

# 17

## Fourier-Reihe mit Sinus und Cosinus. Fast Fourier Transform

Jörn Loviscach

Versionsstand: 20. März 2012, 16:00

Die nummerierten Felder sind absichtlich leer, zum Ausfüllen in der Vorlesung.

Videos dazu: <http://www.j3L7h.de/videos.html>



This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Germany License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/de/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

### 1 Fourier-Reihe mit Sinus und Cosinus

Gegeben sei eine reell- oder komplexwertige Funktion  $f$  mit Periode  $T$ . Dann kann man die in eine komplexe Fourier-Reihe entwickeln:

1

---

Dabei sind die  $c_n$  die komplexen Fourier-Koeffizienten:

2

---

Das kann man alles mit Sinus und Cosinus umschreiben, indem man die Eulersche Identität anwendet. Dabei ergibt sich die Fourier-Reihe mit Sinus und Cosinus:

3

Nicht wundern: Dass  $a_0$  mit dem Faktor  $1/2$  steht, macht nachher einige Formeln einfacher. Und Vorsicht mit der Reihenfolge: Die  $a_n$  stehen mit dem Cosinus.

Zwei Unterschiede zur komplexen Fourier-Reihe:

- Es tauchen keine negativen Frequenzen mehr auf.
- Die Amplitude und die Phase der  $n$ -ten Oberwelle ist nun in  $a_n$  und  $b_n$  versteckt statt im Betrag und Winkel von  $c_n$ .

Ist  $f$  eine gerade (und weiterhin periodische!) Funktion, d. h.  $f(-t) = f(t)$  für alle  $t$ , dann kann offensichtlich kein Sinus vorkommen, also sind alle  $b_n$  gleich null. Umgekehrt kann in einer ungeraden Funktion  $f$ , d. h.  $f(-t) = -f(t)$  für alle  $t$ , kein Cosinus vorkommen, also sind alle  $a_n$  gleich null.

## 2 Fourier-Koeffizienten für Sinus und Cosinus

Man könnte aus  $c_n$  und  $c_{-n}$  die Fourier-Koeffizienten  $a_n$  und  $b_n$  bestimmen. Aber es gibt auch einen direkten Weg. Dazu überlegt man sich Folgendes:

$$\int_0^T (\cos(2\pi nt/T))^2 dt = \left[ \right] \quad \text{für alle } n = 1, 2, 3, \dots,$$

denn:

5

$$\int_0^T (\sin(2\pi nt/T))^2 dt = \left[ \right] \quad \text{für alle } n = 1, 2, 3, \dots,$$

denn:

$$\int_0^T \sin(2\pi nt/T) \sin(2\pi mt/T) dt = \begin{cases} 0 & \text{für alle } n, m = 1, 2, 3, \dots \text{ mit } n \neq m, \\ T/2 & \text{für } n = m \end{cases}$$

denn mit zweifacher partieller Integration ergibt sich:

Und entsprechend:

$$\int_0^T \sin(2\pi nt/T) \cos(2\pi mt/T) dt = \begin{cases} 0 & \text{für alle } n, m = 1, 2, 3, \dots \text{ mit } n \neq m \\ T/2 & \text{für } n = m \end{cases}$$

und das sogar für  $n = m$  (einfache partielle Integration) sowie

$$\int_0^T \cos(2\pi nt/T) \cos(2\pi mt/T) dt = \begin{cases} 0 & \text{für alle } n, m = 1, 2, 3, \dots \text{ mit } n \neq m. \\ T/2 & \text{für } n = m \end{cases}$$

Daran sieht man, dass sich die Sinus- und Cosinus-Funktionen der Fourier-Reihe fast so verhalten wie die Funktionen  $t \mapsto e^{2\pi i n t}$  der komplexen Fourier-Reihe. Sie stehen senkrecht aufeinander und haben alle die gleiche Norm („Länge“).

Wenn man also  $f(t)$  mit  $\cos(2\pi mt/T)$  integriert ( $m = 1, 2, 3, \dots$ ), wird man erhalten:

Also muss für die Fourier-Koeffizienten  $a_n$  gelten:

Durch den Trick mit dem  $\frac{1}{2}a_0$  gilt diese Formel auch für den Gleichspannungsanteil  $n = 0$ . Entsprechend gibt für die Fourier-Koeffizienten  $b_n$ :

---

<sup>14</sup>

### 3 Fast Fourier Transform (FFT)

Typischerweise hat man Signale als Folgen von Messwerten (Samples) gegeben statt als kontinuierliche Funktionen (Demo mit Audacity).

---

<sup>15</sup>

Dann lassen sich die Fourier-Koeffizienten mit Summen über die Samples statt mit Integralen berechnen (Diskrete Fourier-Transformation, DFT). Indem man diese Summen geschickt zusammenfasst, kann man die Rechnung beschleunigen (Fast Fourier Transform, FFT). Dies ist die übliche Art, Fourier-Analyse zu betreiben. Dafür stehen auch diverse Programmierbibliotheken bereit, insbesondere FFTW ([Link](#)).

DFT/FFT beziehen sich immer nur auf einen endlichen Ausschnitt eines Signals (Demo mit Audacity). Diese Ausschnitte werden weich mit „Fensterfunktionen“ [window functions] aus dem Signal gebildet. Harte Schnitte würden unsinnige Anteile mit hohen Frequenzen in der DFT/FFT erzeugen:

---

<sup>16</sup>

MATLAB<sup>®</sup> beherrscht selbstverständlich FFT und Fensterfunktionen.