

# Fourier-Methoden: Theorie und Anwendungen

Felix Wager

11. September 2025

## Inhaltsverzeichnis

<b>1</b>	<b>Abstract</b>	<b>3</b>
<b>2</b>	<b>Einleitung</b>	<b>3</b>
2.1	Motivation . . . . .	3
2.2	Zielsetzung der Arbeit . . . . .	3
2.3	Aufbau der Arbeit . . . . .	3
<b>3</b>	<b>Mathematische Grundlagen</b>	<b>3</b>
3.1	Komplexe Zahlen und die Eulerformel . . . . .	3
3.2	Das Skalarprodukt von Funktionen und Orthonormalsysteme .	4
3.3	Die Fourierreihe . . . . .	5
3.4	Von der Fourierreihe zur Fouriertransformation . . . . .	8
3.5	Die Fourierre Transformation in der Praxis: Die DFT . . . . .	9
3.6	Die Fourierre Transformation in der Praxis: Die FFT . . . . .	9
<b>4</b>	<b>Fourier-Reihe</b>	<b>12</b>
4.1	Herleitung der Reihe . . . . .	12
4.2	Berechnung der Koeffizienten . . . . .	12
4.3	Eigenschaften und Konvergenz . . . . .	12
<b>5</b>	<b>Von der Reihe zur Fourier-Transformation</b>	<b>12</b>
5.1	Übergang zum Integral . . . . .	12
5.2	Fourier-Transformation und inverse Transformation . . . . .	12
5.3	Eigenschaften . . . . .	12
<b>6</b>	<b>Diskrete Fourier-Transformation (DFT)</b>	<b>13</b>
6.1	Definition und Motivation . . . . .	13
6.2	Herleitung aus der Fourier-Transformation . . . . .	13

6.3	Beweise für die Korrektheit . . . . .	13
6.4	FFT als effiziente Berechnung . . . . .	13
<b>7</b>	<b>Eigene Beiträge</b>	<b>13</b>
7.1	Eigene Herleitung der DFT . . . . .	13
7.2	Beweise . . . . .	13
7.3	Eigener FFT-Algorithmus in Python . . . . .	13
7.4	Audio-Programm zur Echtzeit-Visualisierung . . . . .	13
7.5	Bildverarbeitung: Moiré-Muster entfernen mit 2D-FFT . . . . .	13
<b>8</b>	<b>Anwendungen</b>	<b>13</b>
8.1	Audio . . . . .	13
8.2	Bild . . . . .	13
<b>9</b>	<b>Anhang</b>	<b>13</b>
9.1	Eigene Beweise . . . . .	13
9.2	Bedeutung im größeren Kontext . . . . .	16
<b>10</b>	<b>Fazit und Ausblick</b>	<b>16</b>
10.1	Zusammenfassung der Ergebnisse . . . . .	16
10.2	Ausblick: Erweiterungen und Anwendungen . . . . .	16

# 1 Abstract

## 2 Einleitung

### 2.1 Motivation

Warum Fourier-Methoden heute so wichtig sind.

### 2.2 Zielsetzung der Arbeit

### 2.3 Aufbau der Arbeit

## 3 Mathematische Grundlagen

### 3.1 Komplexe Zahlen und die Eulerformel

Um sich die Arbeit mit Fourier Transformationen, Reihen und sonstigem erheblich zu erleichtern ist es sehr sinnvoll mit komplexen Zahlen zu arbeiten. Doch was sind komplexe Zahlen? Komplexe Zahlen sind prinzipiell nichts anderes, als eine Erweiterung der Menge der Reellen Zahlen, welche es ermöglicht die Wurzel von negativen Zahlen zu ziehen. Dafür wird die Wurzel von  $-1$  als die imaginäre Einheit  $i$  definiert. Mathematisch korrekt sieht das wie folgt aus:

$$i^2 := -1$$

Die Menge der komplexen Zahlen wird mit dem Symbol  $\mathbb{C}$  abgekürzt. Eine typische komplexe Zahl hat in der algebraischen Schreibweise die Form:

$$z = a + ib \quad \text{mit} \quad a, b \in \mathbb{R}$$

$a$  ist hierbei der Realteil  $\Re(z)$  und  $b$  der Imaginärteil  $\Im(z)$  von  $z$ .

Um mit komplexen Zahlen bei Fourier Reihen zu arbeiten, benötigt man auch ein paar Rechenoperationen. Die wichtigsten beiden sind hier der Betrag und das komplex konjugierte einer komplexen Zahl. Eine komplexe Zahl lässt sich auch als Vektor im  $\mathbb{R}^2$  betrachtet, so ist dann der Betrag als euklidische Norm dieses Vektors definiert.

$$|z| := \|z\| = \sqrt{a^2 + b^2}$$

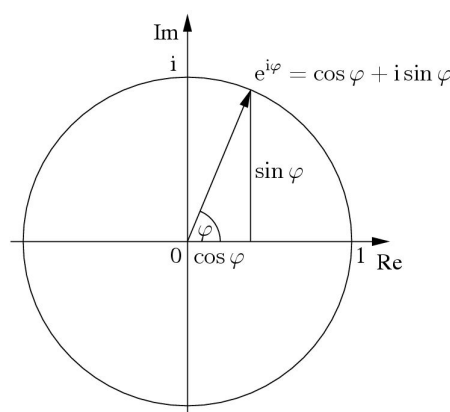
Bedeutet in einem Koordinatensystem, bei dem der Real- und Imaginärteil einer Zahl auf den x- und y-Achsen festgehalten wird, ist der Betrag die Länge vom Ursprung bis zum Punkt im Koordinatensystem dieser Zahl. Und

das komplex konjugierte einer Zahl ist eine Abbildung, welche lediglich den Imaginärteil mit  $-1$  multipliziert:

$$\bar{\cdot} : \mathbb{C} \rightarrow \mathbb{C}, z = x + iy \mapsto \bar{z} := x - iy$$

Mithilfe von komplexen Zahlen und den Taylorreihen von Sinus, Cosinus und der Exponentialfunktion  $e^x$  kann man nun die Eulerformel herleiten.

Wenn man den Betrag dieses Ausdrucks bildet, so fällt auf, dass hier das Ergebnis, unabhängig von  $x$ , 1 ist. Dies bedeutet, dass  $e^{ix}$  als ein gegen den Uhrzeigersinn rotierender Vektor, für steigende Werte für  $x$ , gesehen werden kann. Im vorher genannten Koordinatensystem, welches man auch die Gaußsche Zahlenebene nennt, sieht das so aus:



### 3.2 Das Skalarprodukt von Funktionen und Orthonormalsysteme

Des Weiteren spielen Orthonormalsysteme eine große Rolle, wenn man sich mit der Fourier Analyse beschäftigt. Allgemein lässt sich sagen, dass ein Orthonormalsystem eine Menge von Vektoren oder Funktionen, aus einem Vektorraum mit Skalarprodukt, sind, welche sowohl orthogonal zueinander, aber auch normiert zu sich selbst sind. Orthogonal sind sie, wenn das Skalarprodukt zweier unterschiedlicher Vektoren 0 ergibt und normiert, wenn das Skalarprodukt eines Vektors mit sich selbst 1 ergibt. Für alle Vektoren  $v_n$  im  $\mathbb{R}^n$  muss also folgendes gelten, damit die Menge der Vektoren ein Orthonormalsystem bildet:

$$1. \text{ Orthogonalität: } \langle v_i, v_j \rangle = 0 \quad \forall i \neq j$$

$$2. \text{ Normiertheit: } \langle v_i, v_j \rangle = \sum_{i=1}^n v_i^2 = \|v_i\|^2 = 1 \quad \text{mit } i = j$$

$$\left( \text{Skalarprodukt: } \langle v, w \rangle := \sum_{i=1}^n v_i * w_i \right)$$

Hier sieht man auch, dass das Skalarprodukt eines Vektors mit sich selbst, das gleiche ist wie die quadrierte euklidische Norm des Vektors, wodurch der Begriff der Normiertheit anschaulicher wird. Wie schon erwähnt lassen sich diese Eigenschaften auch auf Funktionen anwenden. Hierfür definiert man die Normiertheit und die Orthogonalität auch, exakt gleich wie bei Vektoren, über das Skalarprodukt. Das Skalarprodukt für zwei Funktionen ist wie folgt definiert:

$$\langle v, w \rangle := \int_a^b v(x) * w(x) dx \quad \text{für } v, w : [a, b] \rightarrow \mathbb{R}$$

Falls eine Menge von Funktionen Orthogonalität und Normiertheit erfüllt, ist diese Menge auch ein Orthogonalsystem. Anschaulich kann man sich die Funktionen  $v(x)$  und  $w(x)$  noch als zwei Vektoren mit unendlich vielen Dimensionen vorstellen, wobei der  $x$  Wert angibt in welcher Dimension man sich befindet. Da das Skalarprodukt die jeweiligen Dimensionen von Vektoren multipliziert und diese schließlich aufsummiert, macht es Sinn, dass man bei Funktionen ähnlich vorgeht. So lässt sich also die Erweiterung der Summe zum Integral erklären. Die Formel für die Norm von Funktionen  $\left( \|f\| = \sqrt{\int_a^b |f(x)|^2 dx} \right)$  ergibt sich, wenn man das für Funktionen definierte Skalarprodukt ähnlich wie bei Vektoren auf die Funktion selbst anwendet. Im komplexen Fall wird das Skalarprodukt leicht angepasst, indem der zweite Faktor komplex konjugiert wird:

$$\langle v, w \rangle := \int_a^b v(x) * \overline{w(x)} dx \quad \text{für } v, w : [a, b] \rightarrow \mathbb{C}$$

### 3.3 Die Fourierreihe

Der erste große Schritt, um die Fourier Transformation herzuleiten, ist die Fourier Reihe. Eine Reihe selbst ist in der Mathematik ein Begriff für eine unendliche Summe von Termen. Die Fourier Reihe ist hierbei eine besondere Reihe. Ihr Sinn ist es periodische Funktionen mithilfe von Sinus- und Kosinustermen zu approximieren. Joseph Fourier hat in seinem Werk „Théorie analytique de la chaleur“ schließlich auch bewiesen, dass jede periodische Funktion auf diese Weise dargestellt werden kann. Die Approximation selbst geschieht durch trigonometrische Polynome, mit welchen man später die Fourierreihe einer Funktion bildet. Ein trigonometrisches Polynom ist hier eine Funktion der Form:

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

Ziel ist es nun die Faktoren  $a_k$  und  $b_k$  in Abhängigkeit zur anzunähernden Funktion zu bestimmen. Denn durch die Zählervariable  $k$ , welche die Periodenlänge der Sinus- und Kosinusterme bestimmt, kann man durch die Faktoren  $a_k$  und  $b_k$  festlegen wie dominant die Anteile der Sinus und Kosinusterme, mit der jeweiligen Periodenlänge, in der zu approximierende Funktion sind. Um die Berechnung der Faktoren kompakter zu gestalten, kann man mit dem Zusammenhang  $2\cos x = e^{ix} + e^{-ix}$  und  $2i\sin x = e^{ix} - e^{-ix}$ , einer Umstellung der Eulerformel, nun das trigonometrische Polynom zu dem komplexen trigonometrischen Polynom zusammenfassen:

$$p(x) = \sum_{k=-n}^n c_k e^{ikx} \quad \text{für geeignete } c_k \in \mathbb{C}$$

Um jetzt den komplexen Faktor  $c_k$ , zunächst für  $2\pi$  periodische Funktionen, zu berechnen, verwendet man ein Orthonormalsystem, welches aus den Funktionen  $\phi_n(x)$  besteht:

$$\phi_n(x) : [-\pi, \pi] \rightarrow \mathbb{C}, \quad \phi_n(x) = \frac{1}{\sqrt{2\pi}} e^{inx} \quad \text{mit } n \in \mathbb{Z}$$

Dass die Menge an Funktionen  $\phi_n$  ein Orthonormalsystem ist, habe ich im Anhang gezeigt (Beweis A. 1). Um endlich  $c_k$  zu berechnen, setzt man zunächst die zu approximierende Funktion  $f$ , mit der Periodizität  $2\pi$ , mit dem trigonometrischen Polynom gleich und schränkt sie zudem ein:

$$f : [-\pi, \pi] \rightarrow \mathbb{C}, \quad f(x) = \sqrt{2\pi} * \sum_{k=-n}^n c_k \phi_k(x)$$

Anschließend bildet man das Skalarprodukt von  $f$  und  $\phi_m$ , was möglich ist, da  $f$  auf der Definitionsbereich von  $f$  und  $\phi_m$  gleich ist.

$$\langle f, \phi_m \rangle = \int_{-\pi}^{\pi} f(x) \overline{\phi_m(x)} dx = \int_{-\pi}^{\pi} \sqrt{2\pi} \sum_{k=-n}^n c_k \phi_k(x) \overline{\phi_m(x)} dx$$

Da Integral und Summe beide linear sind, darf man die Summe mit dem Integral vertauschen. Zudem hängt  $c_k$  nicht von  $x$  ab, wodurch der Faktor  $c_k$  für das Integral eine Konstante ist. Nach einer Abwandlung des Distributivgesetzes darf er somit herausgezogen werden. Man erhält also diesen Ausdruck:

$$\int_{-\pi}^{\pi} f(x) \overline{\phi_m(x)} dx = \sqrt{2\pi} \sum_{k=-n}^n c_k \int_{-\pi}^{\pi} \phi_k(x) \overline{\phi_m(x)} dx$$

Aufgrund dessen, dass die Menge der Funktionen  $\phi_m$  ein Orthonormalsystem ist, ist jeder Summand, außer  $k = m$ , 0. Durch die Normiertheit des Orthonormalsystems bleibt übrig:

$$c_m = \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} f(x) \overline{\phi_m(x)} dx = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{imx} dx$$

Jetzt ist es möglich mit dieser Formel und dem komplexen trigonometrischen Polynom eine  $2\pi$  periodische Funktion zu approximieren. Jedoch wäre es hilfreich, wenn dies für alle Perioden möglich wäre. Um auch Funktionen mit einer Periodizität von  $2L$  annähern zu können, muss man den Faktor  $c_k$  und das komplexe trigonometrische Polynom auf einer Periodizität von  $2L$  ausweiten. Damit das trigonometrische Polynom  $2L$  periodisch wird muss folgendes gelten:

$$e^{i\omega x} = e^{i\omega(x+2L)}$$

$\omega$  gilt es dabei herauszufinden, damit das trigonometrische Polynom  $2L$  periodisch wird:

$$e^{i\omega x} = e^{i\omega(x+2L)} \Leftrightarrow e^{i\omega 2L} = 1 \Leftrightarrow \omega 2L = 2\pi k \Leftrightarrow \omega_k = \frac{k\pi}{L}$$

Das neue trigonometrische Polynom sieht also so aus:

$$\sum_{k=-n}^n c_k e^{i\omega_k x} \Leftrightarrow \sum_{k=-n}^n c_k e^{i \frac{k\pi}{L} x}$$

Um  $c_k$  zu berechnen, nimmt man sich eine  $2L$  periodische Funktion  $f$  und definiert sich eine  $2\pi$  periodische Hilfsfunktion  $g$  in Abhängigkeit von  $f$  wie folgt:

$$g(t) := f\left(\frac{L}{\pi}t\right)$$

Da  $g$  eine  $2\pi$  periodische Funktion ist, kann man mit der hergeleiteten Formel  $c_k$  von  $g$  berechnen. Um die Formel für  $2L$  periodische Funktionen zu erhalten, ist das Ziel die Formel über  $g$  auf  $f$  auszuweiten. Das funktioniert über die Substitutionsregel bei Integralen. Um sie anzuwenden, führen wir die Ableitung der inneren Funktion von  $f$ , also  $\frac{L}{\pi}$  in Form einer 1 ein und erhalten die Formel für  $c_k$ , für die Periodizität  $2L$ :

$$c_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} g(t) e^{-ikt} dt \Leftrightarrow c_k = \frac{1}{2\pi} * \frac{\pi}{L} \int_{-\pi}^{\pi} \frac{L}{\pi} f\left(\frac{L}{\pi}t\right) e^{-i\omega_k t} dt$$

Wendet man jetzt die Substitutionsregel an, erhält man:

$$c_k = \frac{1}{2L} \int_{-L}^L f(x) e^{-i\omega_k x} dx = \frac{1}{2L} \int_{-L}^L f(x) e^{-i \frac{k\pi}{L} x} dx$$

### 3.4 Von der Fourierreihe zur Fouriertransformation

Die Fourierreihe erlaubt es uns, periodische Signale als Summe von Sinus- und Kosinusfunktionen oder komplexen Exponentialfunktionen darzustellen. Doch viele Signale in der Praxis sind nicht-periodisch. Um auch diese Signale in ihre Frequenzanteile zerlegen zu können, verallgemeinern wir das Konzept der Fourierreihe zur Fouriertransformation. Mit der Fouriertransformation werden wir so eine Funktion erhalten, welche uns sagen wird wie groß der Anteil einer Frequenz oder Periodenlänge in einem Signal ist. Dazu nehmen wir die Fourierreihe einer Funktion  $f$  mit der Periode  $2L$  und der Frequenz  $\omega_k$ :

$$f(x) = \sum_{k=-\infty}^{\infty} c_k e^{i\omega_k x}, \quad c_k = \frac{1}{2L} \int_{-L}^L f(x) e^{-i\omega_k x} dx$$

Dazu definieren wir eine Hilfsfunktion  $F(\xi)$ :

$$F : \mathbb{R} \rightarrow \mathbb{C}, \quad F(\xi) = \int_{-L}^L f(x) e^{-i\xi x} dx$$

damit gilt

$$c_k = \frac{1}{2L} F(\omega_k)$$

Setzt man den neu gewonnen Ausdruck für  $c_k$  in die Fourierreihe ein, so erhält man

$$f(x) = \sum_{n=-\infty}^{\infty} F(\omega) e^{i\omega x}$$

Anschließend bildet man  $\Delta\omega$  aus  $\omega_k$ :

$$\omega_k = k \frac{\pi}{L} \Rightarrow \Delta\omega = \frac{\pi}{L} \Rightarrow \frac{1}{2L} = \frac{\Delta\omega}{2\pi}$$

Und setzt die Gleichung auch in die Fourierreihe ein:

$$\frac{1}{2\pi} \sum_{k=-\infty}^{\infty} F(\omega_k) e^{i\omega_k x} \Delta\omega$$

Dies ist eine Riemann Summe. Lässt man nun  $L \rightarrow \infty$  laufen, erhält man ein Integral, welches die inverse Fouriertransformation ist:

$$\lim_{L \rightarrow \infty} \frac{1}{2\pi} \sum_{n=-\infty}^{\infty} F(\omega_k) e^{i\omega_k x} \Delta\omega = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega x} d\omega$$

Die Funktion  $F(\omega)$  selbst ist die Fouriertransformation von  $f$ :

$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i\omega x} dx$$



Man kann das Integral in der Fouriertransformation am besten verstehen, wenn man es als eine Art Durchschnitt betrachtet. Ähnlich wie beim Berechnen eines Mittelwerts summiert das Integral nicht nur eine endliche Anzahl von Werten auf, sondern unendlich viele, unendlich kleine Beiträge von  $f(x)$ . Im Integral steht dabei das Produkt  $f(x) \cdot e^{-i\omega x}$ . Dieses Produkt misst, wie stark die Frequenz  $\omega$  in  $f(x)$  enthalten ist:

- Ist der Anteil der Frequenz  $\omega$  in  $f(x)$  hoch, dann verstärkt  $e^{i\omega_k x}$  den Wert der Funktion  $f(x)$  im Integral konstruktiv und  $F(\omega)$  wird groß.
- Enthält  $f(x)$  diese Frequenz nicht, heben sich die positiven und negativen Anteile im Integral weitgehend auf und  $F(\omega)$  wird klein.

So zeigt die Fouriertransformation, wie groß der Anteil jeder einzelnen Frequenz  $\omega$  in der Funktion  $f$  ist.

### 3.5 Die Fouriertransformation in der Praxis: Die DFT

In der Praxis arbeitet man selten mit kontinuierlichen Funktionen, sondern mit diskreten Messwerten, zum Beispiel bei Audiosignalen oder digitalen Bildern. Um diese Signale in ihre Frequenzanteile zu zerlegen, verwendet man die diskrete Fouriertransformation (DFT). Formell lässt sich die DFT für eine endliche Folge von  $N$  Messwerten

$$x_0, x_1, \dots, x_{N-1}$$

durch die folgende Formel darstellen:

$$X_k = \sum_{j=0}^{N-1} x_n e^{-i2\pi \frac{kj}{N}} \quad \text{mit } k \in [0, N-1]$$

Anzumerken ist aber noch, dass es eine Besonderheit gibt. Wenn das Eingangssignal reell ist, liefert die DFT zwar  $N$  Koeffizienten, jedoch sind die Frequenzen oberhalb von  $N/2$  wegen des Nyquist-Theorems lediglich Spiegelungen der tieferen Frequenzen und enthalten keine neuen Informationen. Warum das so ist, werde ich hier nicht näher eingehen. Wie man die Formel für die DFT herleitet, erkläre ich im nächsten Kapitel.

### 3.6 Die Fouriertransformation in der Praxis: Die FFT

Obwohl die DFT so in der Praxis anwendbar ist, ist sie heutzutage in fast keinem Programm oder Algorithmus zu finden. Denn will man eine Funktion oder Signal mit  $N$  Werten komplett transformieren, dann muss man für  $N$

verschiedene Frequenzen den Wert über die obere Formel berechnen. Also sind es insgesamt  $N^2$  Berechnungen, die man durchführen muss. In der Informatik spricht man für den Abschnitt der DFT in einem Programm von einer Laufzeit von:  $O(N^2)$

Im Jahre 1965 aber, entdeckte James W. Cooley und John W. Tukey eine neue Art die DFT zu berechnen, was die Laufzeit auf  $O(N \log N)$  verringerte. Somit konnte man die DFT erheblich schneller berechnen. Daher kommt auch der Name FFT für den neuen Algorithmus, was für Fast Fourier Transform, also schnelle Fouriertransformation, steht. Um die Umformung gut zu sehen, kann man die diskrete Fouriertransformation etwas umschreiben. Dafür definiere ich die Folge  $(\omega_n)_{n \in \mathbb{N}}$  wie folgt:

$$\omega_n := e^{-2\pi i/n}$$

Schreibt man jetzt die Werte der Ausgangsfunktion  $f$  als Vektor, kann man aufgrund der Struktur der Matrixmultiplikation, diesen Vektor mit einer besonderen Matrix multiplizieren und erhält das Ergebnis der Transformation auch in Form eines Vektors derselben Länge, mit den Werten  $\hat{f}$ .

$$\begin{bmatrix} \hat{f}_0 \\ \hat{f}_1 \\ \hat{f}_2 \\ \vdots \\ \hat{f}_n \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_n & \omega_n^2 & \cdots & \omega_n^{n-1} \\ 1 & \omega_n^2 & \omega_n^4 & \cdots & \omega_n^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \cdots & \omega_n^{(n-1)^2} \end{bmatrix} \cdot \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} = \begin{bmatrix} \omega_n^{0 \cdot 0} & \omega_n^{0 \cdot 1} & \cdots & \omega_n^{0 \cdot (n-1)} \\ \omega_n^{1 \cdot 0} & \omega_n^{1 \cdot 1} & \cdots & \omega_n^{1 \cdot (n-1)} \\ \omega_n^{2 \cdot 0} & \omega_n^{2 \cdot 1} & \cdots & \omega_n^{2 \cdot (n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_n^{(n-1) \cdot 0} & \omega_n^{(n-1) \cdot 1} & \cdots & \omega_n^{(n-1)^2} \end{bmatrix} \cdot \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}$$

Jeder Eintrag im Ergebnisvektor entsteht, indem man die Zahlen in der Zeile der Matrix mit den Zahlen im Vektor multipliziert und die Produkte zusammenzählt. Deshalb muss jeder Eintrag der Matrix aus  $\omega_n$  noch mit  $k$ , der die Frequenz bestimmt, und der Zählvariable  $j$ , die über die Zeilen läuft, als Exponent potenziert werden.

Was Cooley und Tukey herausgefunden haben ist, dass man diesen Ausdruck wie so umschreiben kann:

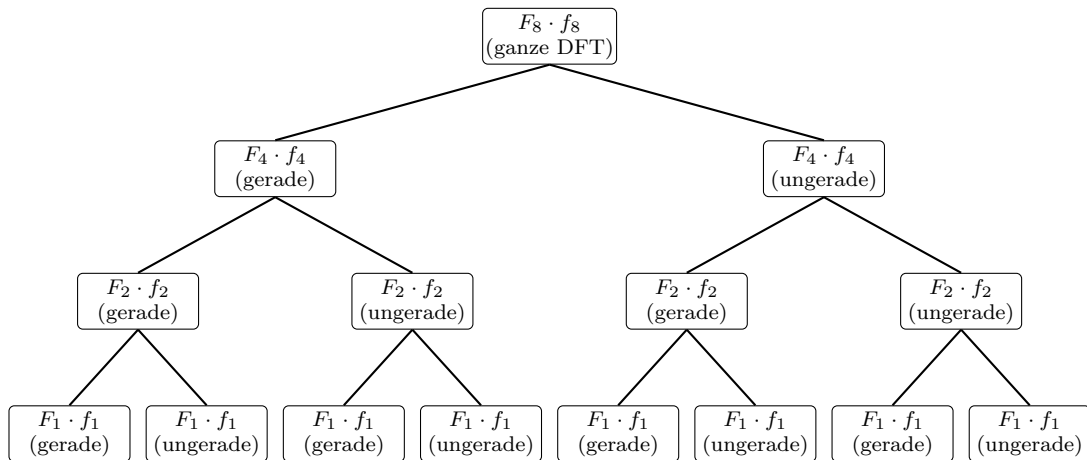
$$\hat{f} = F_N \cdot f = \begin{bmatrix} I_{N/2} & D_{N/2} \\ I_{N/2} & -D_{N/2} \end{bmatrix} \cdot \begin{bmatrix} F_{N/2} & 0 \\ 0 & F_{N/2} \end{bmatrix} \cdot \begin{bmatrix} f_{gerade} \\ f_{ungerade} \end{bmatrix}$$

$F$  ist die DFT-Matrix, mit der der ursprüngliche Wertevektor  $f$  multipliziert wurde.  $I_N$  ist die Einheitsmatrix: Alle Einträge außerhalb der Diagonale sind 0, auf der Diagonale stehen 1, von links oben nach rechts unten.  $D_N$  ist ebenfalls eine Diagonalmatrix wie  $I_N$ , jedoch bestehen die Diagonaleinträge aus  $\omega_N^k$ , wobei  $k$  von 0 bis  $N - 1$  läuft. Der Index der Matrizen gibt jeweils die Anzahl der Zeilen und Spalten an.  $f$  wurde den Indizes nach aufgeteilt.

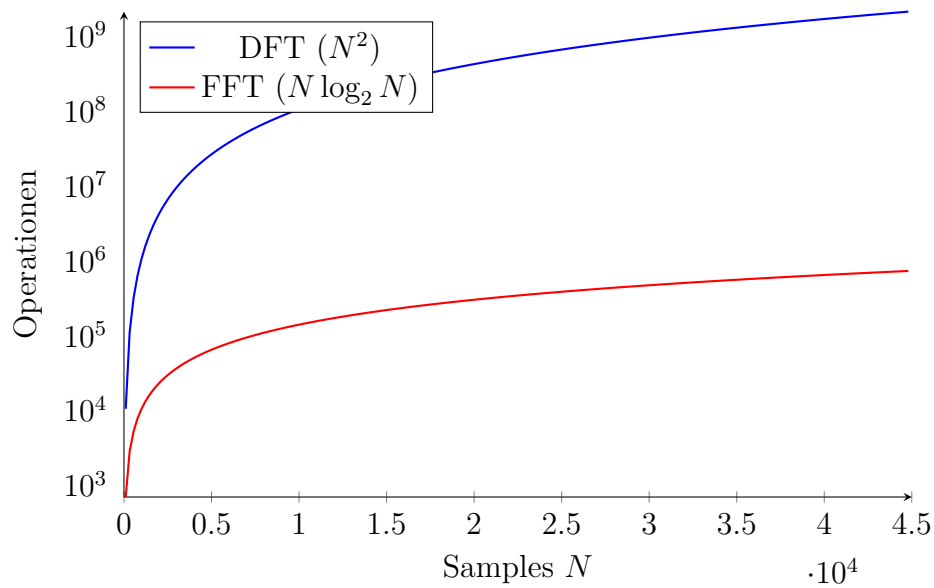
Wichtig ist, diese Formel funktioniert nur, wenn die Anzahl der Werte der Ausgangsfunktion eine Zweierpotenz ist. Zumal das genau der Grund ist, warum die Umstellung es überhaupt schafft die Laufzeit so drastisch zu reduzieren. Das was diese Formel ausmacht sind eigentlich nur die letzten beiden Faktoren des Produkts. Denn multipliziert man diese Faktoren miteinander, dann fällt auf, dass man diesen Ausdruck erhält:

$$\begin{bmatrix} F_{N/2} \cdot f_{gerade} & 0 \\ 0 & F_{N/2} \cdot f_{ungerade} \end{bmatrix}$$

Auffällig ist jetzt, dass man zwei mal das Anfangsproblem, in halber Größe erhält. Man könnte zunächst denken, dies mache keinen Unterschied, aber dadurch dass die Berechnung der komplette Matrix, einen Aufwand von  $O(N^2)$  erfordert, reduziert man mit jeder Anwendung der Formel die Laufzeit für die Berechnung von  $F_{N/2} \cdot f$  auf  $O((N/2)^2)$ . Wendet man diese Formel rekursiv, also immer wieder auf sich selbst, an, bis man Matrizen mit einer Größe von 1 oder 2 erhält, so schafft man es, durch ein zeitsparendes zusammenfügen, die Laufzeit der Berechnung auf  $O(N \log N)$  zu kürzen. Die Struktur einer FFT mit 8 Werten, kann man auch so darstellen:



In meinem Programm, zu welchem ich im nächsten Kapitel komme, mit welchem ich eine Audiospur aufnehme und verarbeite, nutze ich eine auf meinem Gerät eine Samplerate von 44800 Samples pro Sekunde. Bedeutet ich nehme pro Sekunde 44800 Datenpunkte auf. Wenn man nun die Laufzeit einer DFT und einer FFT grob skizziert, sieht man, dass die Verarbeitung eines Signals, mit nur einer Sekunde, die DFT mehr als doppelt so lang dauert, als die FFT.



## 4 Fourier-Reihe

### 4.1 Herleitung der Reihe

### 4.2 Berechnung der Koeffizienten

### 4.3 Eigenschaften und Konvergenz

## 5 Von der Reihe zur Fourier-Transformation

### 5.1 Übergang zum Integral

### 5.2 Fourier-Transformation und inverse Transformation

### 5.3 Eigenschaften

Linearität, Verschiebung, Faltung.

## 6 Diskrete Fourier-Transformation (DFT)

### 6.1 Definition und Motivation

### 6.2 Herleitung aus der Fourier-Transformation

Eigene Herleitung

### 6.3 Beweise für die Korrektheit

### 6.4 FFT als effiziente Berechnung

## 7 Eigene Beiträge

### 7.1 Eigene Herleitung der DFT

### 7.2 Beweise

### 7.3 Eigener FFT-Algorithmus in Python

### 7.4 Audio-Programm zur Echtzeit-Visualisierung

### 7.5 Bildverarbeitung: Moiré-Muster entfernen mit 2D-FFT

## 8 Anwendungen

### 8.1 Audio

Echtzeitaufnahme und Visualisierung, Tonhöhenerkennung oder Noise Cancelling, Ergebnisse.

### 8.2 Bild

Röntgenbild und Moiré-Filterung, Ergebnisse.

## 9 Anhang

### 9.1 Eigene Beweise

**Theorem 9.1.** *Die Menge  $\{\phi_n(x)\}_{n \in \mathbb{Z}}$  mit  $\phi_n : [-\pi, \pi] \rightarrow \mathbb{C}$ ,  $\phi_n(x) = \frac{1}{\sqrt{2\pi}} e^{inx}$  bildet ein Orthonormalsystem.*

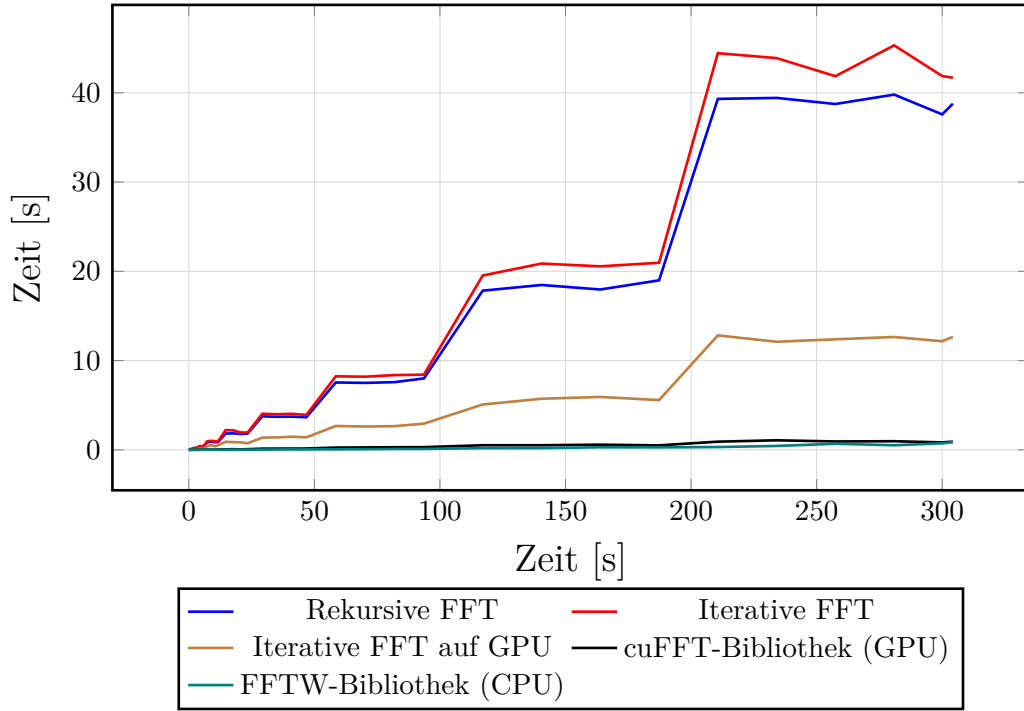


Abbildung 1: Vergleich der FFT-Methoden anhand der Benchmarks.

*Beweis.* Ich zeige die Orthonormalität in zwei Schritten:

**1. Orthogonalität:** Für  $m \neq n$  gilt

$$\langle \phi_n, \phi_m \rangle = \int_{-\pi}^{\pi} \phi_n(x) \overline{\phi_m(x)} = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{i(n-m)x} dx$$

Jetzt definiere ich  $l := n - m$

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} e^{i(n-m)x} dx = \frac{1}{2\pi} \int_{-\pi}^{\pi} \cos(lx) + i \sin(lx) dx$$

$$= \frac{1}{2\pi l} \left( \int_{-\pi}^{\pi} \cos(lx) dx + i \int_{-\pi}^{\pi} \sin(lx) dx \right) = \frac{1}{2\pi l} \left( \sin(lx) \Big|_{-\pi}^{\pi} - i \cos(lx) \Big|_{-\pi}^{\pi} \right)$$

$$= \frac{1}{2\pi l} \cdot \left( \sin(l^2\pi) - \sin(-l^2\pi) - i \left( \cos(l^2\pi) - \cos(-l^2\pi) \right) \right)$$

Da  $l \in \mathbb{Z}$  ist  $l^2 \in \mathbb{Z}$ . Ein ganzzahliges Vielfaches von  $\pi$  eingesetzt im Sinus ist zudem immer 0. Denn der Sinus ist eine  $2\pi$  periodische Funktion, und da sowohl  $\sin(0) = 0$ , als auch  $\sin(\pi) = 0$  ist, ist jedes ganzzahliges Vielfaches

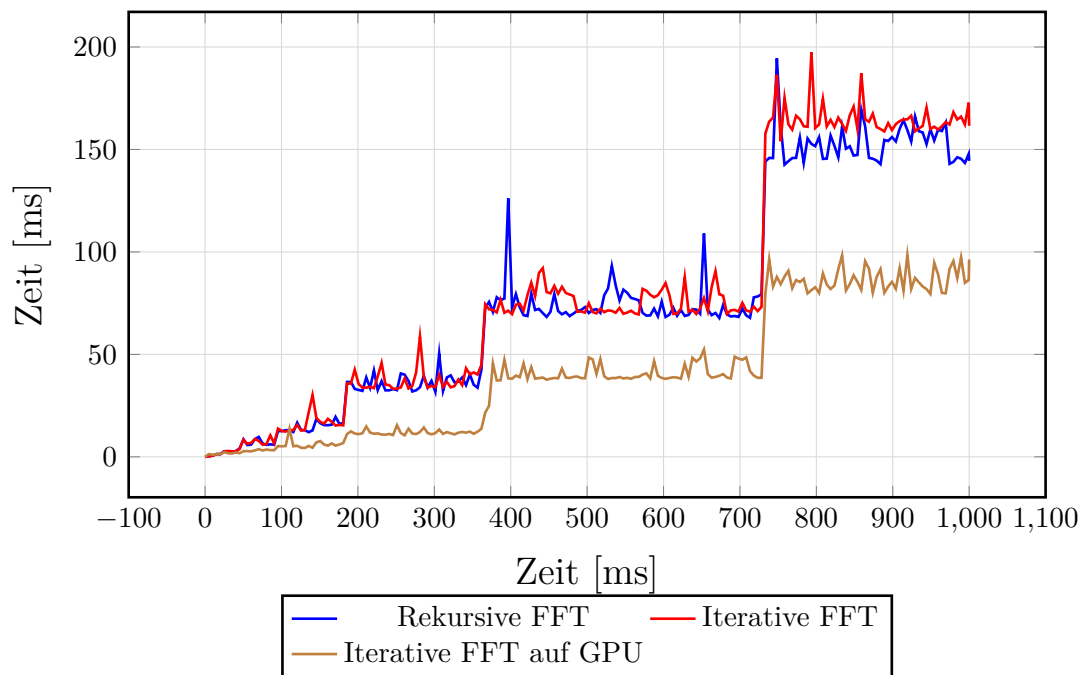


Abbildung 2: Vergleich der FFT-Methoden für die zweite Messreihe.

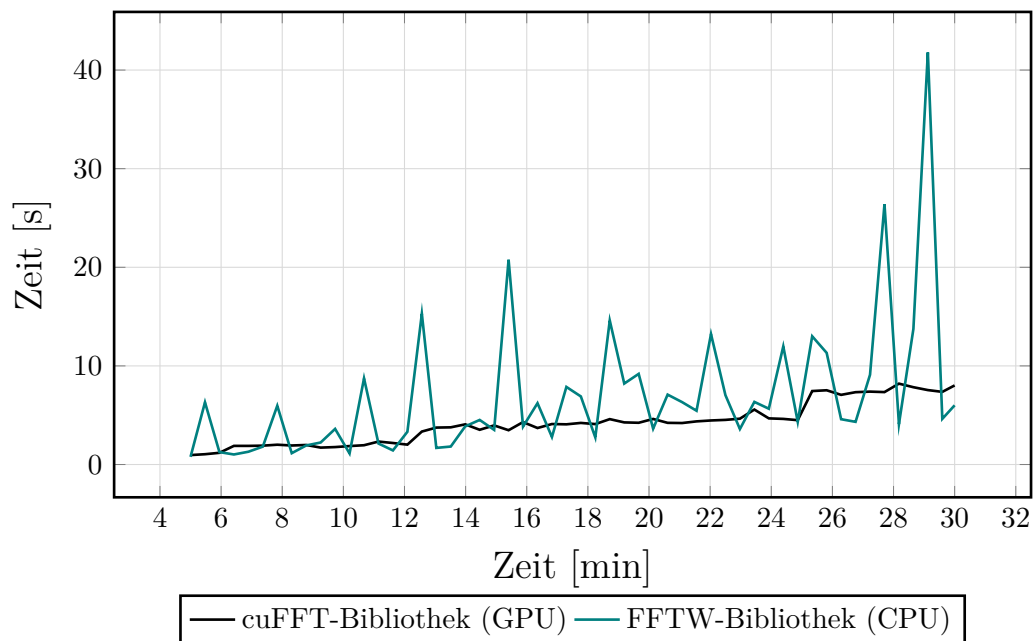


Abbildung 3: Vergleich der FFT-Methoden für die zweite Messreihe.

von  $\pi$  eingesetzt im Sinus 0. Hinzu kommt, dass der Kosinus Achsensymmetrisch bezüglich der Y-Achse ist, also gilt:  $\cos(x) = \cos(-x)$ . Daraus folgt:

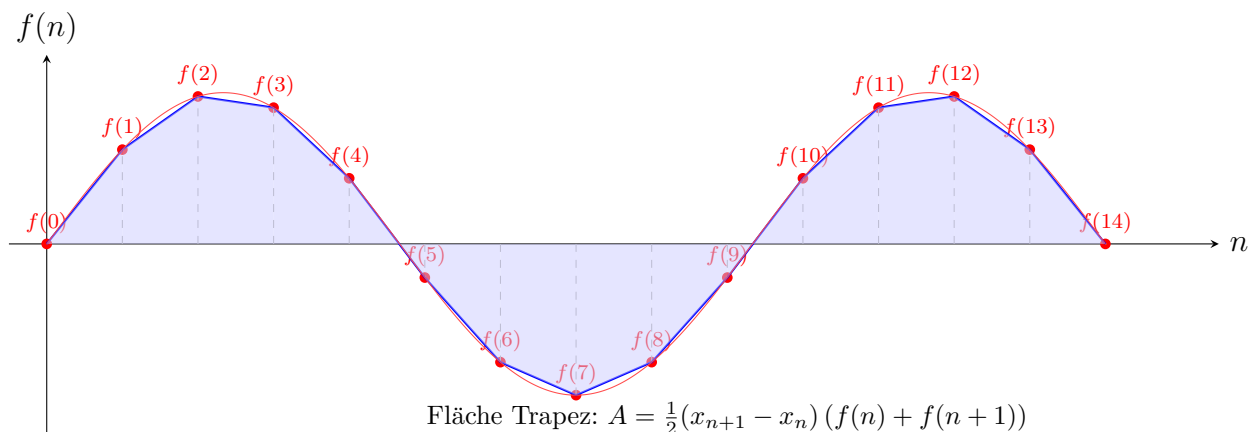
$$= \frac{1}{2\pi l} \cdot (0 - 0 - i \cdot (\cos(l\pi) - \cos(l\pi))) = \frac{1}{2\pi l} \cdot 0 = 0$$

**2. Normiertheit:** Für  $m = n$  gilt

$$\begin{aligned} \langle \phi_n, \phi_m \rangle &= \int_{-\pi}^{\pi} \phi_n(x) \overline{\phi_n(x)} dx = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{i(n-n)x} dx = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{i \cdot 0} dx \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} 1 dx = \frac{1}{2\pi} \cdot x \Big|_{-\pi}^{\pi} = \frac{1}{2\pi} \cdot (\pi - (-\pi)) = 1 \end{aligned}$$

Damit ist gezeigt, dass  $\{\phi_n(x)\}_{n \in \mathbb{Z}}$  ein Orthonormalsystem bildet.  $\square$

## 9.2 Bedeutung im größeren Kontext



## 10 Fazit und Ausblick

### 10.1 Zusammenfassung der Ergebnisse

### 10.2 Ausblick: Erweiterungen und Anwendungen