0xFFFF$, wegen $U \leq 0x10FFFF$.

V lässt sich in zwei Bitfolgen à 10 Bit zerlegen:

$V = \text{yyyyyyyyyy xxxxxxxx}$.

Bilde nun ein Paar $W1, W2$ aus jeweils zwei Oktetten:

$W1 = 110110\text{yyyyyyyyyy}$, $W2 = 110111\text{xxxxxxxxxx}$.

Das Paar $W1, W2$ codiert nun U . Diese Codierung ist eindeutig.

Beispiel: CUNEIFORM NUMERIC SIGN NINE DISH,
Codepunkt: $0x1240E$.

$1240E - 10000 = 240E$
 $= 0000\ 0010\ 0100\ 0000\ 1110$,

$W1 = 1101\ 1000\ 0000\ 1001 = 0xD809$,

$W2 = 1101\ 1100\ 0000\ 1110 = 0xDC0E$,

Codepaar: $0xD809\ 0xDC0E$.

Bemerkungen zu UTF-16:

- (i) UTF-16 wird beschrieben in RFC 2781.
- (ii) Es existieren drei Formen der UTF-16-Codierung: UTF-16BE, UTF-16LE und UTF-16.
- (iii) Zur Unterscheidung, wie die Reihenfolge von UTF-16-Oktetten interpretiert werden soll, benutzt man die BOM-Kennzeichnung. (BOM = Byte Order Mark) Beginnt ein UTF-16 Text mit dem Oktett-paar $0xFE\ 0xFF$, dann handelt es sich um Text im Big Endian Format, beginnt er mit $0xFF\ 0xFE$, dann handelt es sich um Text im Little Endian Format. Das Paar $0xFEFF$ ist gleichzeitig ein Unicode-Zeichen, ZERO WIDTH NON-BREAKING SPACE, ein Unicode-Zeichen $0xFFFE$ existiert nicht.
- (iv) Bei Wandlung in UTF-8 wird jedes Surrogate-Paar als Folge von vier Oktetten dargestellt.
- (v) Darstellung des Textes ABCDEFGHIJ ... unter Windows:
als ANSI-Text: 41 42 43 45 46 47 48 ...
als Unicode-Text: FF FE 41 00 42 00 43 ...
als Unicode-Text: FE FF 00 41 00 42 00 ...
als UTF-8-Text: EF BB EF 41 42 43 44 45 ...

Bemerkung: Die Angabe "ANSI-Text" ist eine Bezeichnung aus dem Wortschatz von Microsoft Windows.

UTF-8: (UTF = Universal Transformation Format, beschrieben u. a. in RFC 3629)

Transformationsbeschreibung:

UCS-4 Bereich (sedezimal)	UTF-8 Oktett Sequenz (binär)
0000 0000 - 0000 007F	0xxxxxxx
0000 0080 - 0000 07FF	110xxxxx 10xxxxxx
0000 0800 - 0000 FFFF	1110xxxx 10xxxxxx 10xxxxxx
0001 0000 - 001F FFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx
0020 0000 - 03FF FFFF	111110xx 10xxxxxx 10xxxxxx 10xxxxxx
0400 0000 - 7FFF FFFF	10xxxxxx 1111110x 10xxxxxx 10xxxxxx 10xxxxxx

Beispiel:

Die UCS-2 Folge "A<NOT IDENTICAL TO><ALPHA>" (0041, 2262, 0391,002E) wird in UTF-8 wie folgt codiert: 41 E2 89 A2 CE 91 2E

UTF-7:

Einführung zweier Alphabete, ein Teilalphabet T von ASCII und ein 64-Zeichen-Alphabet.

Das Teilalphabet T besteht aus folgenden ASCII-Zeichen:

den Großbuchstaben A .. Z,
den Kleinbuchstaben a .. z,
den Ziffern 0 .. 9,
dem Zeichen ' (dezimal 39),
dem Zeichen ((dezimal 40),
dem Zeichen) (dezimal 41),
dem Zeichen , (dezimal 44),
dem Zeichen - (dezimal 45),
dem Zeichen . (dezimal 46),
dem Zeichen / (dezimal 47),
dem Zeichen : (dezimal 58),
dem Zeichen ? (dezimal 63).

Base64 Content-Transfer-Encoding:

Ein Beispiel:

Gegeben sei die Zeichenkette “ABCD”, sie ist 32 Bit lang.

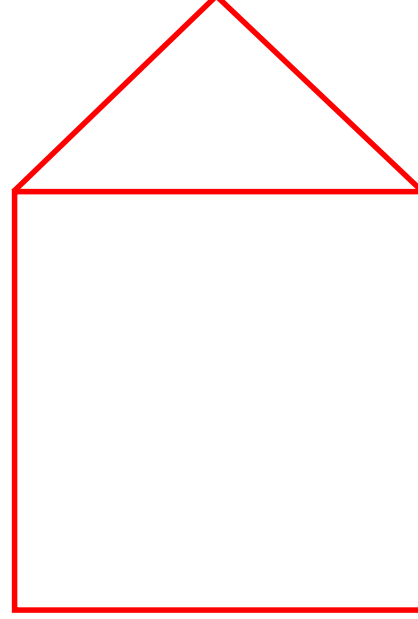
ABCD

A (65)	B (66)	C (67)	D (68)	(leer)	(leer)
01000001	01000010	01000011	01000100		
16 (Q)	20 (U)	9 (J)	3 (D)	17 (R)	0 (A) (=)
010000	010100	001001	000011	010001	000000 000000 000000

QUJDRA==

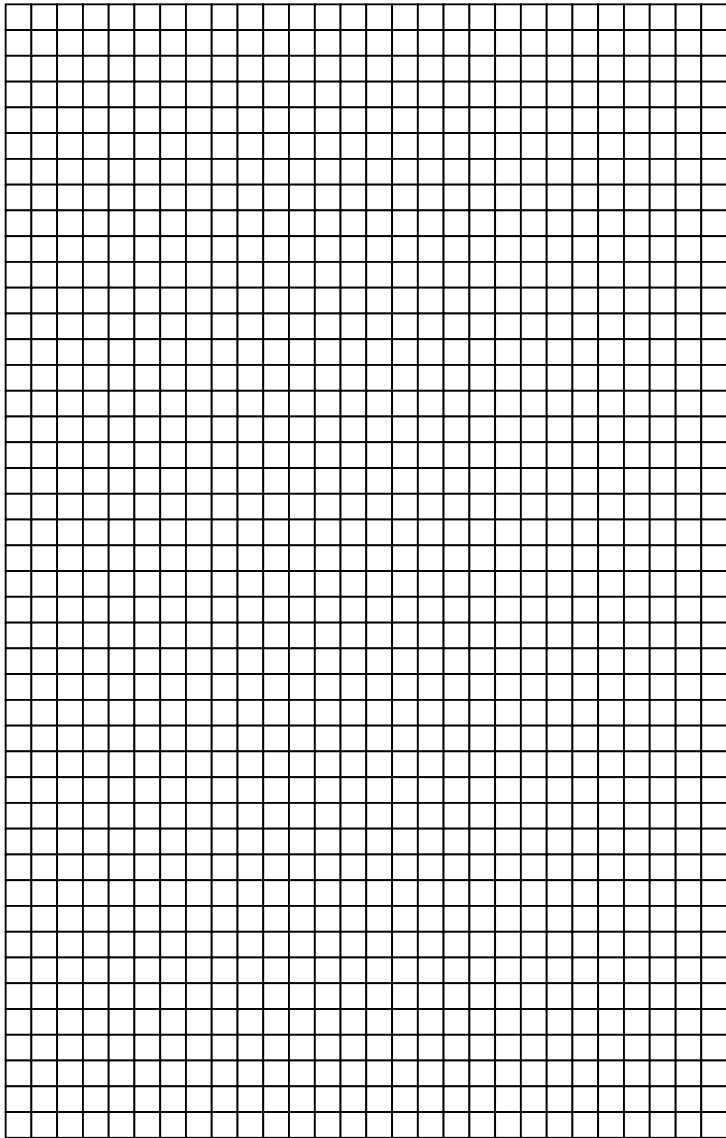
Bemerkung: In den codierten Text können auch andere Zeichen eingefügt werden. Sie werden bei der Decodierung nicht berücksichtigt. Ein Beispiel für ein hinzuzufügendes Zeichen ist das Zeilenendezeichen.

Beispiel zur Digitalisierung:

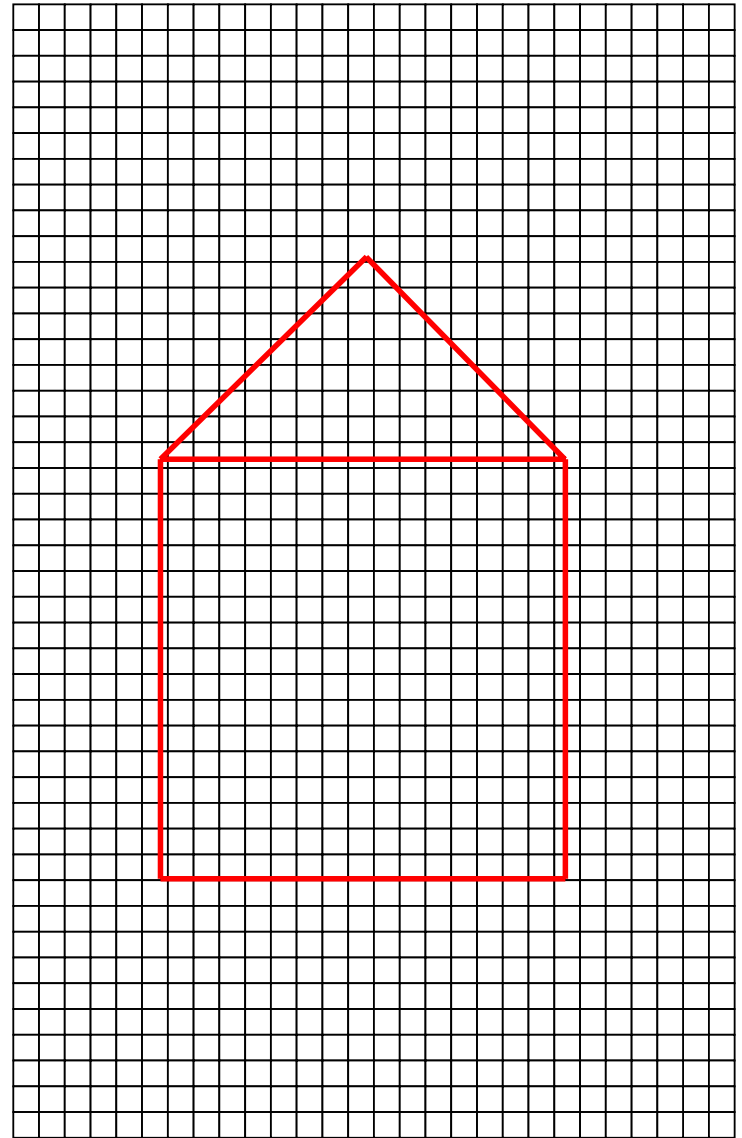


Das Haus des Nikolaus

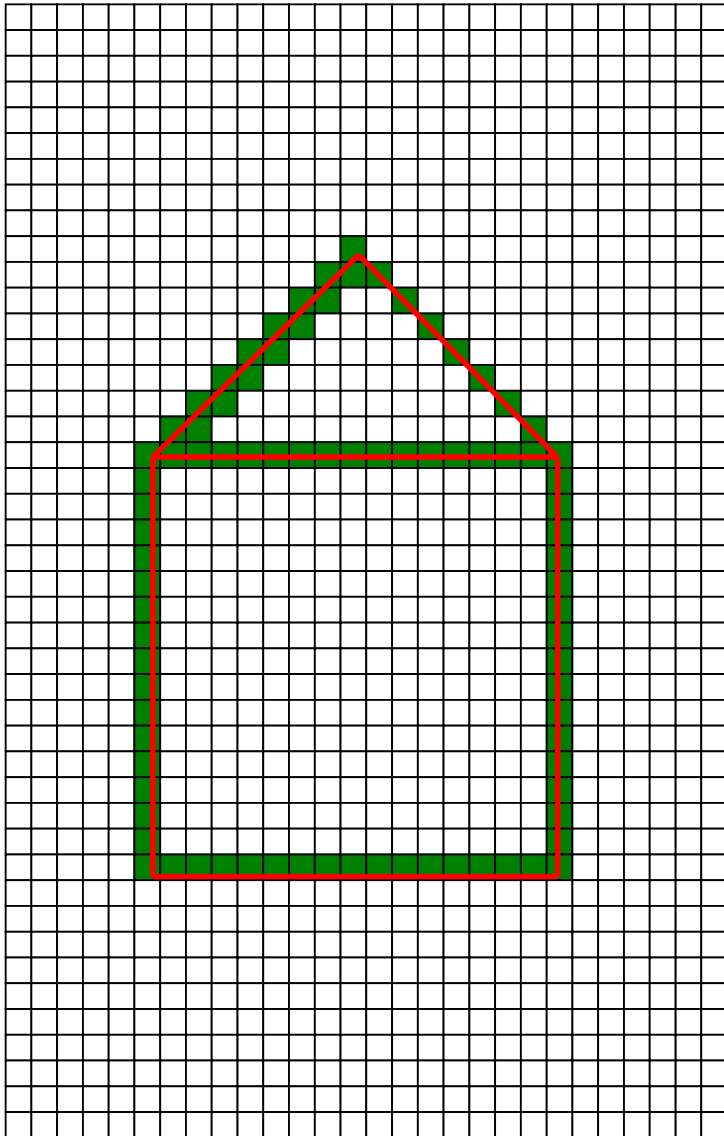
Ein Raster:



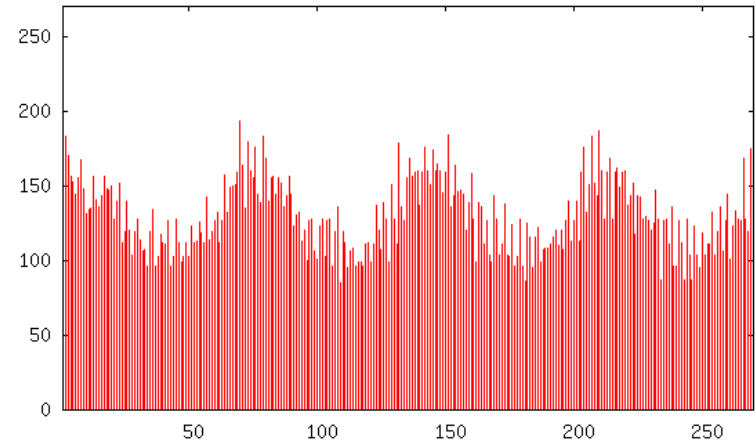
Raster mit Bild:



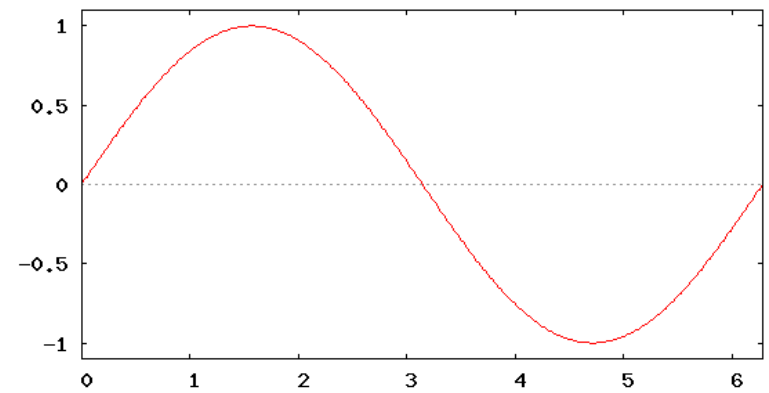
Gerastertes Bild:



Periodennahe Vorgänge:



270 fortlaufende Werte aus einer mit einer Abtastrate von 11 kHz erstellten Sounddatei.



Sinus-Funktion

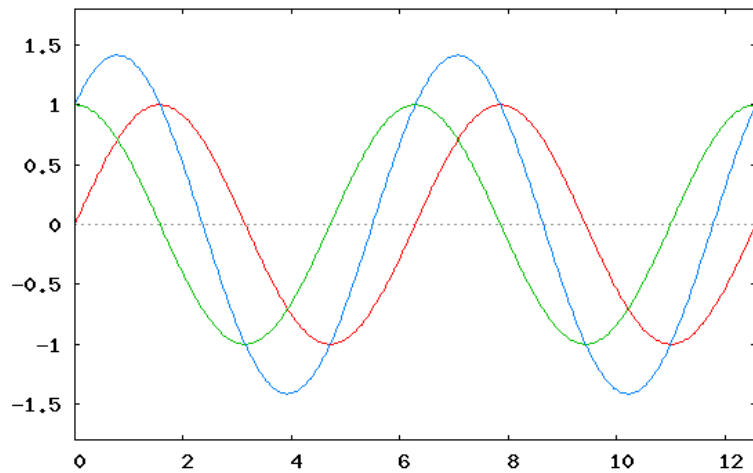


Bild der Funktionen Sinus, Kosinus, Sinus + Kosinus

Bemerkung: Viele Funktionen lassen sich darstellen als Summen von Sinus- und Kosinusfunktionen.

Beispiele sind:

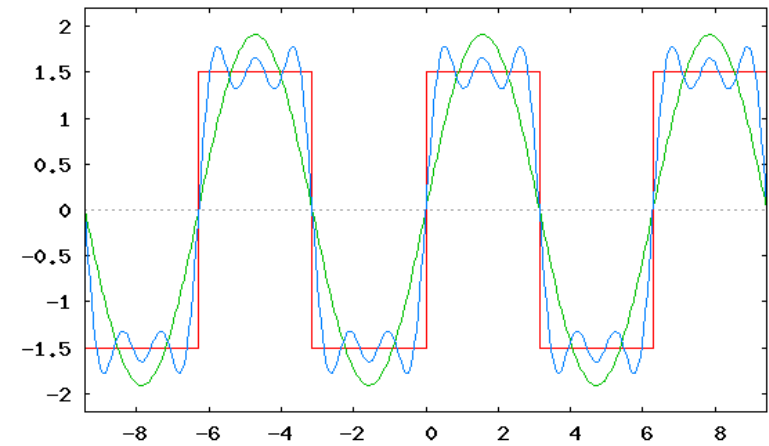
Rechteckkurve der Höhe 2*h:

$$f(x) = \frac{4 \cdot h}{\pi} \cdot (\sin(x) + \frac{1}{3} \cdot \sin(3 \cdot x) + \frac{1}{5} \cdot \sin(5 \cdot x) + \dots)$$

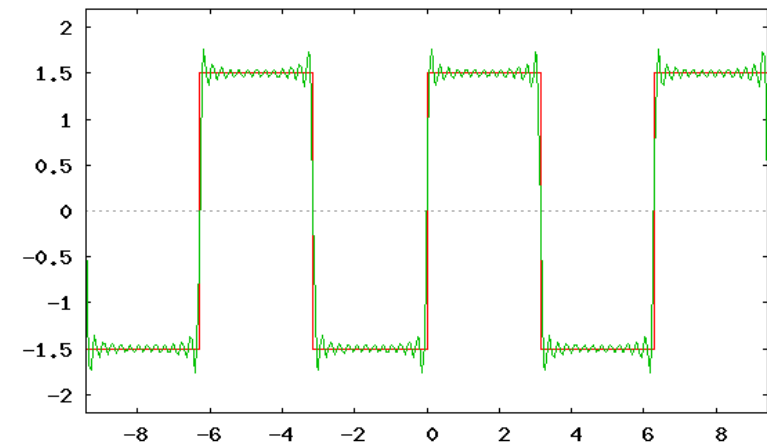
Sägezahnkurve der Höhe 2*h:

$$f(x) = -\frac{2 \cdot h}{\pi} \cdot (\sin(x) + \frac{1}{2} \cdot \sin(2 \cdot x) + \frac{1}{3} \cdot \sin(3 \cdot x) + \dots)$$

Rechteckkurve:

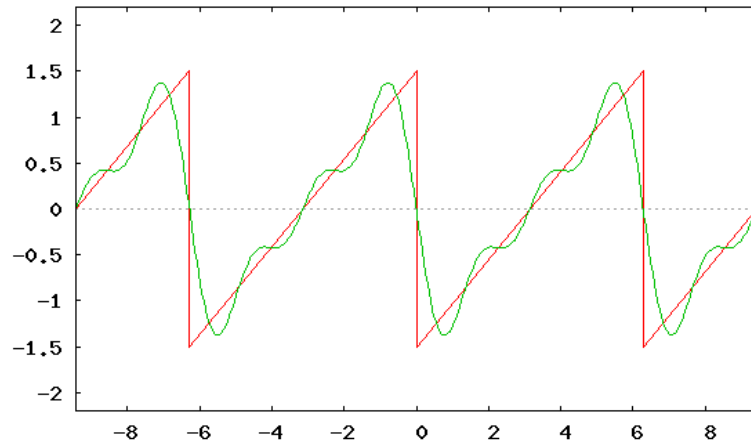


Ein- und dreigliedrige Näherung der Rechteckkurve

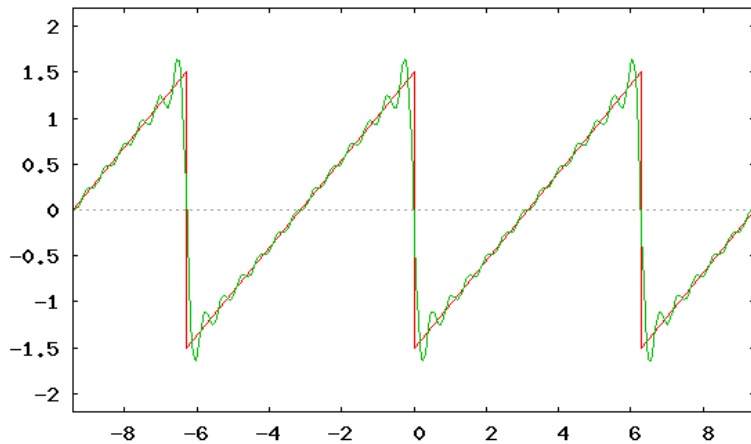


Dreizehngliedrige Näherung der Rechteckkurve, man erkennt gut das Gibbsche Phänomen.

Sägezahnkurve:



Dreigliedrige Näherung der Sägezahnkurve



Zwölfgliedrige Näherung der Sägezahnkurve

Fourier-Darstellung periodischer Funktionen:

Jede hinreichend gutmütige periodische Funktion $f(x)$ der Periode T kann durch ihre Fourierreihe $g(x)$ dargestellt werden:

$$g(x) = \frac{1}{2} * c + \sum_{i=1}^{\infty} a_i * \sin(2\pi i f x) + \sum_{i=1}^{\infty} b_i * \cos(2\pi i f x)$$

wobei $f = 1/T$ die Grundfrequenz ist.

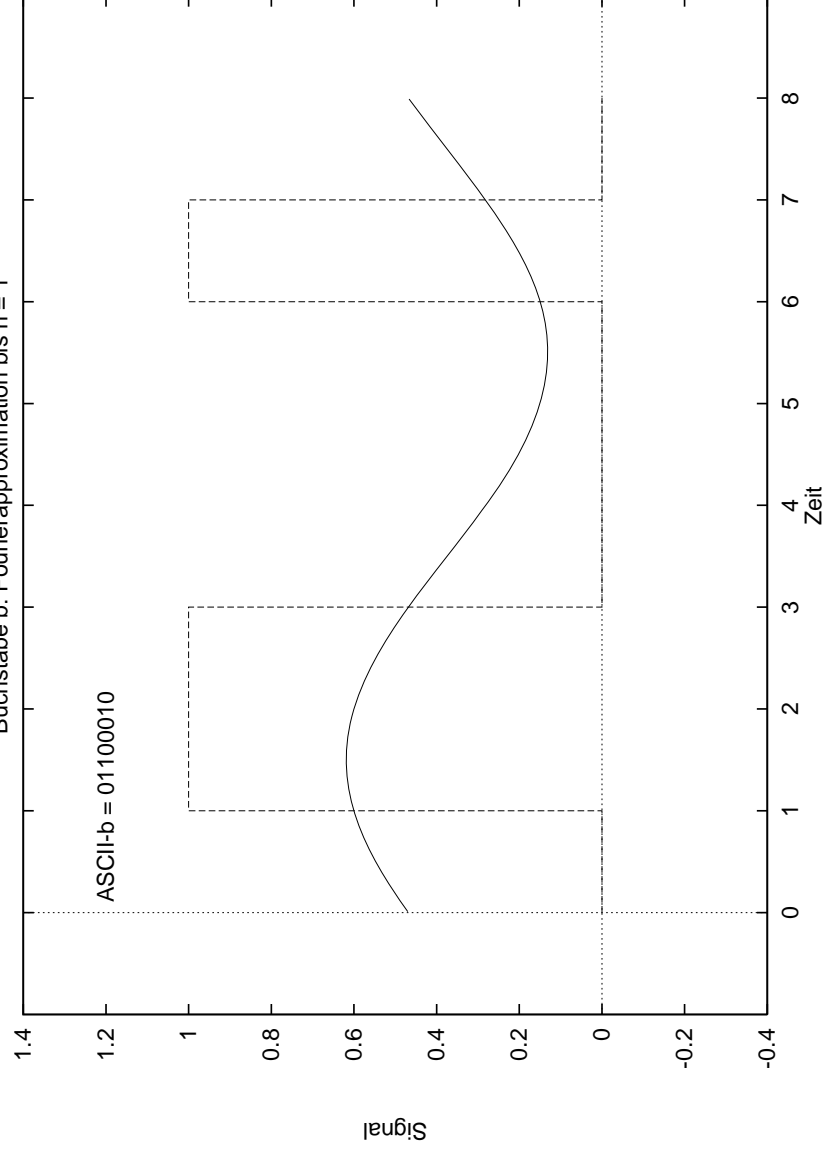
Die Fourier-Koeffizienten c , a_i , b_i ($i = 1, \dots, \infty$) berechnet man gemäß:

$$c = \frac{2}{T} * \int_0^T g(t) dt$$

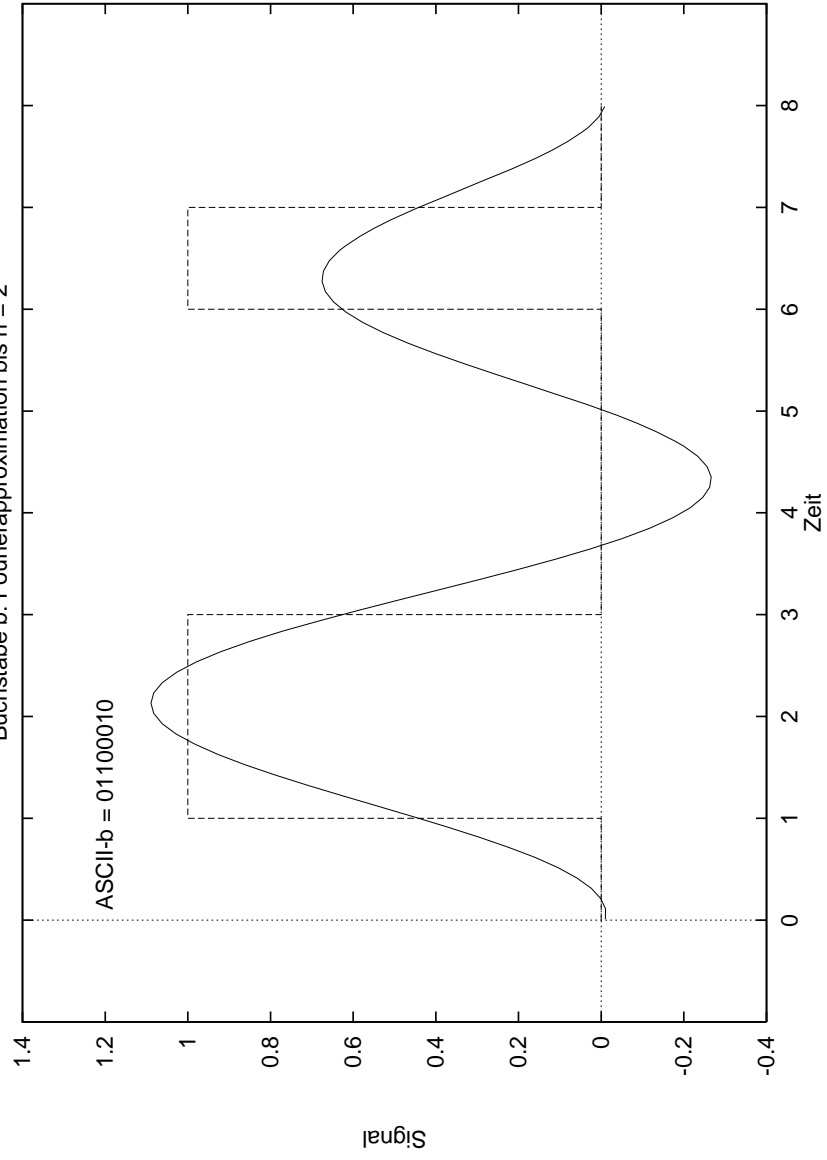
$$a_i = \frac{2}{T} * \int_0^T g(t) * \sin(2\pi i f t) dt \quad (i=1, 2, \dots, \infty)$$

$$b_i = \frac{2}{T} * \int_0^T g(t) * \cos(2\pi i f t) dt \quad (i=1, 2, \dots, \infty)$$

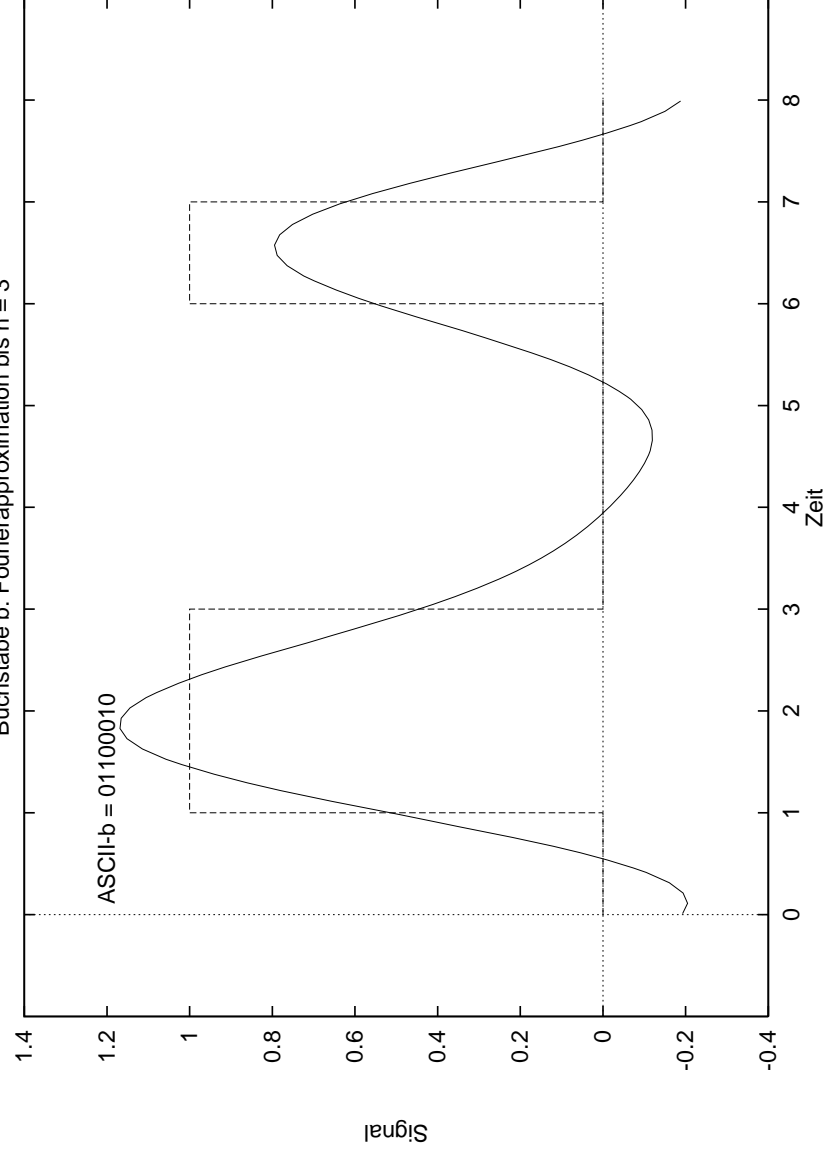
Buchstabe b: Fourierapproximation bis n = 1



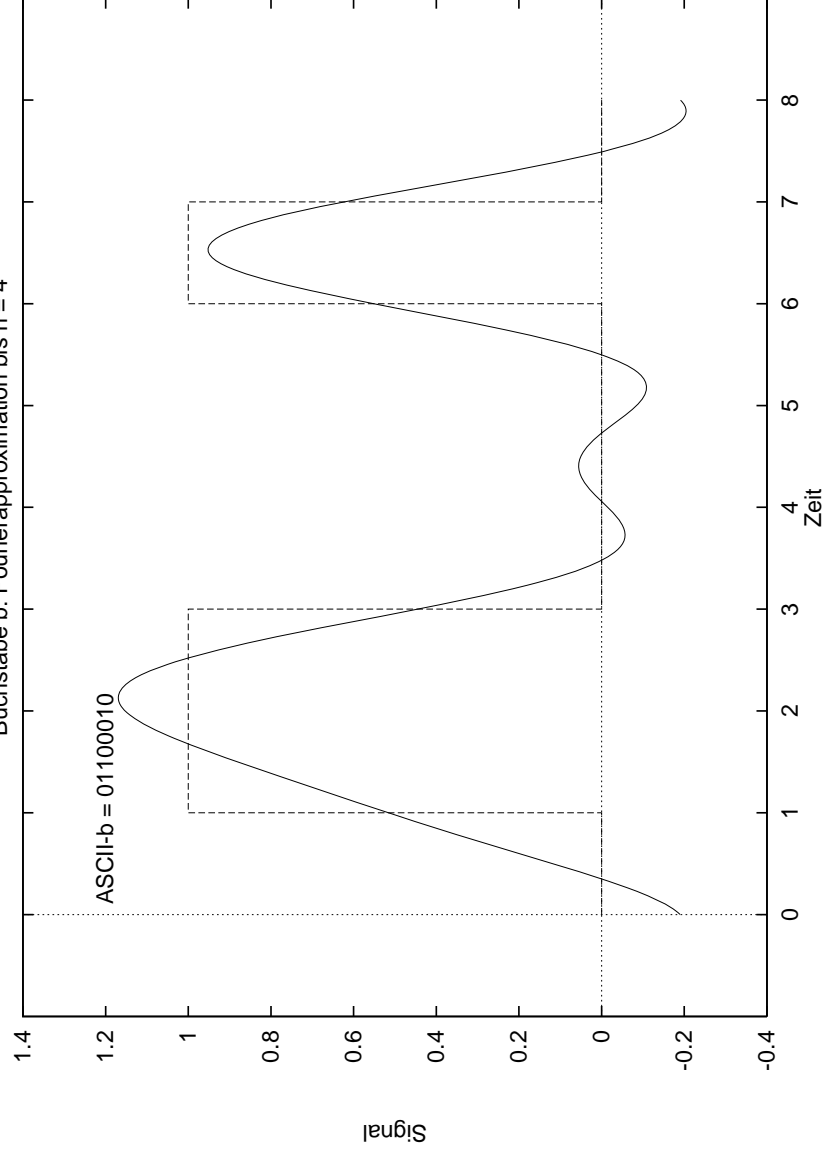
Buchstabe b: Fourierapproximation bis n = 2



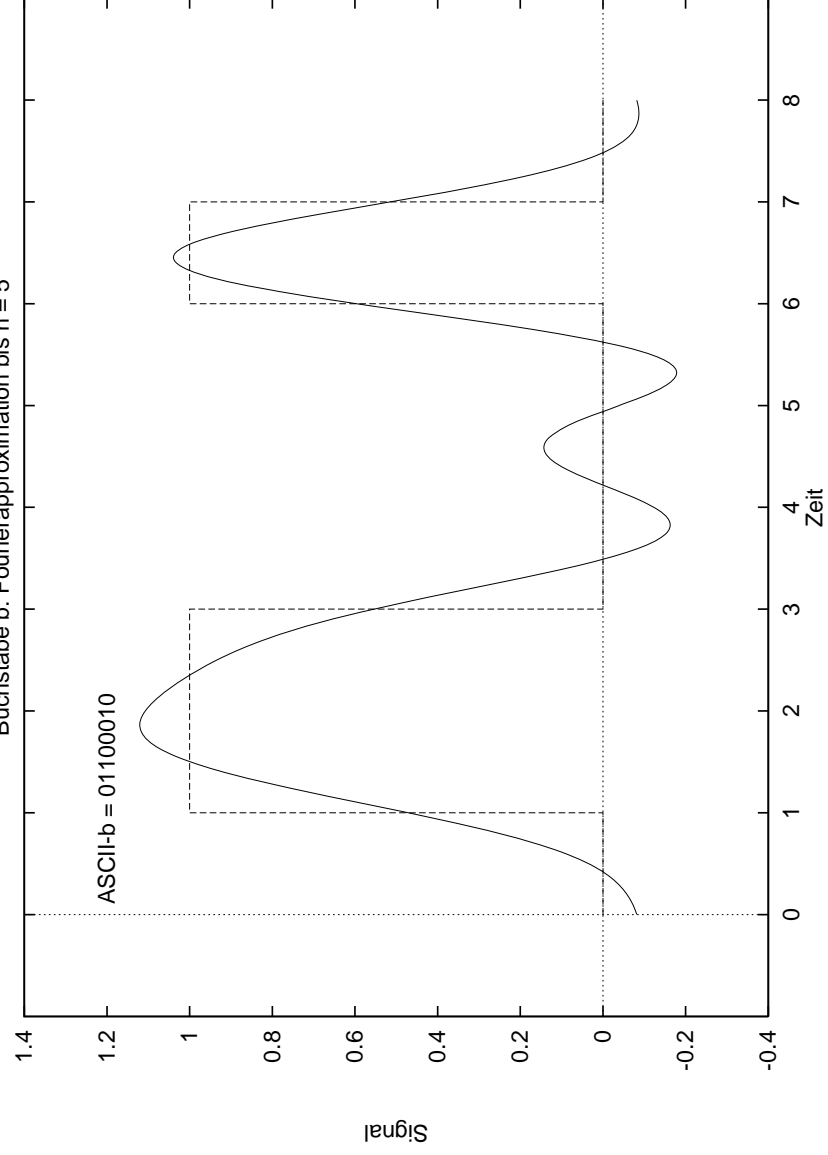
Buchstabe b: Fourierapproximation bis n = 3



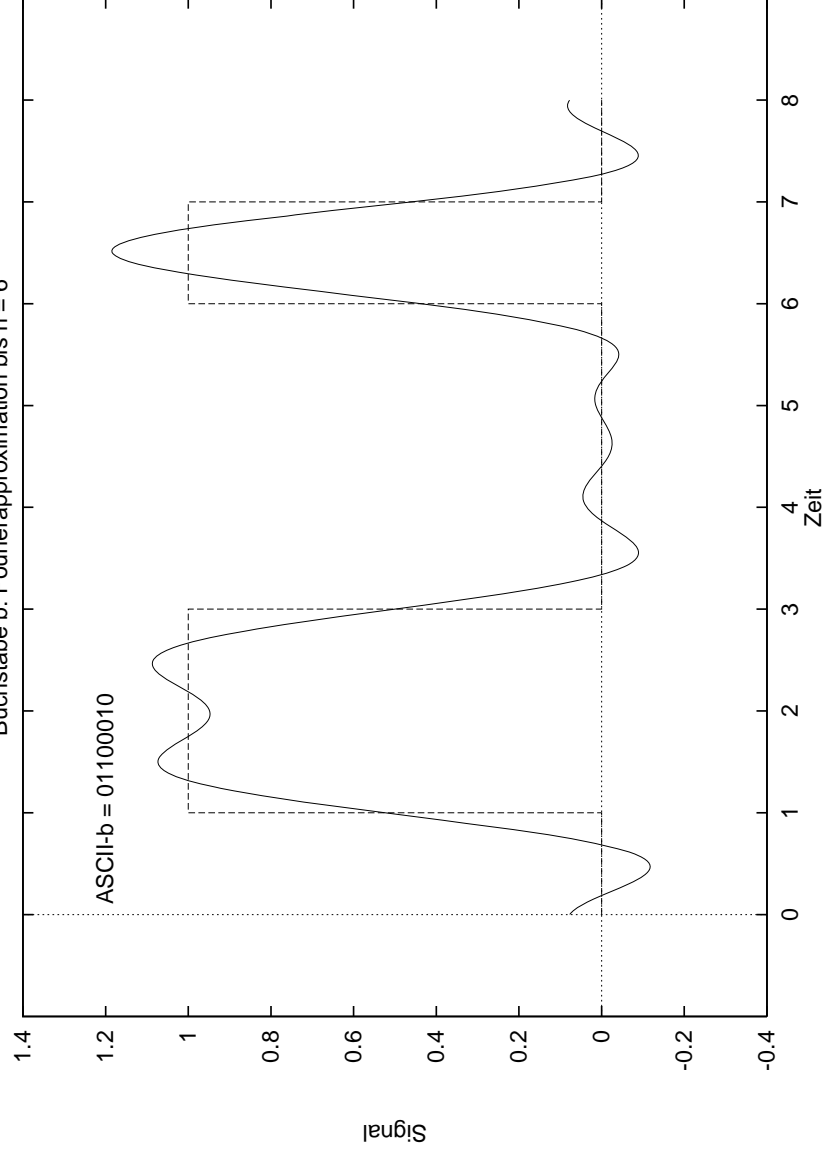
Buchstabe b: Fourierapproximation bis n = 4



Buchstabe b: Fourierapproximation bis n = 5



Buchstabe b: Fourierapproximation bis n = 6



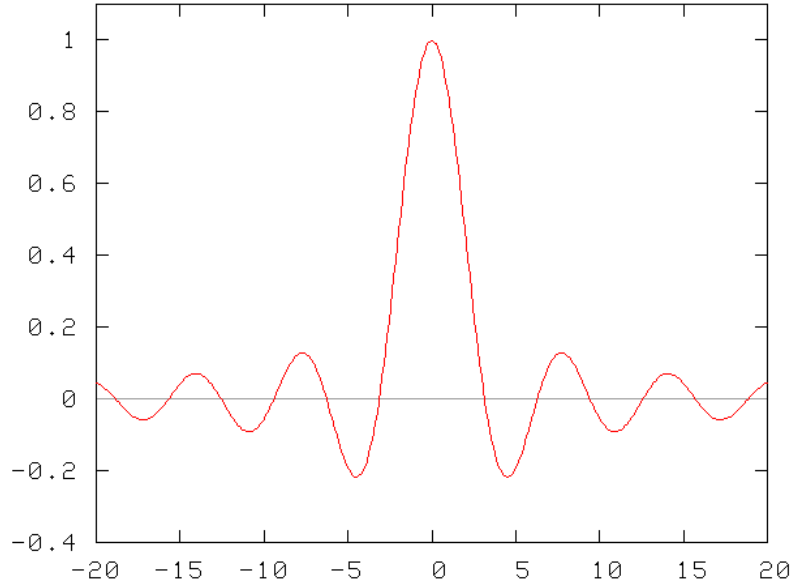
Nyquist-Theorem:

Sei f eine Funktion der Form

$$f(t) = \int_0^g (a(v) \cdot \cos(2 \cdot \pi \cdot v \cdot t) + b(v) \cdot \sin(2 \cdot \pi \cdot v \cdot t)) dv.$$

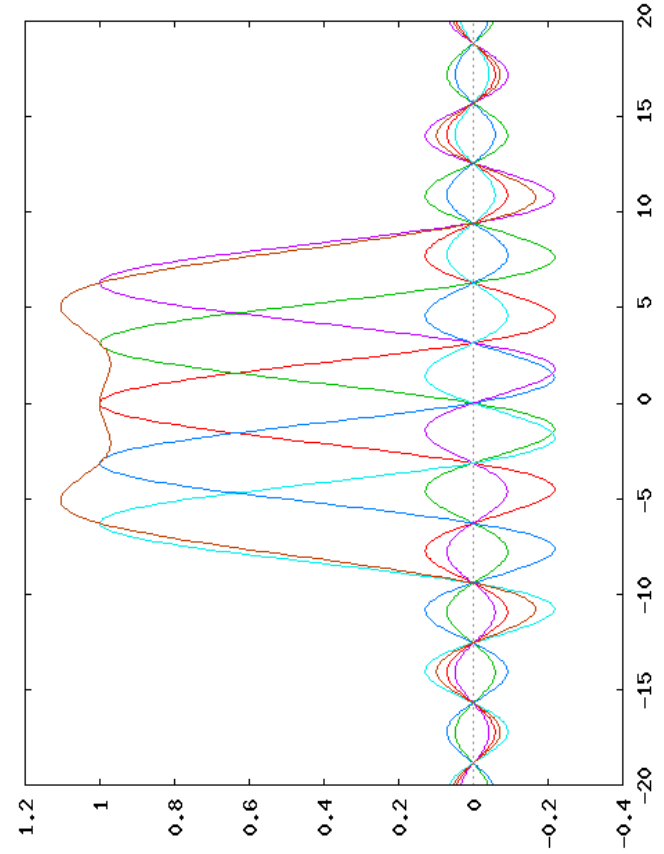
Sei $s \leq 1 / (2 \cdot g)$, dann ist $f(t)$ darstellbar in der Form

$$f(t) = \sum_n f(n \cdot s) \cdot \frac{\sin\left(\frac{\pi \cdot t}{s} - n \cdot \pi\right)}{\frac{\pi \cdot t}{s} - n \cdot \pi}$$



Graph der Funktion $\sin(x)/x$

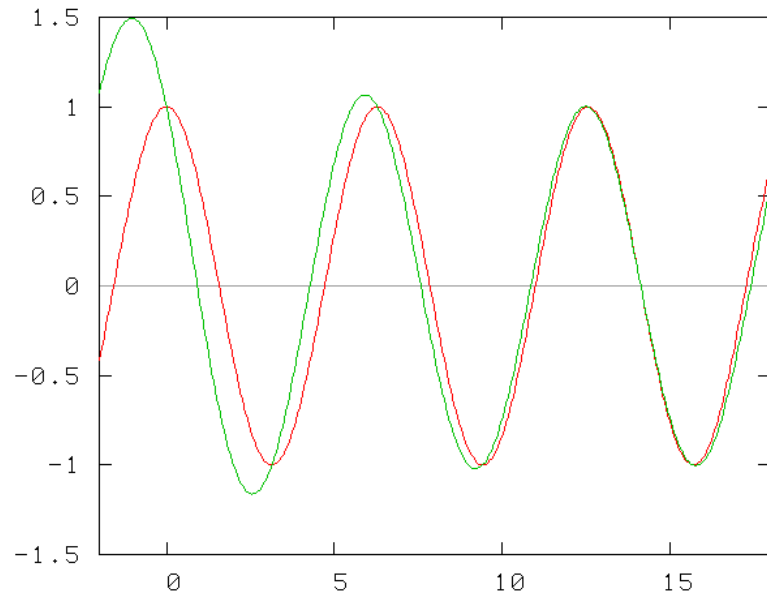
Zur Approximation mittels Nyquist-Theorem:



Fünfmal die Funktion $\sin(x)/x$, jeweils um π versetzt.

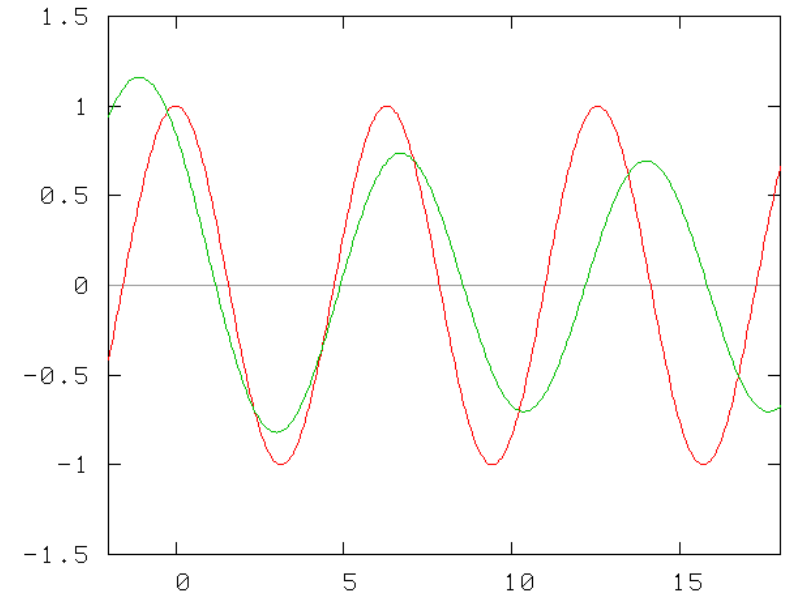
Man sieht sehr schön die Gibbschen Spitzen der Summenfunktion.

Demonstration der Nyquist-Formel: Approximation der Funktion Kosinus:



Verwendet wurden 10 Stützstellen, $\cos(0,0)$, $\cos(\pi)$, $\cos(2*\pi)$, $\cos(3*\pi)$, $\cos(4*\pi)$, $\cos(5*\pi)$, $\cos(6*\pi)$, $\cos(7*\pi)$, $\cos(8*\pi)$, $\cos(9*\pi)$; der Approximationsfehler rührt daher, daß die Summenbildung nur über zehn Terme erfolgte.

Demonstration des Aliaseffekts bei zu geringer Abtastrate; statt der Abtastperiode π wurde die Abtastperiode $1,1*\pi$ verwendet.

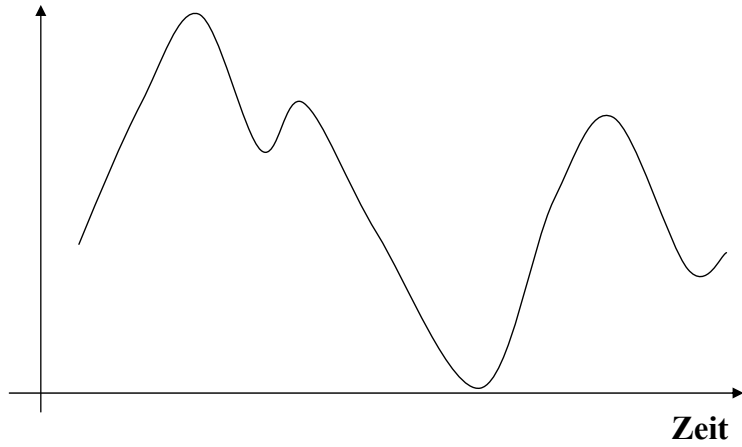


Zur Approximation von $\cos(x)$ wurden zehn Stützstellen verwendet, $\cos(0,0)$, $\cos(1,1*\pi)$, $\cos(2,2*\pi)$, $\cos(3,3*\pi)$, $\cos(4,4*\pi)$, $\cos(5,5*\pi)$, $\cos(6,6*\pi)$, $\cos(7,7*\pi)$, $\cos(8,8*\pi)$, $\cos(9,9*\pi)$; man sieht deutlich den Approximationsfehler.

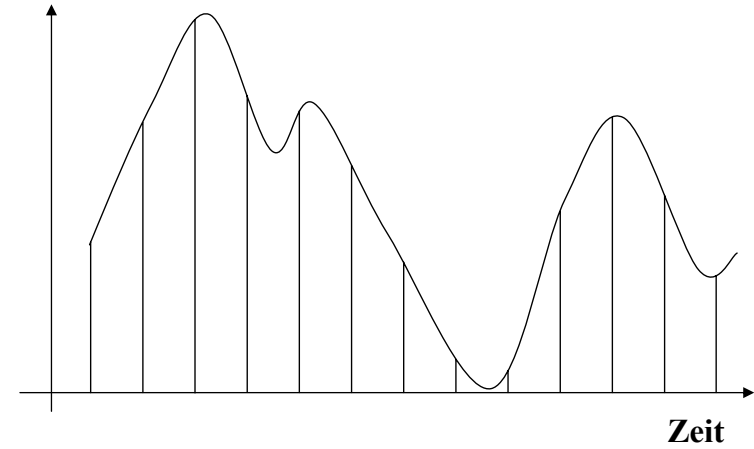
Bemerkung: Man erhält Aliaseffekte, falls die Abtastfrequenz kleiner als die doppelte Grenzfrequenz ist .

Zur Diskretisierung kontinuierlicher Signale:

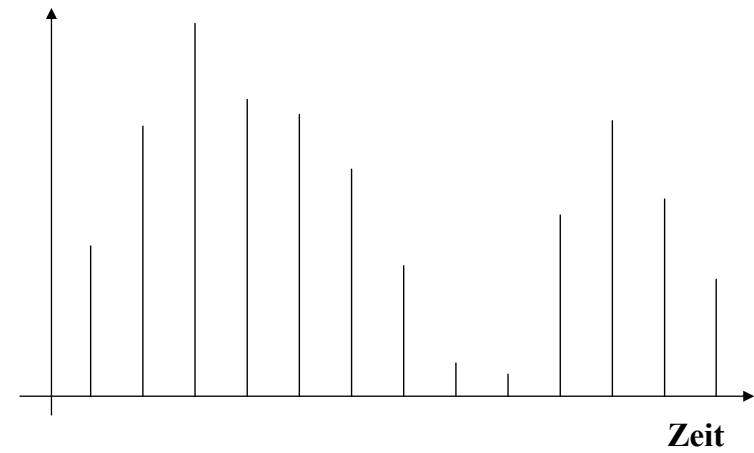
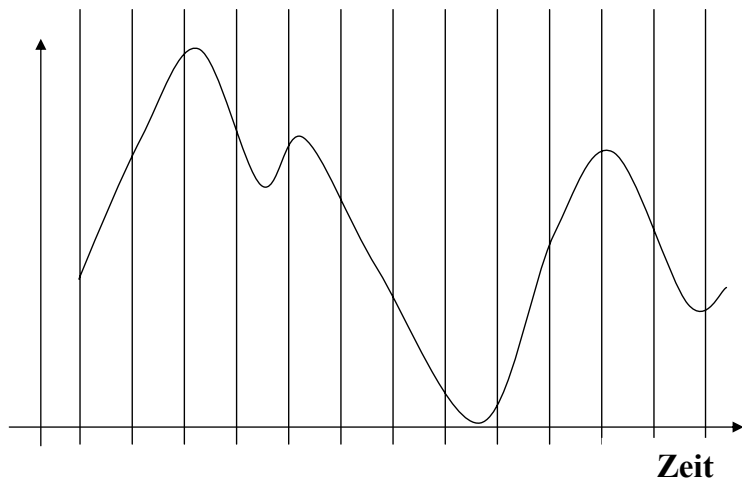
Beispiel eines wert- und zeitkontinuierlichen Signals:



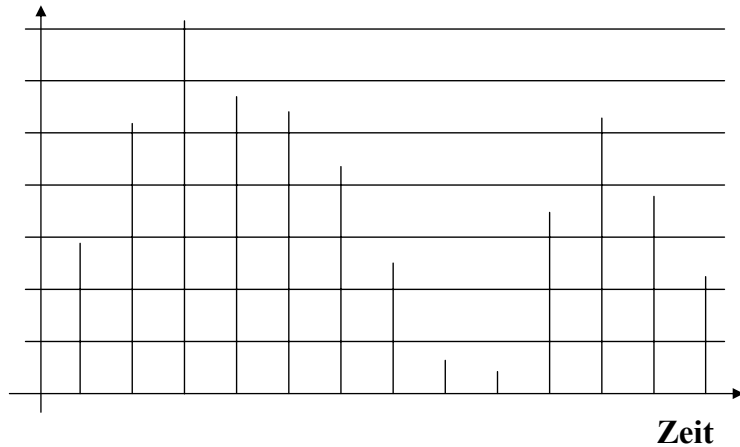
Man erhält ein wertkontinuierliches und zeitdiskretes Signal:



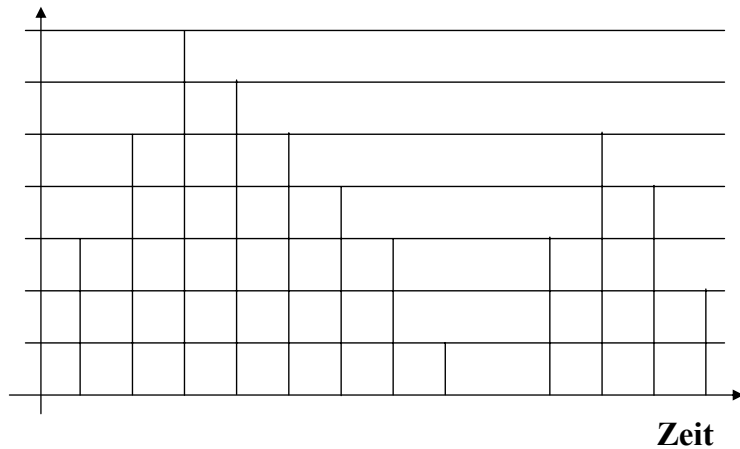
Überlagerung eines äquidistanten Zeitrasters:



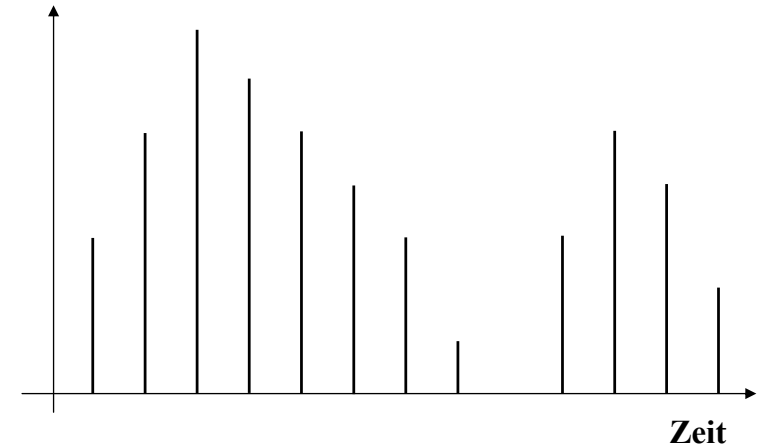
Quantelung der Amplitudenwerte ergibt ein wert- und zeitdiskretes Signal:



Äquidistante Quantelung auf acht Werte:



Das erhaltene Signal ist wert- und zeitdiskret.



Bemerkungen:

- (i) Bei der Zeitdiskretisierung sollte man auf die Einhaltung des Nyquist-Kriteriums achten, sonst erhält man Aliaseffekte.
- (ii) Die Quantisierung führt zu einem Quantisierungsfehler.
- (iii) Zu einer ersten Abschätzung des Quantisierungsrauschens QR:

Repräsentieren die Amplitudenwerte Energien oder Leistungen, dann gilt bei 2^n Linearstufen:

$$\begin{aligned} \text{QR} &\approx 10 * \log_{10} (1/2^n) = 10 * \log_{10} (2^{-n}) \\ &= -10 * n * \log_{10} 2 \approx -3 * n \text{ [dB]} \end{aligned}$$

Pulscode-Modulation im Zusammenhang mit menschlichem Hören und Telephonie:

Man schätzt, daß die kleinste Bestrahlungsstärke, die der Mensch noch wahrnimmt, $2 \cdot 10^{-12}$ Watt/m² beträgt. Die größte "noch ertragbare" Bestrahlungsstärke ist etwa um den Faktor 10^{12} größer. Ein Telephonkanal stellt einen maximalen Frequenzbereich von 4.000 Hz zur Verfügung. Das Abtasttheorem fordert eine Mindestabtastrate von 8.000 Abtastungen pro Sekunde, damit erfolgt alle 125 µs eine Amplitudenbestimmung. Es bleibt die Festlegung der Zahl der Quantisierungsstufen.

Fordert man für die Quantisierung einen minimalen Signal-Rausch-Abstand von 65 dB, dann gilt:

$$65 = 20 \cdot \log_{10} x$$
$$x = 10^{3,25} \approx 1778$$

(Bemerkung: Man verwendet den Faktor 20, weil man auf Telephonleitungen Spannungen oder Stromstärken mißt und nicht Energien oder Leistungen.)

Die Unterscheidung von 1778 Werten erfordert bei einer Bitcodierung 11 Bit. Im Telefonsystem verwendet man nur 8 Bit, indem man kleine Amplitudenwerte relativ genauer als große Amplitudenwerte darstellt. Es sind zwei Kompondierungsfunktionen weit verbreitet, in Europa die A-Kennlinie (13 Bit Diskriminierung) und in den USA und Japan die µ-Kennlinie (14 Bit Diskriminierung).

A-Kennlinie:

Seien $A = 87,56$ und $B = 1 + \ln(A)$.

$$y = \begin{cases} (1 + \ln(Ax)) / B & \text{für } 1/A \leq x \leq 1 \\ A \cdot x / B & \text{für } -1/A \leq x \leq 1/A \\ (1 + \ln(-Ax)) / B & \text{für } -1 \leq x \leq -1/A \end{cases}$$

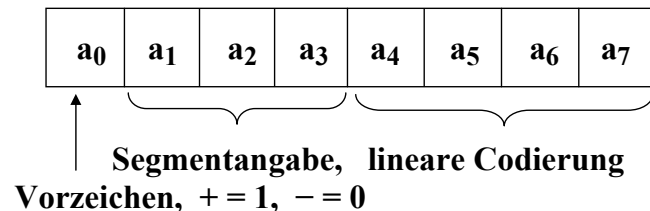
µ-Kennlinie:

$$y = \operatorname{sgn}(x) \cdot \ln(1 + \mu|x|) / \ln(1 + \mu) \quad \text{für } -1 \leq x \leq 1$$

mit $\mu = 255$.

Bemerkung: Den Argumentbereich teilt man bei beiden Kompondierungen in einzelne Segmente auf, innerhalb deren man jeweils linear interpoliert. Beide Codierungsvorschriften sind in der ITU-Empfehlung G.711 festgeschrieben.

Format eines PCM-Codeworts:



Intervalleinteilung für positiven Teil der A-Kennlinie:

Segment +7	$x = 1$	$y = 1$
Segment +6	$x = 1/2$	$y = 0,873336$
Segment +5	$x = 1/4$	$y = 0,746672$
Segment +4	$x = 1/8$	$y = 0,620008$
Segment +3	$x = 1/16$	$y = 0,493343$
Segment +2	$x = 1/32$	$y = 0,366679$
Segment +1b	$x = 1/64$	$y = 0,240015$
Segment +1a	$x = 1/128$	$y = 0,125004$
	$x = 0$	$y = 0$

Bemerkung: Innerhalb der einzelnen Segmente interpoliert man linear.

Bemerkung: 8.000 Werte pro Sekunde je zu 8 Bit ergibt einen Telephonkanal von 64 kb/s.

Die Umkehrfunktionen:

für das A-Gesetz: $F^{-1}(x) =$

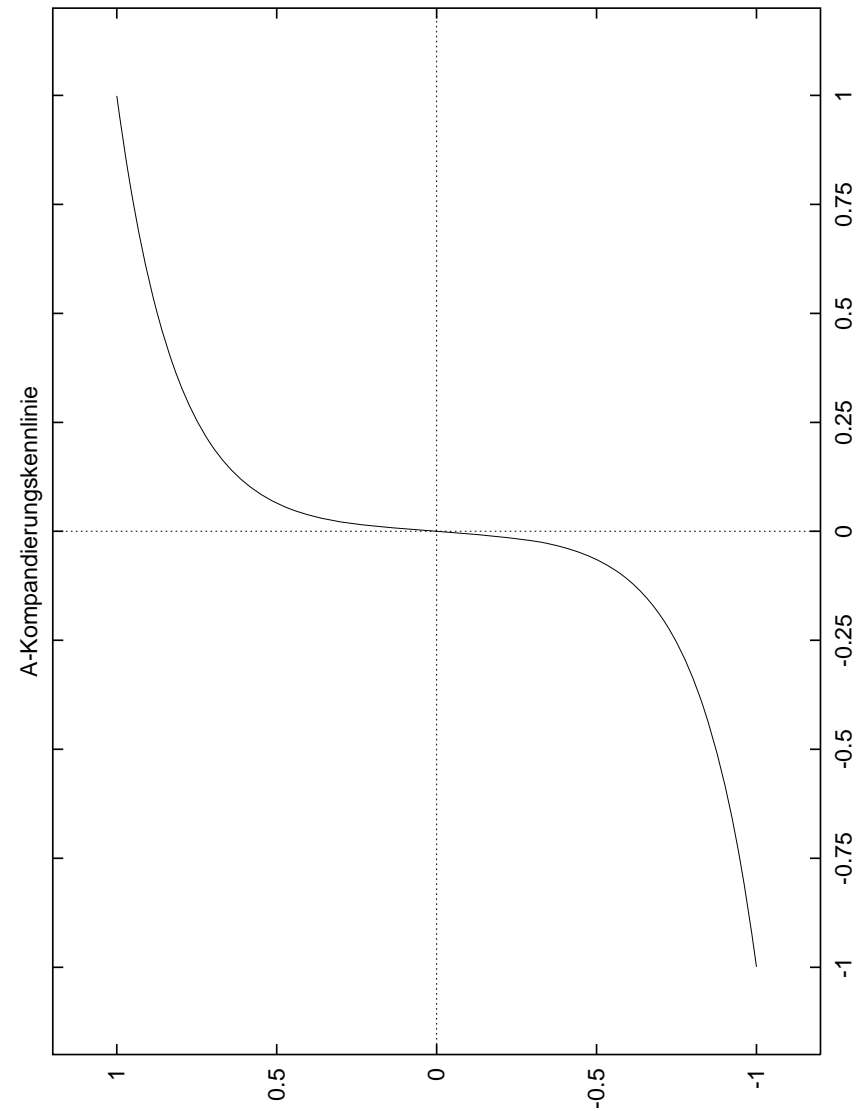
$$\text{sgn}(x) * |x| * (1 + \ln(A)) / A$$

für $0 \leq |x| \leq 1/(1+\ln(A))$

$$\text{sgn}(x) * \exp(|x| * (1 + \ln(A)) - 1) / A$$

für $1 / (1 + \ln(A)) \leq |x| \leq 1$

für das μ -Gesetz: $F^{-1}(x) = \text{sgn}(x) * ((1 + \mu)^{|x|} - 1) / \mu$



Wandlungstabelle für das μ -Gesetz (nur positive Werte):

Eingabewerte													Ausgabewerte						
Bitpositionen													Segment Linearwert						
12	11	10	9	8	7	6	5	4	3	2	1	0	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1	a	b	c	d	x	0	0	0	a	b	c	d
0	0	0	0	0	0	1	a	b	c	d	x	x	0	0	1	a	b	c	d
0	0	0	0	0	1	a	b	c	d	x	x	x	0	1	0	a	b	c	d
0	0	0	0	1	a	b	c	d	x	x	x	x	0	1	1	a	b	c	d
0	0	0	1	a	b	c	d	x	x	x	x	x	1	0	0	a	b	c	d
0	0	1	a	b	c	d	x	x	x	x	x	x	1	0	1	a	b	c	d
0	1	a	b	c	d	x	x	x	x	x	x	x	1	1	0	a	b	c	d
1	a	b	c	d	x	x	x	x	x	x	x	x	1	1	1	a	b	c	d

Bemerkung: Auf den Meßwert wird vor der Wandlung 33 addiert.

Wandlungstabelle für das A-Gesetz (nur positive Werte):

Eingabewerte													Ausgabewerte					
Bitpositionen													Segment Linearwert					
11	10	9	8	7	6	5	4	3	2	1	0	6	5	4	3	2	1	0
0	0	0	0	0	0	0	a	b	c	d	x	0	0	0	a	b	c	d
0	0	0	0	0	0	1	a	b	c	d	x	0	0	1	a	b	c	d
0	0	0	0	0	1	a	b	c	d	x	x	0	1	0	a	b	c	d
0	0	0	0	1	a	b	c	d	x	x	x	0	1	1	a	b	c	d
0	0	0	1	a	b	c	d	x	x	x	x	1	0	0	a	b	c	d
0	0	1	a	b	c	d	x	x	x	x	x	1	0	1	a	b	c	d
0	1	a	b	c	d	x	x	x	x	x	x	1	1	0	a	b	c	d
1	a	b	c	d	x	x	x	x	x	x	x	1	1	1	a	b	c	d

Bemerkung: Man sieht die lineare Wandlung im Nullsegment.

Eine C-Routine zur Wandlung gemäß μ -Gesetz:

```

unsigned char ulaw (short linear) {
    unsigned i, sign;
    unsigned char segment, segv;
    unsigned short out, absv, tp;

    /* Addition des Korrekturwerts
       und Bestimmung der Magnitude. */
    absv = abs (linear) + 33;
    tp = absv;
    sign = (linear >= 0) ? 1 : 0;
    for (i = 0; i < 16; ++i){
        out = tp & 0x8000;
        if (out)
            break;
        tp <<= 1;
    }
    segment = 11 - i;
    segv = (absv >> segment) & 0x0F;
    --segment;
    segment <<= 4;
    out = segment + segv;
    if (absv > 8191)
        out = 0x7F;
    /* Es wird der invertierte  $\mu$ -Wert gegeben. */
    if (sign)
        return out ^= 0xFF;
    else
        return out ^= 0x7F;
} /*ulaw*/

```

Beispiel zur Reduzierung von 11-Bit-Werten auf 8-Bit-Werte mittels μ -Gesetz, hier nur nichtnegative Werte:

0/1024	--	0
1/1024	--	5
2/1024	--	9
3/1024	--	13
4/1024	--	16
5/1024	--	19
6/1024	--	21
7/1024	--	23
8/1024	--	25
	:	
256/1024	--	96
257/1024	--	96
258/1024	--	96
259/1024	--	97
	:	
512/1024	--	112
513/1024	--	112
514/1024	--	112
515/1024	--	112
	:	
1000/1024	--	127
1001/1024	--	127
1002/1024	--	127
1003/1024	--	127

Bemerkung: Bei der Rücktransformation von 8-Bit-Werten zu 11-Bit-Werten tritt ein Informationsverlust auf, z. B. aus dem 8-Bit-Wert 96 wird der 11-Bit-Wert 252 unabhängig vom Originalwert.