

# Anwendungsmöglichkeiten der Fourierreihe

*Marcel Garus*

*Betreuung durch Herrn Krüger*

*Mathematik LK*

*Q1 16/17*

# Inhaltsverzeichnis

<b>1</b>	<b>Hinführung</b>	<b>1</b>
<b>2</b>	<b>Fourierreihe</b>	<b>1</b>
2.1	Definitionen . . . . .	1
2.2	Herleitung . . . . .	2
2.2.1	Orthogonalität von Kosinus und Sinus . . . . .	2
2.2.2	Berechnen von $a_0$ . . . . .	4
2.2.3	Berechnen von $a_k$ und $b_k$ . . . . .	5
2.3	Beispiele . . . . .	6
2.3.1	Beispiel 1: Sägezahnkurve . . . . .	6
<b>3</b>	<b>Anwendungsmöglichkeiten</b>	<b>8</b>
3.1	Spektralanalysen . . . . .	8
3.1.1	Strahlungsmessung . . . . .	8
3.1.2	Stimmgeräte . . . . .	8
3.2	Signale filtern und speichern . . . . .	8
3.2.1	EKGs . . . . .	8
3.2.2	Tonlagenveränderung . . . . .	8
3.2.3	Kompression von Bild- und Tonmaterial . . . . .	9
3.2.4	Streamen von Multimedialinhalten . . . . .	10
3.2.5	Bildverarbeitung . . . . .	10
3.3	Erzeugung periodischer Signale . . . . .	11
3.4	Lösung von Differenzialgleichungen . . . . .	11
<b>4</b>	<b>Konkretes Anwendungsbeispiel</b>	<b>11</b>
<b>5</b>	<b>Zusammenfassung</b>	<b>11</b>
<b>6</b>	<b>Anhang</b>	<b>11</b>
6.1	Python-Skript als Stimmgerät . . . . .	11
6.2	Python-Skript zur Berechnung der Fourierkoeffizienten . . . . .	13
<b>7</b>	<b>Literaturverzeichnis</b>	<b>14</b>
<b>8</b>	<b>Selbstständigkeitserklärung</b>	<b>15</b>

# 1 Einführung

In letzter Zeit spielt Datenspeicherung und -übertragung eine immer größere Rolle in unserem Leben. Dabei wird häufig die Fourierreihe zur Darstellung von Signalgrößen verwendet.

Da ich Mathe-, Informatik- und Physik-interessiert bin und im Rahmen der Deutschen Schülerakademie bereits Erfahrungen mit der Fourierreihe zur Bildverarbeitung gesammelt habe, werde ich das Thema hier vertieft behandeln.

Mithilfe der Fourier-Analyse ist es hierbei möglich, eine periodische Funktion in einzelne harmonische Schwingungen aufzuteilen, die wieder zu der Ausgangsfunktion zusammengesetzt werden können. Da die anschließende Rekonstruktion die Ausgangsfunktion nur annähert, kann diese alle Daten repräsentieren, bei denen es nicht auf exakte Werte ankommt. Sie eignet sich deshalb sehr gut dazu, vom Menschen zu erfassende Daten wie Musik, Bilder und Videos zu komprimieren.

Grundgedanke der Fourierreihe ist, nach der Aufteilung der Funktion in mehrere Elementarschwingungen unterschiedlicher Frequenz durch eine Analyse deren Amplituden die Wichtigkeit dieser zu bewerten und unwichtige Schwingungen herauszufiltern.

Die Fourierreihe dient somit auch als Grundlage für die Filterung von Signalen, um unerwünschte Signalanteile wie zum Beispiel Messrauschen zu unterdrücken und erleichtert den Vergleich mehrerer Datensätze auf grundsätzliche Ähnlichkeit.

In dieser Facharbeit wird zunächst die formale Definition der Fourierreihe nach einschlägiger Literatur genannt und anschließend die Berechnung der Koeffizienten hergeleitet. Danach werden Eigenschaften der Fourierreihe beschrieben und Beispiele genannt. Nachdem die mathematische Grundlage gelegt wurde, werden verschiedene Anwendungsmöglichkeiten von Fourierreihen beschrieben. Am Ende der Arbeit wird ein Programm stehen, welches eine Tonschwingung in ihre einzelnen Bestandteile zerlegt.

## 2 Fourierreihe

### 2.1 Definitionen

Sei  $f : \mathbb{R} \rightarrow \mathbb{R}$  eine  $2\pi$ -periodische Funktion und auf dem Intervall  $[-\pi, \pi]$  integrierbar<sup>1</sup>. Die Zahlen<sup>2</sup>

$$\begin{aligned} a_0 &:= \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) dx, \\ a_k &:= \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(kx) dx, \\ b_k &:= \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin(kx) dx \end{aligned}$$

heißen **Fourierkoeffizienten** der Funktion  $f$ . Die darüber erklärte Reihe<sup>3</sup>

$$a_0 + \sum_{k=1}^{\infty} (a_k \cos(kx) + b_k \sin(kx))$$

<sup>1</sup>Eine Funktion  $f : [a, b] \rightarrow \mathbb{R}$  ist genau dann (Riemann)-integrierbar, wenn zu jedem  $\epsilon > 0$  Treppenfunktionen  $\alpha, \beta \in T[a, b]$  existieren mit  $\alpha < f < \beta$  und  $\int_a^b \alpha(x) dx - \int_a^b \beta(x) dx \leq \epsilon$  (*Analysis 1*, S. 125)

<sup>2</sup>vgl. *Mathematik verstehen und anwenden*, S. 677; In einigen Abhandlungen wird  $a_0$  mit dem Faktor  $\frac{1}{\pi}$  statt  $\frac{1}{2\pi}$  definiert, in der Fourierreihe dann aber als  $\frac{a_0}{2}$  verwendet. Das hat inhaltlich jedoch keine Konsequenzen.

<sup>3</sup>*Theorie Analytique de la chaleur*, Joseph Fourier, S. 255

heißt **Fourierreihe** von  $f$ . Die Partialsummen bezeichnen wir mit

$$S_n(f, x) := a_0 + \sum_{k=1}^n (a_k \cos(kx) + b_k \sin(kx)), n \in \mathbb{N}_0.$$

## 2.2 Herleitung

In der Praxis sind die Ausgangsfunktionen meist integrierbar, aber nicht  $2\pi$ -periodisch; der zu betrachtende Bereich im Intervall  $[0, T]$  wird deshalb in  $x$ -Richtung auf die Breite von  $2\pi$  gestaucht oder gestreckt und periodisch fortgesetzt:

$$f(x + 2\pi k) := f(x * \frac{2\pi}{T}), k \in \mathbb{Z}$$

Wir können nun also davon ausgehen, dass die Ausgangsfunktion  $2\pi$ -periodisch und integrierbar ist. Um sie als *trigonometrisches Polynom*<sup>4</sup>, also als Überlagerung von Sinus- und Kosinusfunktionen in der Form

$$S_n(f, x) = a_0 + \sum_{k=1}^n (a_k \cos(kx) + b_k \sin(kx)), \text{ mit } a_0, a_k, b_k \in \mathbb{R},$$

darzustellen, müssen die Koeffizienten  $a_0$ ,  $a_k$  und  $b_k$  bestimmt werden. Für  $k \in \mathbb{N}$  sind die Summanden  $\cos(kx)$  und  $\sin(kx)$  auch  $2\pi$ -periodisch und somit auch die gesamte Funktion  $S$ .

Die „Fehlerdifferenz“ zwischen  $S_n(f, x)$  und der Ausgangsfunktion  $f$  kann als Betrag der Differenz in jedem Punkt  $x$  bestimmt werden. Da der Betrag jedoch über eine Fallunterscheidung definiert ist, ist es einfacher, den auch in der Informatik weit verbreiteten *quadratischen Fehler* zu benutzen; die Differenz wird also quadriert.

Man kann eine Fehlerfunktion  $g(a_0, \dots, a_n, b_1, \dots, b_n)$  definieren, die den Fehler von  $S_n(f, x)$  zur Ausgangsfunktion berechnet.

Diese wird so umgeformt, dass  $a_0$  alleine steht, damit sie später einfacher nach  $a_0$  abzuleiten ist. Dazu wurde ausgenutzt, dass  $(a - b)^2 = (b - a)^2$  gilt.

$$\begin{aligned} g(a_0, \dots, a_n, b_1, \dots, b_n) &:= \int_{-\pi}^{\pi} \left[ f(x) - \left( a_0 + \sum_{k=1}^n (a_k \cos(kx) + b_k \sin(kx)) \right) \right]^2 dx \\ &= \int_{-\pi}^{\pi} \left[ a_0 + \left( -f(x) + \sum_{k=1}^n (a_k \cos(kx) + b_k \sin(kx)) \right) \right]^2 dx \end{aligned}$$

Da der quadratische Fehler der Fourierreihe zur Ausgangsfunktion möglichst klein sein soll, wird das Minimum der Fehlerfunktion gesucht.

### 2.2.1 Orthogonalität von Kosinus und Sinus

Für spätere Rechnungen ist es hilfreich, die Orthogonalität von Kosinus und Sinus zu beweisen.

Sinus- und Kosinus-Terme der gleichen Frequenz lassen sich mittels der komplexen Exponentialfunktion zusammenfassen. Mit der Eulerschen Formel  $e^{ix} = \cos x + i \sin x$ ,  $x \in \mathbb{R}$ , können wir die Fourierreihe also auch folgendermaßen darstellen<sup>5</sup>:

$$\sum_{k=-\infty}^{\infty} c_k e^{ikx} := \lim_{n \rightarrow \infty} \sum_{k=-n}^n c_k e^{ikx}$$

<sup>4</sup>Höhere Mathematik 2, S.288

<sup>5</sup>vgl. Analysis 1, S. 86

mit

$$\begin{aligned} c_0 &= a_0, \\ c_{-k} &= \frac{1}{2}(a_k + ib_k), k \in \mathbb{N}, \\ c_k &= \frac{1}{2}(a_k - ib_k), k \in \mathbb{N}, \\ \Rightarrow a_k &= c_{-k} + c_k, k \in \mathbb{N}, \\ \Rightarrow b_k &= i[c_k - c_{-k}], k \in \mathbb{N} \end{aligned}$$

Fasst man die Summanden zu  $k$  und  $-k$ ,  $k \in \mathbb{N}$  zusammen, erhält man:

$$c_{-k}e^{-ikx} + c_k e^{ikx}$$

Einsetzen der soeben definierten  $c_{-k}$  und  $c_k$  und Ausschreiben der komplexen Exponentialfunktion als Sinus- und Kosinusterm liefert:

$$= \frac{1}{2}(a_k + ib_k)(\cos(kx) - i \sin(kx)) + \frac{1}{2}(a_k - ib_k)(\cos(kx) + i \sin(kx))$$

Ausmultiplizieren der Produkte und Sortieren in Terme mit  $i$  und Terme ohne  $i$  ergibt einen Term, bei dem sich die komplexen Summanden paarweise zu reellen ergänzen:

$$\begin{aligned} &= \frac{1}{2}(a_k \cos(kx) + b_k \sin(kx) + a_k \cos(kx) + b_k \sin(kx)) \\ &\quad + \frac{i}{2}(-a_k \sin(kx) + b_k \cos(kx) + a_k \sin(kx) - b_k \cos(kx)) \\ &= \frac{1}{2}(2a_k \cos(kx) + 2b_k \sin(kx)) + 0 \\ &= a_k \cos(kx) + b_k \sin(kx) \end{aligned}$$

Die **Orthogonalität**<sup>6</sup> der Funktionen  $e^{ikx}$ , also

$$\int_{-\pi}^{\pi} e^{ikx} e^{ilx} dx = \begin{cases} 2\pi & \text{für } k+l=0 \\ 0 & \text{für } k+l \neq 0 \end{cases}$$

lässt sich aufs Reelle in die folgenden Terme übertragen, die die **Orthogonalität von Kosinus und Sinus**<sup>7</sup> beschreiben:

$$\begin{aligned} \int_{-\pi}^{\pi} \cos(kx) \cdot \cos(lx) dx &= \begin{cases} \pi & \text{für } k = \pm l \\ 0 & \text{für } k \neq \pm l \end{cases} & \int_{-\pi}^{\pi} \sin(kx) \cdot \sin(lx) dx &= \begin{cases} \pi & \text{für } k = \pm l \\ 0 & \text{für } k \neq \pm l \end{cases} \\ \int_{-\pi}^{\pi} \sin(kx) \cdot \cos(lx) dx &= 0 \end{aligned}$$

<sup>6</sup>Die komplexe  $e$ -Funktion kann als Vektor angesehen werden, wobei der Betrag der Länge des Vektors und das übergebene Argument dem Winkel zur reellen Achse entspricht. Orthogonal („Senkrecht aufeinander stehend“) heißt, dass zwei Elemente eines Vektorraums das Skalarprodukt 0 haben, außer sie sind gleich gerichtet. Für  $k+l=0$ :  $\int_{-\pi}^{\pi} e^{ikx} e^{ilx} dx = \int_{-\pi}^{\pi} e^{i(k+l)x} dx = \int_{-\pi}^{\pi} e^0 dx = \int_{-\pi}^{\pi} 1 dx = 2\pi$

Für  $k+l \neq 0$ :  $\int_{-\pi}^{\pi} e^{ikx} e^{ilx} dx = \int_{-\pi}^{\pi} e^{i(k+l)x} dx = \left[ \frac{1}{i(k+l)} e^{i(k+l)x} \right]_{-\pi}^{\pi} = \frac{1}{i(k+l)} [e^{i(k+l)\pi} - e^{-i(k+l)\pi}] = \frac{1}{i(k+l)} [(\cos((k+l)\pi) + i \sin((k+l)\pi)) - (\cos((l+k)\pi) - i \sin((l+k)\pi))] = \frac{2 \sin((k+l)\pi)}{k+l} = 0$

<sup>7</sup>vgl. *Mathematik verstehen und Anwenden*, S. 687

### 2.2.2 Berechnen von $a_0$

Zunächst versuchen wir,  $a_0$  herauszufinden.<sup>8</sup> Beim Minimum der Funktion  $g$  abhängig von  $a_0$  hat nach der notwendigen Bedingung für Extremstellen die erste Ableitung<sup>9</sup> eine Nullstelle. Wir berechnen also zunächst die erste Ableitung von  $g$  nach  $a_0$ :

$$\frac{\delta g(a_0, a_1, \dots, a_n, b_1, \dots, b_n)}{\delta a_0}$$

Wir setzen den Ausdruck für  $g$  ein, dabei wenden wir die erste binomische Formel an, um die Summe zu quadrieren:

$$\begin{aligned} &= \frac{\delta}{\delta a_0} \int_{-\pi}^{\pi} \left[ a_0^2 + 2a_0 \left( -f(x) + \sum_{k=1}^n (a_k \cos(kx) + b_k \sin(kx)) \right) \right. \\ &\quad \left. + \left( -f(x) + \sum_{k=1}^n (a_k \cos(kx) + b_k \sin(kx)) \right)^2 \right] dx \end{aligned}$$

Da sich Ableitungen und Integrale von Summen auseinanderziehen und konstante Koeffizienten vorwegschreiben lassen, erhalten wir:

$$\begin{aligned} &= \frac{\delta}{\delta a_0} a_0^2 \int_{-\pi}^{\pi} 1 dx + \frac{\delta}{\delta a_0} 2a_0 \left[ \int_{-\pi}^{\pi} \left( -f(x) + \sum_{k=1}^n (a_k \cos(kx) + b_k \sin(kx)) \right) dx \right] \\ &\quad + \frac{\delta}{\delta a_0} \int_{-\pi}^{\pi} \left( -f(x) + \sum_{k=1}^n (a_k \cos(kx) + b_k \sin(kx)) \right)^2 dx \end{aligned}$$

Im ersten Summanden wird das Integral von 1 aufgelöst und  $a_0^2$  zu  $2a_0$  abgeleitet. Im zweiten Summanden wird  $2a_0$  zu 2 abgeleitet und das Integral wird in zwei Integrale aufgeteilt. Da im dritten Summanden kein  $a_0$  vorkommt und er deshalb konstant ist, fällt er beim Ableiten weg. Es bleibt:

$$= 2a_0 * 2\pi + 2 \left[ \int_{-\pi}^{\pi} -f(x) dx + \int_{-\pi}^{\pi} \sum_{k=1}^n (a_k \cos(kx) + b_k \sin(kx)) dx \right]$$

Weil Sinus und Kosinus  $2\pi$ -periodisch und die Flächen über und unter dem Graphen gleich groß sind, also  $\int_{-\pi}^{\pi} \sin(kx) = 0 = \int_{-\pi}^{\pi} \cos(kx)$  gilt, ist das zweite Integral von  $-\pi$  bis  $\pi$  von der Summe dieser Funktionen 0 und fällt weg. Übrig bleibt:

$$= 2a_0 2\pi - 2 \int_{-\pi}^{\pi} f(x) dx$$

Setzen wir wegen der notwendigen Bedingung diese Ableitung gleich 0, können wir nach  $a_0$  auflösen:

$$4a_0\pi - 2 \int_{-\pi}^{\pi} f(x) dx = 0 \quad \Leftrightarrow \quad 2a_0\pi = \int_{-\pi}^{\pi} f(x) dx \quad \Leftrightarrow \quad a_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) dx \quad (1)$$

<sup>8</sup>Rechnungen angelehnt an *Mathematik verstehen und anwenden*, S. 530

<sup>9</sup>Gemeint ist hier die *partielle Ableitung*: „Wir halten alle Variablen bis auf eine fest und rechnen mit der verbliebenen so, als wäre es die Einzige.“ *Mathematik verstehen und anwenden*, S. 512

### 2.2.3 Berechnen von $a_k$ und $b_k$

Um ein beliebiges  $a_k$  zu berechnen, müssen wir nun nach der notwendigen Bedingung für Extrempunkte die Ableitung mit 0 gleichsetzen. Dazu leiten wir zunächst unsere vorher definierte Fehlerfunktion nach  $a_k$  ab:

$$0 = \frac{\delta g(a_0, a_1, \dots, a_n, b_1, \dots, b_n)}{\delta a_k} = \frac{\delta}{\delta a_l} \int_{-\pi}^{\pi} \left[ a_0 + \left( -f(x) + \sum_{l=1}^n (a_l \cos(lx) + b_l \sin(lx)) \right) \right]^2 dx$$

Wie beim Ableiten nach  $a_0$  wenden wir hier die binomische Formel an, allerdings betrachten wir nun  $a_0 - f(x) + \sum_{l=1}^n b_l \sin(lx)$  und  $\sum_{l=1}^n a_l \cos(lx)$  als getrennte Summanden, wir ziehen die Summe also auseinander. Beim Multiplizieren von zwei Summen müssen wir darauf achten, dass die Zählvariablen unverfänglich benannt werden. Außerdem wird das Integral aufgeteilt.

$$\begin{aligned} &= \frac{\delta}{\delta a_k} \left[ \int_{-\pi}^{\pi} \left( a_0 - f(x) - \sum_{l=1}^n b_l \sin(lx) \right)^2 dx \right. \\ &\quad - 2 \int_{-\pi}^{\pi} \left( a_0 - f(x) - \sum_{l=1}^n b_l \sin(lx) \right) \left( \sum_{m=0}^n a_m \cos(mx) \right) dx \\ &\quad \left. + \int_{-\pi}^{\pi} \left( \sum_{l=1}^n a_l \cos(lx) \right) \left( \sum_{m=1}^n a_m \cos(mx) \right) dx \right] \end{aligned}$$

Die Ableitung des ersten Summanden ist 0, da er nicht von  $a_k$  abhängig ist. Die zweite Zeile ist für  $m \neq k$  auch nicht von  $a_k$  abhängig und muss deshalb nur für den Fall  $m = k$  betrachtet werden. Bei der dritten Zeile können die Summen nach vorne und das Integral in die Summe gezogen werden.

$$= -2 \int_{-\pi}^{\pi} \cos(mx) \left( a_0 - f(x) - \sum_{l=1}^n b_l \sin(lx) \right) dx + \frac{\delta}{\delta a_k} \sum_{l=1}^n \sum_{m=1}^n a_l a_m \int_{-\pi}^{\pi} \cos(lx) \cos(mx) dx$$

Der ersten Summand wurde ausgeklammert, beim zweiten wird die oben angesprochene Orthogonalität von Sinus und Kosinus ausgenutzt, um das Integral zu vereinfachen. Außerdem müssen die Summen im letzten Summanden nur für  $k = l = m$  betrachtet werden.

$$= -2 \int_{-\pi}^{\pi} f(x) \cos(kx) dx + 2a_0 \int_{-\pi}^{\pi} \cos(kx) dx + 2 \sum_{l=1}^n b_l \int_{-\pi}^{\pi} \cos(kx) \sin(kx) dx + \frac{\delta}{\delta a_k} a_k^2 \pi$$

Da das zweite Integral Kosinus über ein Intervall von  $2\pi$  integriert, ist dieser Summand 0. Beim dritten Summanden kommt uns wieder die oben angesprochene Orthogonalität von Sinus und Kosinus zugute; auch dieser Summand fällt weg.

Da nach  $a_l$  abgeleitet wird, bleibt von der letzten Summe nur das Element mit  $k = l$  bestehen. Dieses wird normal abgeleitet.

$$= -2 \int_{-\pi}^{\pi} f(x) \cos(kx) dx + 2a_k \pi$$

Die notwendige Bedingung ist nun vollständig gekürzt und lässt sich nach  $a_k$  umformen:

$$a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(kx) dx$$

Für die Berechnung von  $b_k$  ist die Berechnung analog.

Bislang wurde nur die notwendige Bedingung für die Koeffizienten durchgeführt. Würde man einen beliebigen Koeffizient jedoch gegen unendlich laufen lassen, würde die Fourierreihe besonders von dieser Schwingung repräsentiert werden; je größer der Koeffizient würde, desto größer würde auch der quadratische Fehler sein. Da die quadratische Fehlerfunktion somit kein Maximum hat, wissen wir, dass es sich bei den berechneten Koeffizienten um die mit dem *geringsten quadratischen Fehler zur Ausgangsfunktion* handeln muss.

Wir haben nun also alle Koeffizienten der Definition auf Seite 1 hergeleitet.

Da keiner der Koeffizienten von der Anzahl der Fourierreihenglieder  $n$  abhängt, müssen bei einer Erhöhung von  $n$  die bereits berechneten Fourierkoeffizienten nicht neu berechnet werden, die Koeffizienten der Partialsumme sind also unabhängig davon, wie viele Summenglieder man berechnet.

Insbesondere lässt sich nun der Limes anwenden:

$$\lim_{n \rightarrow \infty} S_n(f, x) \rightarrow f(x)$$

## 2.3 Beispiele

Funktionen werden **gerade** genannt, wenn sie achsensymmetrisch zur y-Achse sind. Funktionen werden **ungerade** genannt, wenn sie punktsymmetrisch zum Ursprung sind. Multipliziert man eine gerade und eine ungerade Funktion, erhält man eine ungerade Funktion.

Ist  $f$  eine ungerade Funktion, gleichen sich bei der Integration von  $-\pi$  bis  $\pi$  über das Produkt mit dem geraden Kosinus die Flächen links und rechts des Ursprungs aus. Die Kosinusterme fallen weg, also  $a_k = 0$ , und die Fourierreihe wird zur **Sinusreihe**.

Ist  $f$  gerade Funktion, gleichen sich bei der Integration von  $-\pi$  bis  $\pi$  über das Produkt mit dem ungeraden Sinus die Flächen links und rechts des Ursprungs aus. Die Sinusterme fallen weg, also  $b_k = 0$ , und die Fourierreihe wird zur **Kosinusreihe**.

### 2.3.1 Beispiel 1: Sägezahnkurve

Als Beispiel betrachten wir die  $2\pi$ -periodische Funktion

$$f(x) = \begin{cases} x, & -\pi < x < \pi \\ 0, & x = \pi. \end{cases}$$

Diese *Sägezahnkurve* kommt beispielsweise bei Kippschaltungen in der Elektrotechnik vor.

Da die Funktion punktsymmetrisch zum Ursprung und somit ungerade ist, gilt  $a_k = 0$  und die Fourierreihe wird zur Sinusreihe. Die Koeffizienten  $b_k$  ermitteln wir mit partieller Integration:

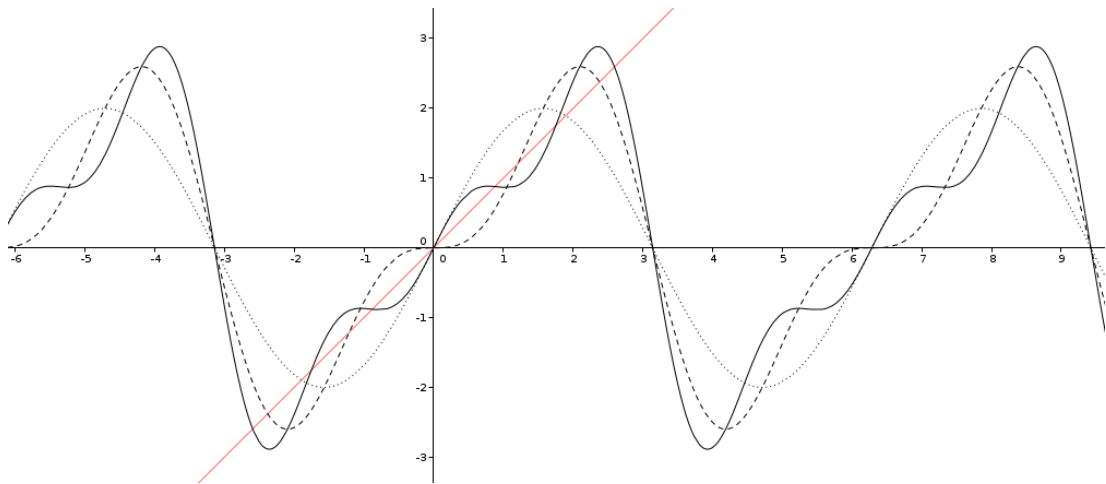
$$\begin{aligned} b_k &= \frac{1}{\pi} \int_{-\pi}^{\pi} x \sin(kx) dx = \frac{1}{\pi} \left[ x \left( -\frac{\cos(kx)}{k} \right) \right]_{-\pi}^{\pi} - \frac{1}{\pi} \int_{-\pi}^{\pi} 1 \left( -\frac{\cos(kx)}{k} \right) dx \\ &= -\frac{\pi}{\pi k} \cos(k\pi) - \frac{-\pi}{\pi} \left( -\frac{\cos(k(-\pi))}{k} \right) + \frac{1}{k\pi} \int_{-\pi}^{\pi} \cos(kx) dx \end{aligned}$$

Im folgenden Bild sind die ersten drei Partialsummen grafisch abgebildet. Der gepunktete Graph entspricht dabei  $S_1(f, x) = 2\sin(x)$ , der gestrichelte  $S_2(f, x) = 2\sin(x) - \sin(2x)$  und der durchgezogene  $S_3(f, x) = 2\sin(x) - \sin(2x) + \frac{2}{3}\sin(3x)$ .

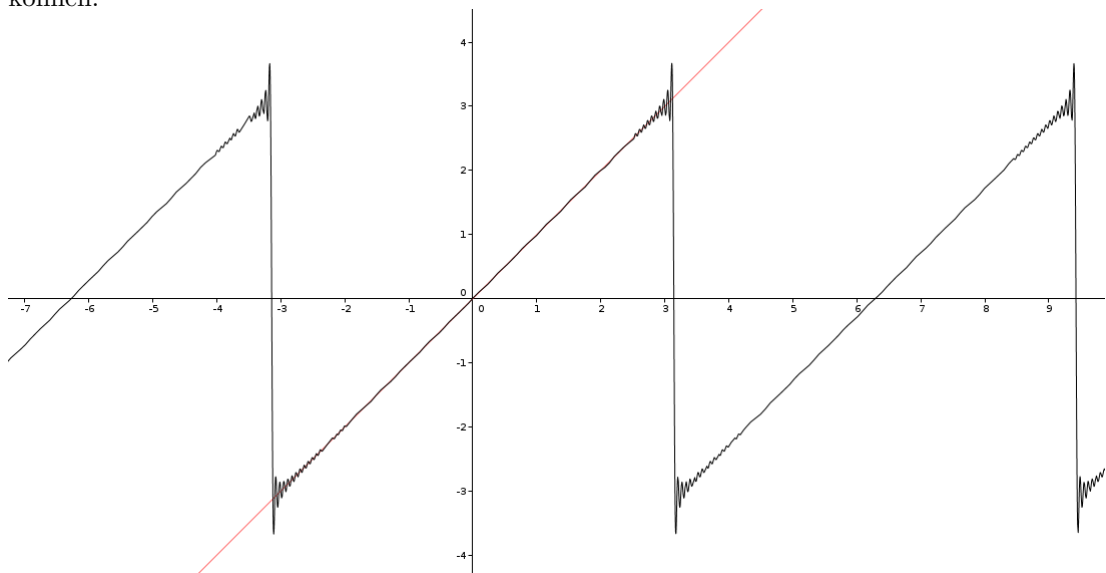
Dabei ist es oft erstaunlich, wie genau sich der Graph der Fourierfunktion schon mit wenigen Koeffizienten der rot gezeichneten Ausgangsfunktion im Intervall  $[-\pi, \pi]$  annähert.

Auch das  $2\pi$ -periodische Verhalten der Fourierfunktion ist hier gut erkenntlich.

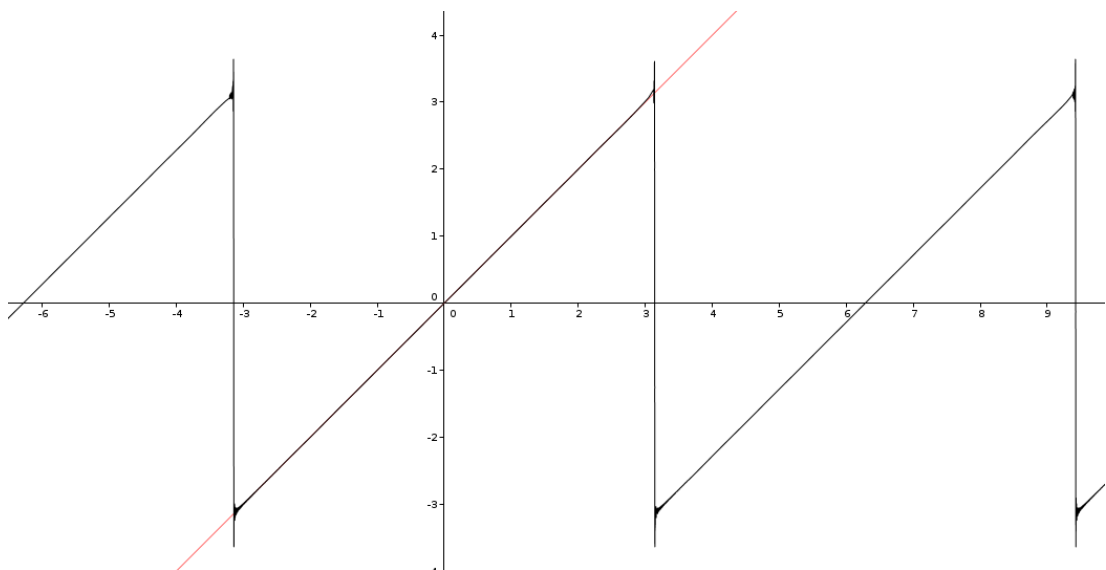




Nähert man die Funktion mit 100 Partialsummen oder 1000 Partialsummen an, ist der Unterschied zur Ausgangsfunktion kaum zu erkennen. Nur an Sprungstellen gibt es Probleme; die Fourierannäherung muss dabei in der Nähe der Stelle „ausholen“, um diese überbrücken zu können.



Fourierannäherung an  $f(x)=x$  mit 100 Fourierkoeffizienten



Fourierannäherung an  $f(x)=x$  mit 1000 Fourierkoeffizienten

## 3 Anwendungsmöglichkeiten

### 3.1 Spektralanalysen

#### 3.1.1 Strahlungsmessung

Die Fourieranalyse wird von Spektralapparaten eingesetzt, um das Frequenzspektrum von elektromagnetischer Strahlung zu messen. Dabei wird Gamma-, Röntgen- optische oder Hochfrequenzstrahlung in die harmonischen Elementarschwingungen zerlegt, deren Amplituden das Frequenzspektrum bilden.<sup>10</sup>

#### 3.1.2 Stimmgeräte

Akustik ist ein großes Einsatzgebiet der Fourierreihe. Der größte Fourierkoeffizient einer Schwingung, die über ein Mikrofon aufgenommen wird, gehört zu der lautesten Teilschwingung. Aus deren Frequenz kann der gespielte Ton und dessen Differenz zum Stimmtton bestimmt werden.<sup>11</sup> Siehe Kapitel 4.

### 3.2 Signale filtern und speichern

Um Signale zu filtern oder platzsparend zu speichern, müssen unwichtige von wichtigen Schwingungen unterschieden werden. Da die Fourierreihe eine zusammengesetzte Schwingung wieder in ihre Einzelkomponenten zerlegt, ist sie gut dafür geeignet.

#### 3.2.1 EKGs

Für Untersuchungen der Herzfrequenz sind Fourierreihen gut geeignet, da Messrauschen ignoriert wird und die Frequenz der stärksten Schwingung, des Herzschlags, direkt bestimmt werden kann.<sup>12</sup>

#### 3.2.2 Tonlagenveränderung

Digitale Equalizer zerlegen Töne mit der Fourierreihe in ihre Elementarschwingungen und verändern deren Amplituden je nachdem, welche Tonlage besonders betont werden soll. Danach werden die Elementarschwingungen wieder zu einer einzigen Schwingung zusammengesetzt.<sup>13</sup>

Das digitale „Pitching“ von Musik erfolgt ebenfalls über eine Zerlegung der Schwingung in Elementarschwingungen, einer Streckung oder Stauchung dieser um einen konstanten Faktor und dem Zusammensetzen der Einzelschwingungen zu der Ausgangsschwingung mit neuer Tonlage.<sup>14</sup>

Konkrete Anwendungen finden sich auch in Autos, die ihre Motorgeräusche aufzeichnen und ein Gegengeräusch erzeugen, um die kleinen Frequenzen auszugleichen. Das Auto erhält dadurch einen tieferen, „besseren Sound“.

Auch Spracherkennungssysteme verwenden Fourierreihen, um die wichtigen Töne vom Umgebungslärm zu isolieren und charakteristische Sprachmerkmale zu erkennen.<sup>15</sup>

<sup>10</sup>Eintrag „Spektralapparate“, Brockhaus Enzyklopädie

<sup>11</sup>Fourieranalyse - Kapitel 3. Audio

<sup>12</sup>Entwicklung eines einfachen EKG-Gerätes mit Webinterface unter Verwendung eines Entwicklungsboards mit ARM9 MCU, S. 4f

<sup>13</sup>Equalizing, Metering and the FFT

<sup>14</sup>Fourieranalyse - Kapitel 3. Audio

<sup>15</sup>Speech Signal Representation, Folie 2, 3, 10

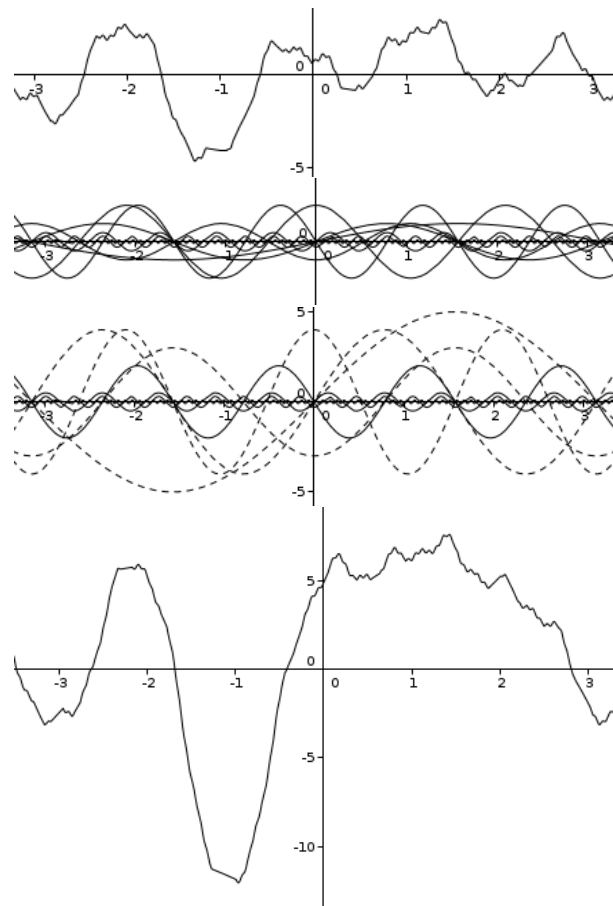


Abbildung 1: Tonveränderung: Die Ausgangsschwingung (1) wird in Elementarschwingungen zerlegt (2). Die Amplituden der zu betonenden Schwingungen - in diesem Fall die mit niedriger Frequenz - werden verstärkt (3). Die neuen Elementarschwingungen werden wieder zu einer Tonschwingung zusammengesetzt (4). Im Gegensatz zu einer gesamten vertikalen Streckung der gesamten Tonschwingung sieht man, dass nur die tiefen Töne lauter wurden und kleine Schwingungen kaum verändert wurden.

### 3.2.3 Kompression von Bild- und Tonmaterial

Will man Ton-, Bild- oder Videodaten speichern, können die Daten mittels Fourierreihen komprimiert werden, da es nur auf die ungefähre Korrektheit aller Datenpunkte ankommt.

Moderne Bildkomprimierungs- und -speicherungsverfahren wie JPEG nutzen Fourierreihen, um die Bilddaten zu komprimieren. Nachdem die Bilder in kleinere Teile unterteilt wurden, werden zu den einzelnen Datenpunkten, den Pixeln des Bildes, die Fourierkoeffizienten erzeugt und gespeichert. Abhängig vom Kompressionslevel, also je nachdem, wie viele Fourierkoeffizienten man speichert, ist die Qualität des Bildes mehr oder weniger gut.

Das Weglassen der hinteren Reihenglieder mit niedriger Amplitude verändert das Bild nicht stark, resultiert jedoch manchmal in einem sichtbaren Bildrauschen, welches zum Beispiel bei stark komprimierten JPEG-Bildern zu beobachten ist.<sup>16</sup>

Insbesondere um harte Kanten kann man häufig ein Wellenmuster erkennen. Wie bei der Sägezahnfunktion müssen die Einzelschwingungen vorher „Anlauf nehmen“, um große Sprünge zu schaffen und pendeln daher auch neben den Kanten schon sichtbar hin und her.

<sup>16</sup> Dokumentation Deutsche Schülerakademie Braunschweig 2, Pixelschubser, S. 46f, JPEG-Kompression



Abbildung 2: Um die Bäume am Horizont entsteht bei starker Kompression ein Wellenmuster.

Tonspeicherung bei MP3 funktioniert ähnlich, wobei nur eindimensionale Daten, nämlich die Ausschläge der Tonschwingung, gespeichert werden müssen. Vom Menschen nicht wahrnehmbare Schwingungen und Schwingungen mit geringer Amplitude können dabei weggelassen werden. Auch hier führt das Speichern von zu wenigen Fourierkoeffizienten zu einem Rauschen, das akustisch wahrnehmbar ist. Auch CDs speichern nur die Koeffizienten der durch die Fourierreihe erzeugten Elementarschwingungen. CD-Spieler setzen sie dann wieder zu einer Schwingung zusammen.

### 3.2.4 Streamen von Multimediainhalten

Kompression durch Fourierreihen spielt beim Streamen von Musik und Filmen eine große Rolle, um Daten in möglichst guter Qualität live übertragen zu können.

Bei Radio- und Fernsehsignalen werden die Fourierkoeffizienten der Daten über die Trägerwelle gesendet. Das kann zu flimmernden Bildstörungen bei sehr feinen Details führen.<sup>17</sup>

Beim Streamen von Musik und Filmen über das Internet werden, je nachdem, wie gut die Verbindung ist, unterschiedlich viele Fourierreihenglieder heruntergeladen, bevor die Stelle abgespielt wird. Das Übertragen durch Fourierreihenglieder ermöglicht dabei eine stufenlose Verbesserung der Qualität über die Zeit. Bei Änderung der Netzwerkgeschwindigkeit wird die Qualität so automatisch hoch- oder heruntergestuft und der Film stockt nicht.

### 3.2.5 Bildverarbeitung

Werden Bilder in Fourierreihen umgewandelt, lassen sie sich einfacher vergleichen, da man „kleine und große“ Bildinformationen unterscheiden kann. In der Makro- und Mikrofotografie werden oft mehrere leicht versetzte Bilder gemacht. Als Fourierreihen lassen sich die einzelnen Bilder dann so verschieben, dass die Differenzfunktion minimal wird. Würde man mit simpler Informatik versuchen, die Bilder passgenau übereinander zu legen, müsste man die Bilder jeweils Pixel für Pixel verschieben und die Differenz der Daten ermitteln. Die Bilder werden schließlich

<sup>17</sup> Anwendung der Fourier-Analyse in der Praxis

verschoben übereinandergelegt und Dreck auf der Linse mittelt sich heraus.<sup>18</sup>

### 3.3 Erzeugung periodischer Signale

...

### 3.4 Lösung von Differenzialgleichungen

Einige in der Physik vorkommende Gleichungen, bei denen Ableitungen oder Integrale mit der Ausgangsfunktion gleichgesetzt werden, lassen sich nur schwer lösen. Dabei hilft die Darstellung als Fourierreihe. Da sie nur aus Sinus- und Kosinustermen besteht, lässt sie sich leicht integrieren und ableiten.<sup>19</sup>

## 4 Konkretes Anwendungsbeispiel

Um Erfahrung mit der konkreten Anwendung der Fourierreihe zu gewinnen, habe ich ein Stimmgerät programmiert. In Python gibt es eine Bibliothek, die das Ansprechen des Mikrofons vereinfacht. Dazu werden in einer bestimmten Frequenz Werte des Tonschwingungsausschlages vom Mikrofon abgefragt. Diese können - in Paketen gebündelt - in (nahezu) Echtzeit abgefragt werden.

Um die Fourierreihe zu berechnen, muss die Ausgangsfunktion integrierbar sein. Da die Integrale der Ausgangsfunktion hier als Summe aller gemessenen Werte betrachtet werden können, ist es möglich, zu diesen Werten die Fourierkoeffizienten zu bilden und in einer Liste zu speichern.

Der maximale Wert in der Liste ist der Koeffizient der Schwingung, die den stärksten Einfluss auf den aufgenommenen Ton hat. Die Frequenz der lautesten Elementarschwingung lässt sich also herausfinden, indem man die Position des Maximums in der Liste betrachtet.

Damit hat man die Frequenz des gespielten Tons herausgefunden. Im Anhang befindet sich der Quelltext des Programms.

## 5 Zusammenfassung

Mit der Eigenschaft jede Funktion beliebig genau anzunähern ist die Fourierreihe ein wichtiges Werkzeug.

## 6 Anhang

### 6.1 Python-Skript als Stimmgerät

---

```

1 #!/usr/bin/python
2 import alsaaudio, time, audioop, cv2
3 import numpy as np
4
5 rate = 8000
6 periodsize = 800
7 samplesize = 2

```

---

<sup>18</sup> *Dokumentation Deutsche Schülerakademie Braunschweig 2, Pixelschubser*, S. 42ff, Image Stacking

<sup>19</sup> *Komplexe Signale aus natürlichen Schwingungen*

```
8  fouriervalues = 100
9
10 # Fourierkoeffizienten berechnen
11 def fourier(values, n):
12     # Listen erzeugen
13     ak = np.zeros(n)
14
15     ak[0] = np.sum(values) / float(periodsize)
16     for ii in range(1, n):
17         ak[ii] = np.sum(map(lambda x: values[x]*np.cos(ii*x), range(periodsize)))
18
19 # Mikrofoneinstellungen: Mono, 8000 Hz, 16 bit little endian samples
20 inp = alsaaudio.PCM(alsaaudio.PCM_CAPTURE,0)
21 inp.setchannels(1)
22 inp.setrate(rate)
23 inp.setformat(alsaaudio.PCM_FORMAT_S16_LE)
24 inp.setperiodsize(periodsize)
25
26 key = 0
27 while key != 27:
28     # Mikrofondaten einlesen
29     start = time.time()
30     l,data = inp.read()
31     end = time.time()
32     if (end - start) < 0.01:
33         continue
34
35     if l:
36         length = 1.0/float(rate/periodsize)
37
38         # Fourieranalysieren
39         values = map(lambda x: audioop.getsample(data, samplesize, x), range(periodsize))
40         ak = fourier(values, fouriervalues)
41
42         for ii in range(len(ak)):
43             if ak[ii] < 0:
44                 ak[ii] = -ak[ii]
45
46         loudest = np.argmax(ak)*(1/length)
47         if loudest > 370 and loudest < 700:
48             print loudest
49         elif loudest >= 700:
50             print loudest/2
51
52     time.sleep(0.02)
```

---

## 6.2 Python-Skript zur Berechnung der Fourierkoeffizienten

Die Formeln wurden dabei in Python-Code in den Zeilen 21, 23 und 24 umgesetzt. In einer Datei werden schließlich die Graphen der Originalfunktion und der Annäherung ausgegeben.

---

```

1  #!/usr/bin/env python
2  # -*- coding: utf8 -*-
3  # Programm zur Bestimmung der Fourierkoeffizienten einer beliebigen Funktion
4  import numpy as np
5
6  # Anzahl der Fourierkoeffizienten
7  # und Breite der Integralbalken
8  n, dy = 3, 0.001
9
10 # Beliebige Funktion
11 def funktion(x):
12     return x
13
14 # Funktion, die das Integral ausrechnet
15 def integral(funktion, untere, obere):
16     return np.sum(map(funktion, np.arange(untere, obere, dy)))*dy
17
18 # Fourierkoeffizienten berechnen
19 print "Fourierkoeffizienten berechnen"
20 ak, bk = np.zeros(n), np.zeros(n)
21 ak[0] = integral(funktion, -np.pi, np.pi) / (2*np.pi)
22 for ii in range(1, n):
23     ak[ii] = integral(lambda x: funktion(x)*np.cos(ii*x), -np.pi, np.pi) / np.pi
24     bk[ii] = integral(lambda x: funktion(x)*np.sin(ii*x), -np.pi, np.pi) / np.pi
25 print ak, bk
26
27 # Fourierfunktion, Summe von Sinus- und Kosinustermen
28 def fourierfunktion(ak, bk, x):
29     result = 0
30     for (n,ii) in enumerate(ak):
31         result += ii*np.cos(n*x)
32     for (n,ii) in enumerate(bk):
33         result += ii*np.sin(n*x)
34     return result
35
36 # Ganz viele Funktionswerte berechnen
37 print "Funktionen plotten"
38 x = np.arange(-10, 10, dy)
39 original = map(funktion, x)
40 fourier = map(lambda x: fourierfunktion(ak, bk, x), x)
41
42 # In Datei ausgeben
43 f=open("fourier.txt","w")
44 for (x,y1,y2) in zip(x, original, fourier):
45     f.write("\t".join(map(str,[x,y1,y2]))+"\n")
46 f.close()

```

---

## 7 Literaturverzeichnis

**Théorie Analytique de La Chaleur**, Joseph Fourier

**Mathematik verstehen und anwenden**, Steffen Goebbels, Stefan Ritter, 2011, Springer Verlag

**Fourierreihen**, Katja Breitenstein, Daniel Frischer, 2009

**Komplexe Signale aus natürlichen Schwingungen**, Michel Gaedtke,  
[www.gaedtke.name/physik/komplexe-signale-aus-naturlichen-schwingungen](http://www.gaedtke.name/physik/komplexe-signale-aus-naturlichen-schwingungen) (26.12.2016)

**Taschenbuch der Mathematik**, Bronstein, Semendjajew, 1987, Verlag Harri Deutsch

**Höhere Mathematik 2**, Meyberg, Vachenaue, 1999<sup>4</sup>, Springer Verlag

**Analysis 1 - Differential- und Integralrechnung einer Veränderlichen**, Otto Forster, 1988<sup>4</sup>, Vieweg & Sohn Verlagsgesellschaft

**Elektroaufgaben - Leitungen, Vierpole, Fourier-Analyse, Laplace-Transformation**, Helmut Lindner, 2007<sup>6</sup>

**Eine Anwendung: Fourieranalyse**, Prof. M. Seibt, Pascal Guderian, 2014,  
[lp.uni-goettingen.de/get/text/4937](http://lp.uni-goettingen.de/get/text/4937) (26.12.2016)

**Munzinger Online / Brockhaus Enzyklopädie**, <sup>21</sup>  
[www.munzinger.de](http://www.munzinger.de) (25.12.2016; Zusammenstellung von der Stadtbücherei Gladbeck)

**Entwicklung eines einfachen EKG-Gerätes mit Webinterface unter Verwendung eines Entwicklungsboards mit ARM9 MCU**, Philipp Meißner

**Fourieranalyse - Kapitel 3. Audio**,  
<http://www.pd-tutorial.com/german/ch03s08.html> (08.02.2017)

**Equalizing, Metering and the FFT**, 2015,  
<http://www.zytrax.com/tech/audio/equalization.html> (08.02.2017)

**Speech Signal Representation**, MIT Lecture, 2003,  
<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-345-automatic-speech-recognition-spring-2003/lecture-notes/lecture5.pdf>

**Dokumentation Deutsche Schülerakademie Braunschweig 2**, Pixelschubser, 2016, Amelie Hintermaier, Selina Dilara Soysal, Moritz Andres, Margarete Ketelsen, Lukas Veit, Jakob Kronawitter, Emily Hauck, Lorna Espiga Santos, Phil Luca Radzik, Tino Beste, Dominik Suelmann, Johanna Kilo, Katharina Schurk, Amelie Hintermaier, Louis Fischer, Marcel Garus

**Technische Mechanik 4**, Gross, Hauger, Schnell, Wriggers, 1995<sup>2</sup>

**Mathematik für Ingenieure und Naturwissenschaftler - Übungen**, Lothar Papula, 1994<sup>3</sup>

**Anwendung der Fourier-Analyse in der Praxis**, Judith Preiner, 2005,  
[https://archive.geogebra.org/de/examples/fourier/sachanalyse/anwendung\\_fourier-analyse.htm](https://archive.geogebra.org/de/examples/fourier/sachanalyse/anwendung_fourier-analyse.htm)



## 8 Selbstständigkeitserklärung

Ich erkläre, dass ich die Facharbeit ohne fremde Hilfe angefertigt habe und nur die im Literaturverzeichnis angeführten Quellen und Hilfsmittel benutzt habe. Ich bin damit einverstanden, dass die Schule meine Facharbeit gegebenenfalls öffentlich zugänglich macht.