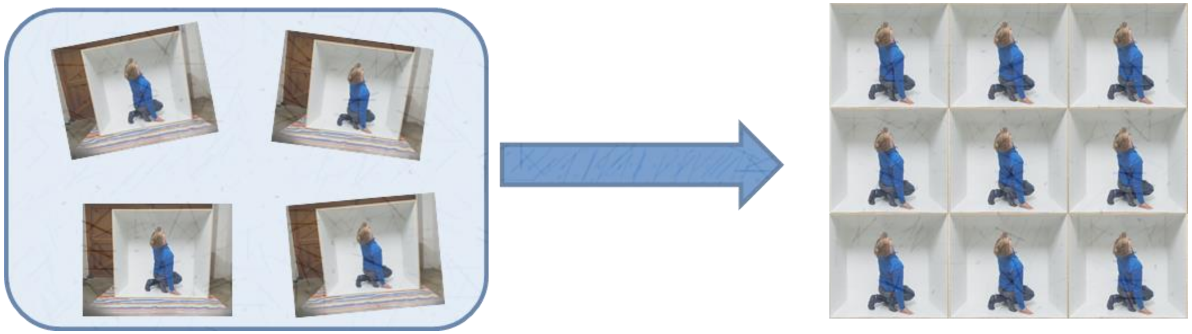
Digitale Bildverarbeitung 1

Einführung in die digitale Bilderverarbeitung für Informatikstudierende im Bachelor

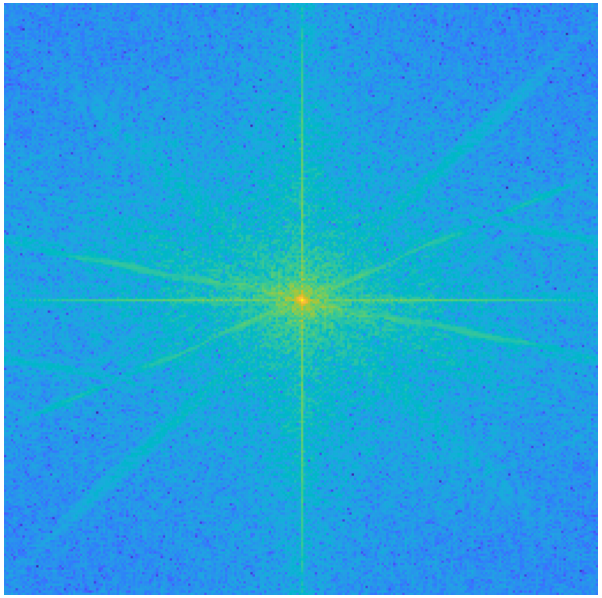
Vorlesung: Michael Möller – michael.moeller@uni-siegen.de
Übungen: Hannah Dröge – hannah.droege@uni-siegen.de



Faltungen

Beim letzten Mal motiviert: Bilder lassen sich als Überlagerung unterschiedlicher Frequenzen darstellen.

Ortsdarstellung

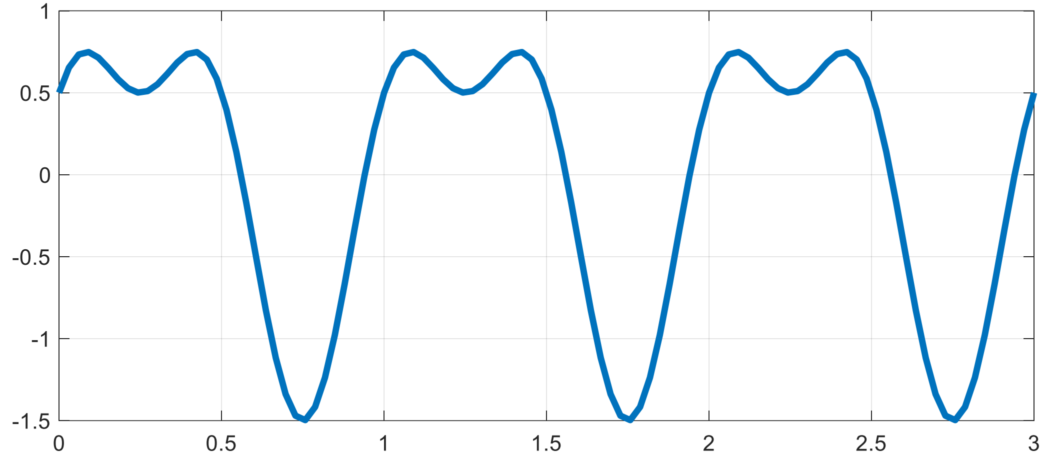


Frequenzdarstellung
(Fourier Koeffizienten)

Wir fangen erstmal einfach, nämlich mit Signalen an!

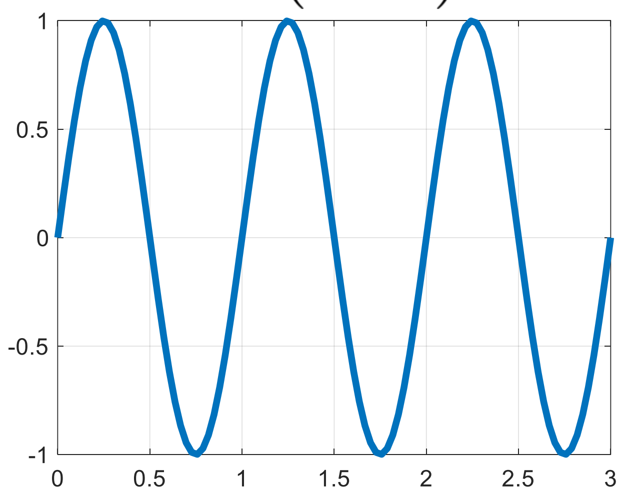
Frequenzdarstellung

Betrachten wir eine Funktion, die periodisch mit Periodenlänge 1 ist und als Überlagerung von Sinus und Consinus Funktionen geschrieben werden kann.



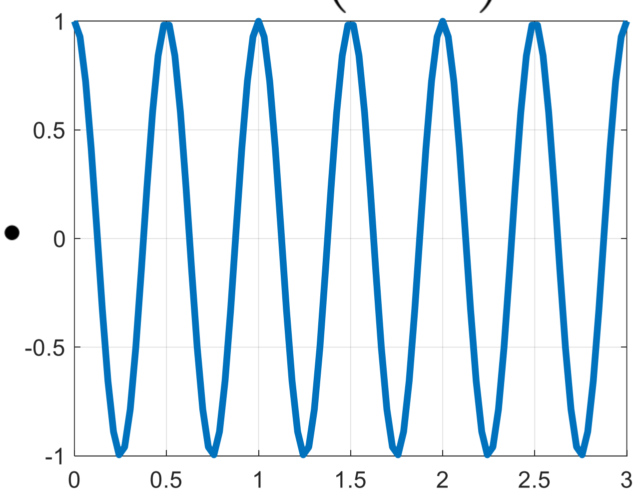
$\sin(2\pi x)$

$= 1 \cdot$

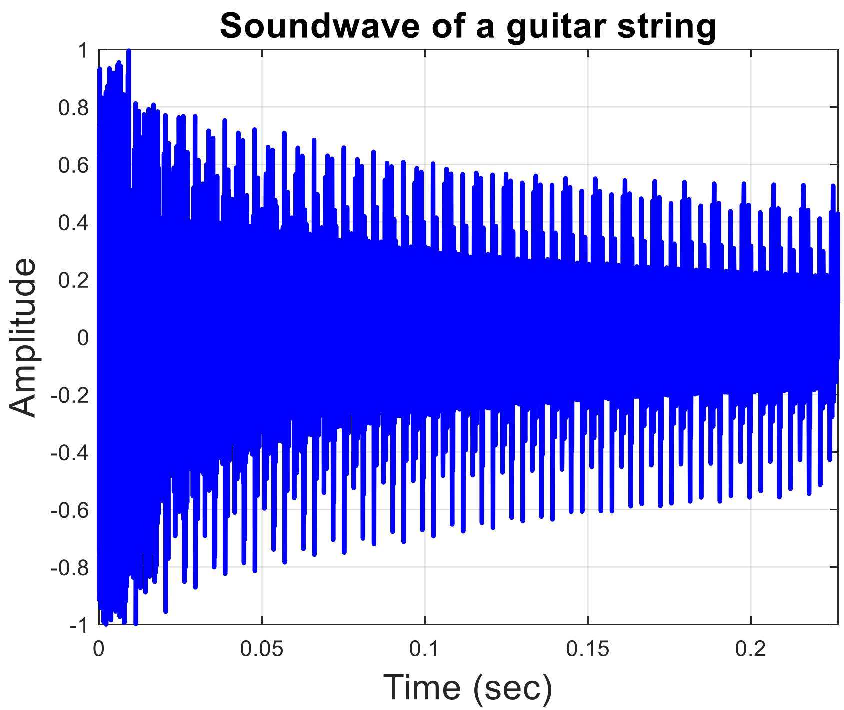


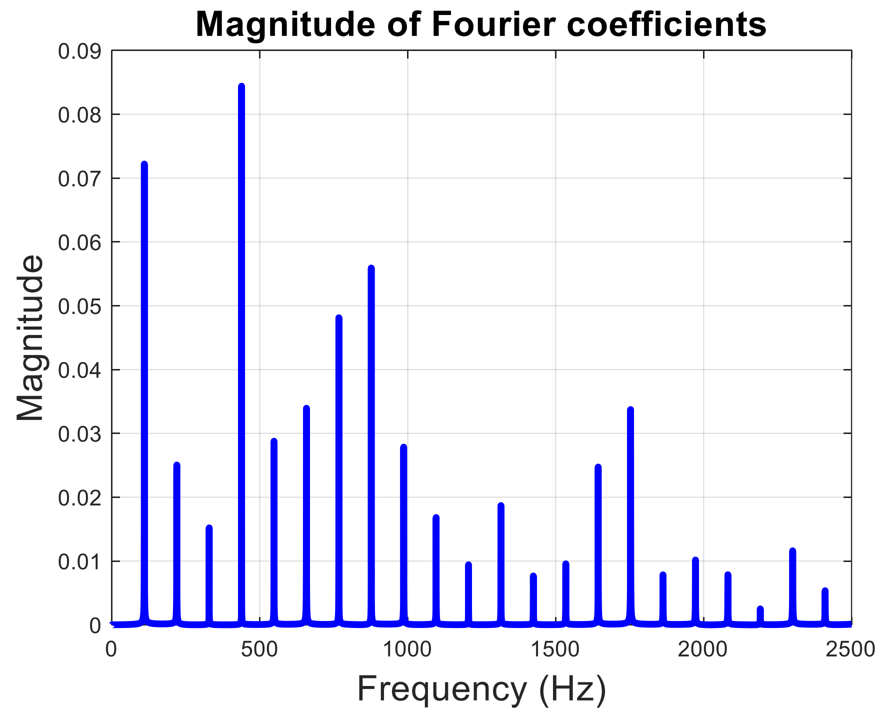
$+ 0.5 \cdot$

$\cos(4\pi x)$



Anwendung





Eine geeignete Transformation namens ***Fouriertransformation*** stellt das Signal als Überlagerung von Wellen unterschiedlicher Frequenzen dar!

https://de.wikipedia.org/wiki/Frequenzen_der_gleichstufigen_Stimmung

Tastennummer	Notation (englisch)	Notation (deutsch)	Frequenz in Hertz (Kammerton 440 Hz)
25	A2	A	110,000

Fourier Reihen

Wir stellen uns das Gitarren-Signal als Funktion vor: $f : \mathbb{R} \to \mathbb{R}$

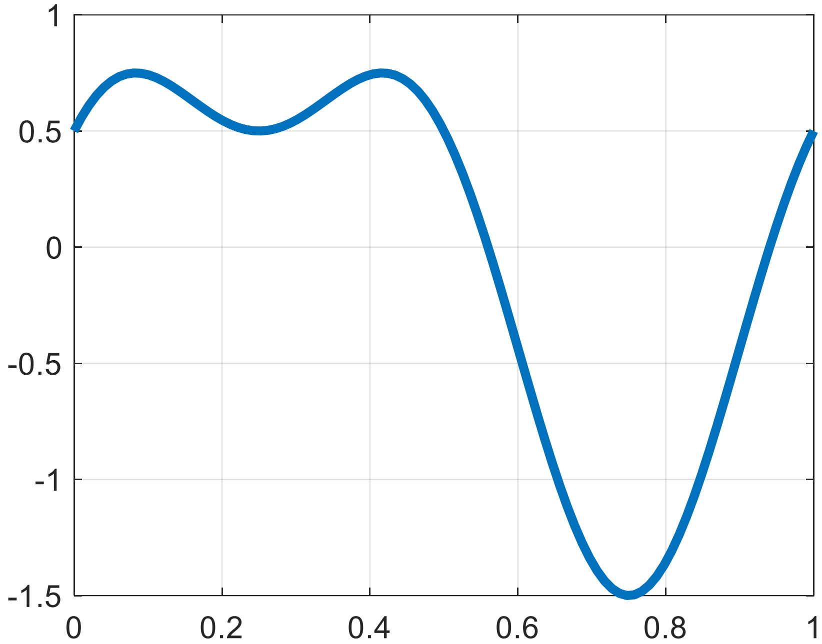
Annahme: f ist 1-periodisch und kann geschrieben werden als

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{N} a_k \cos(2\pi kx) + b_k \sin(2\pi kx)$$

Dies nennt man eine **Fourierreihe**. Die Koeffizienten a_k und b_k geben den Beitrag des Sinus bzw. Cosinus einer bestimmten Frequenz an.

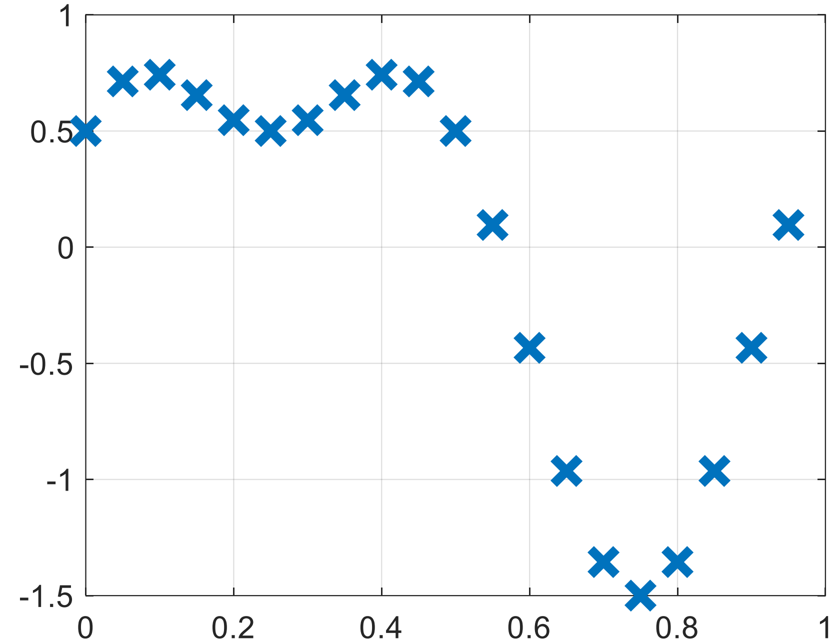
Diese Analyse wie gut man bestimmte Funktionen als Fourierreihe approximieren kann könnte man nun fortführen und so die Fouriertransformation herleiten. Hier schauen wir uns das Problem einmal nur diskret an.

Diskrete Darstellung



Wahre kontinuierliche Funktion, z.B.

$$f(x) = \sin(2\pi x) + 0.5\cos(4\pi x)$$



Im Computer typischer nur diskrete Datenpunkte, z.B. $\vec{f} \in \mathbb{R}^n$

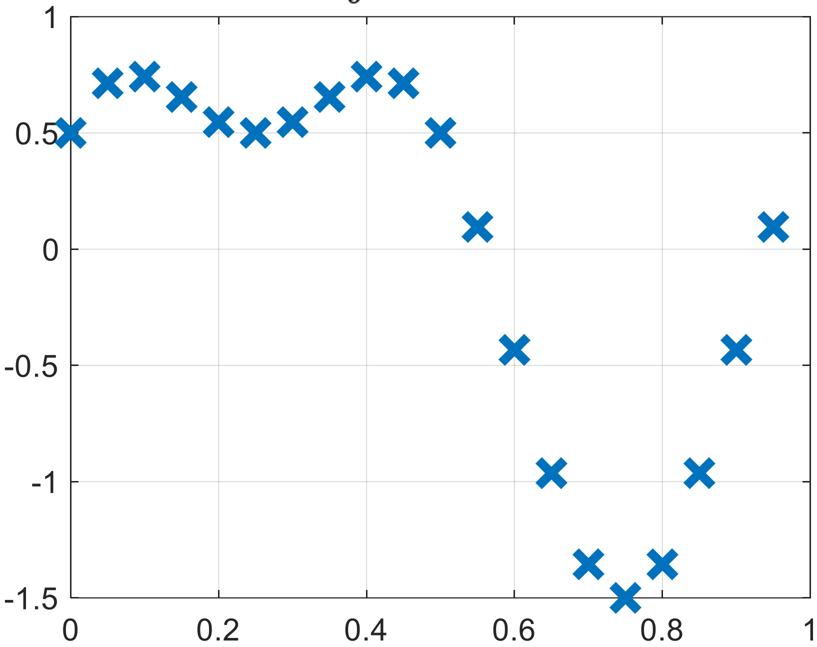
$$\vec{f} = \begin{pmatrix} f_0 \\ f_1 \\ \dots \\ f_{n-1} \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ \dots \\ f(x_{n-1}) \end{pmatrix}$$

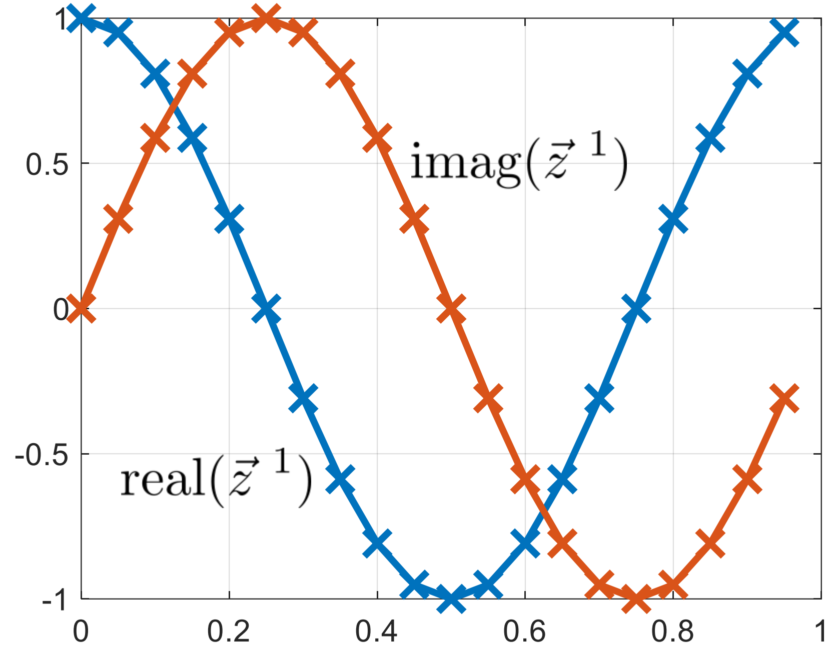
Z.B. für $x_k = k/n$

Diskrete Darstellung

Welche „Frequenzen“ sind in diesem Signal enthalten?

$$\vec{f} \in \mathbb{R}^n$$





Wir verwenden: $\vec{z}^{\,k} \in \mathbb{C}^n$

mit $z_l^k = \exp\left(2\pi \frac{lk}{n}\right)$

Denn so sind sowohl Cosinus wie auch Sinus Anteile einer bestimmten Frequenz in $\vec{z}^{\,k}$ enthalten.

$$\text{real}(z_l^k) = \cos\left(2\pi \frac{lk}{n}\right)$$

$$\text{imag}(z_l^k) = \sin\left(2\pi \frac{kl}{n}\right)$$

Darstellung in neuer Basis

Tolle Eigenschaft der $\vec{z}^{\,k} \in \mathbb{C}^n$ mit $z_l^k = \exp\left(2\pi i l \frac{k}{n}\right)$

Die $\vec{z}^{\,k}$ sind paarweise orthogonal:

$$\langle \vec{z}^{\,k}, \vec{z}^{\,s} \rangle = 0 \qquad \forall k, s \in \{0, \dots, n-1\},\ k \neq s$$

und man kann zeigen, dass:

$$\langle \vec{z}^{\,k}, \vec{z}^{\,k} \rangle = n \qquad \forall k$$

Wir können jedes $\vec{f} \in \mathbb{R}^n$, sogar jedes $\vec{f} \in \mathbb{C}^n$ als Linearkombination der $\vec{z}^{\,k}$ darstellen!

$$\vec{f} = \frac{1}{n} \sum_{k=0}^{n-1} c_k \vec{z}^{\,k}$$ mit geeigneten Koeffizienten $c_k \in \mathbb{C}$

Wie kann man die Koeffizienten c_k bestimmen?

Michael Möller – michael.moeller@uni-siegen.de

Darstellung in neuer Basis

Tafel: $c_k = \langle \vec{f}, \vec{z}^{\,k} \rangle$

Schreibt man nun das Skalarprodukt und die Definition von $\vec{z}^{\,k} \in \mathbb{C}^n$ aus erhält man

$$c_k = \sum_{l=0}^{n-1} f_l \exp\left(-2\pi i \frac{lk}{n}\right)$$

Erinnerung: Dies waren die Koeffizienten zur Darstellung

$$\vec{f} = \frac{1}{n} \sum_{k=0}^{n-1} c_k \vec{z}^{\,k}$$

Schreibt man diese wiederum komponentenweise, so erhält man

$$f_l = \frac{1}{n} \sum_{k=0}^{n-1} c_k \exp\left(2\pi i \frac{lk}{n}\right)$$

Fourier Transformation

Der Koeffizientenvektor $\vec{c} \in \mathbb{C}^n$ zur Darstellung eines Signals $\vec{f} \in \mathbb{C}^n$ als Linearkombination von $\frac{1}{n}\vec{z}^{\,k} \in \mathbb{C}^n$ mit $z_l^k = \exp\left(2\pi i \frac{lk}{n}\right)$

d.h. für eine Darstellung der Form

$$\vec{f} = \frac{1}{n}\sum_{k=0}^{n-1} c_k \vec{z}^{\,k}$$

Bzw. komponentenweise

$$f_l = \frac{1}{n}\sum_{k=0}^{n-1} c_k \exp\left(2\pi i \frac{lk}{n}\right)$$

Inverse Fouriertransformation

ist gegeben durch

$$c_k = \sum_{l=0}^{n-1} f_l \exp\left(-2\pi i \frac{lk}{n}\right)$$

Fouriertransformation

Wir nennen $\vec{c} \in \mathbb{C}^n$ die ***(diskrete) Fouriertransformierte*** von $\vec{f} \in \mathbb{C}^n$!

Zwischenzeitliche Zusammenfassung

Unser Ziel ist es herauszufinden, welche „Frequenzen“ in einem Signal enthalten sind.

Unsere Idee ist es das Signal hierfür als *Überlagerung von Sinus und Cosinus* darzustellen.

Im Computer sind Signal/Bilder/Daten immer diskret repräsentiert, also als Vektor $\vec{f} \in \mathbb{C}^n$

Wir suchen die Koeffizienten $c_k, \; k \in \{0, 1, \dots, n-1\}$ für eine Darstellung der Form

$$\vec{f} = \frac{1}{n} \sum_{k=0}^{n-1} c_k \vec{z}^{\,k}$$

wobei die $\vec{z}^{\,k} \in \mathbb{C}^n, \; k \in \{0, \dots, n-1\}$ Basisvektoren sind, die eine bestimmte Frequenz repräsentieren.

Fourier Transformation

Zwischenzeitliche Zusammenfassung

Was für Basisvektoren sind sinnvoll um einen Eindruck von Freuqenzen zu vermitteln?

Sowohl Sinus als auch Cosinusfunktionen kommen in Frage!

Interessante Option im Kontinuierlichen:

$$\zeta^k(x) = \exp(2\pi i k x)$$ denn dann ist $$\mathrm{real}(\zeta^k(x)) = \cos(2\pi k x)$$ $$\mathrm{imag}(\zeta^k(x)) = \sin(2\pi k x)$$

Erstmal: Je größer k desto größer die Frequenz.

Aber wir benötigen etwas diskretes – Vektoren $\vec{z}^{\,k} \in \mathbb{C}^n$ und nicht Funktionen ζ^k!

Idee: Wähle $\vec{z}^{\,k} \in \mathbb{C}^n$ als Auswertung von ζ^k an n vielen Stellen $x_l \in [0,1]$

$$z_l^k = \zeta^k(x_l), \quad x_l = l/n, \quad l \in \{0, \dots, n-1\}$$

Fourier Transformation

Zwischenzeitliche Zusammenfassung

Idee: Wähle $\vec{z}^{\,k} \in \mathbb{C}^n$ als Auswertung von ζ^k an n vielen Stellen $x_l \in [0,1]$

$$(z^k)_l = \zeta^k(x_l), \quad x_l = l/n, \quad l \in \{0, \dots, n-1\} \qquad\qquad \zeta^k(x) = \exp(2\pi i k x)$$

$$\Rightarrow \quad z_l^k = \exp\left(2\pi i \frac{kl}{n}\right)$$

Für diese $\vec{z}^{\,k} \in \mathbb{C}^n$ haben wir die Koeffizienten der Darstellung

$$\vec{f} = \frac{1}{n} \sum_{k=0}^{n-1} c_k \vec{z}^{\,k}$$

Inverse Fouriertransformation

bestimmt:

$$c_k = \sum_{l=0}^{n-1} f_l \exp\left(-2\pi i \frac{lk}{n}\right)$$

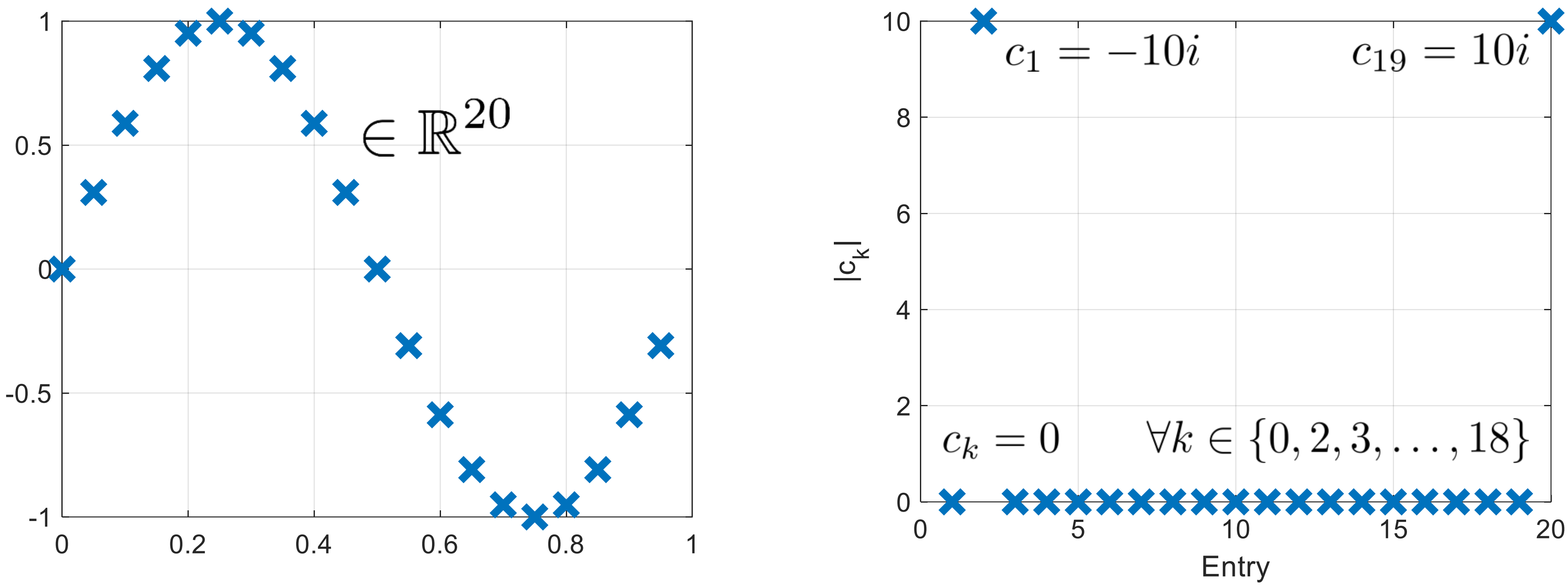
Fouriertransformation

Fourier Transformation

Eingabe: Auswertungen von

$$f(x) = \sin(2\pi x)$$

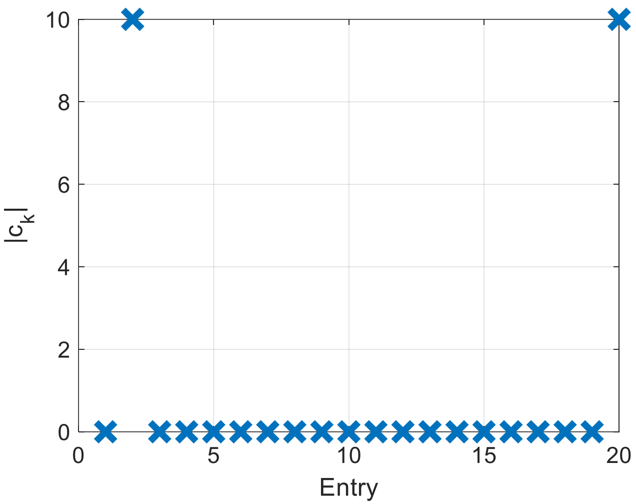
Fourier Transformierte



Michael Möller – michael.moeller@uni-siegen.de

Fourier Transformation

Wieso zwei nicht-null Einträge?



$$\exp(2\pi i x) = \cos(2\pi x) + i \sin(2\pi x)$$

$$\Rightarrow \quad i \exp(2\pi i x) = i \cos(2\pi x) - \sin(2\pi x)$$

$$\Rightarrow \quad i \exp(-2\pi i x) = i \cos(2\pi x) + \sin(2\pi x)$$

$$\Rightarrow \quad \sin(2\pi x) = \frac{1}{2} \left(i \exp(-2\pi i x) - i \exp(2\pi i x) \right)$$

Benötigte Darstellung:

$$f_l = \frac{1}{n} \sum_{k=0}^{n-1} c_k \exp\left(2\pi i \frac{lk}{n} \right)$$

Inverse Fouriertransformation

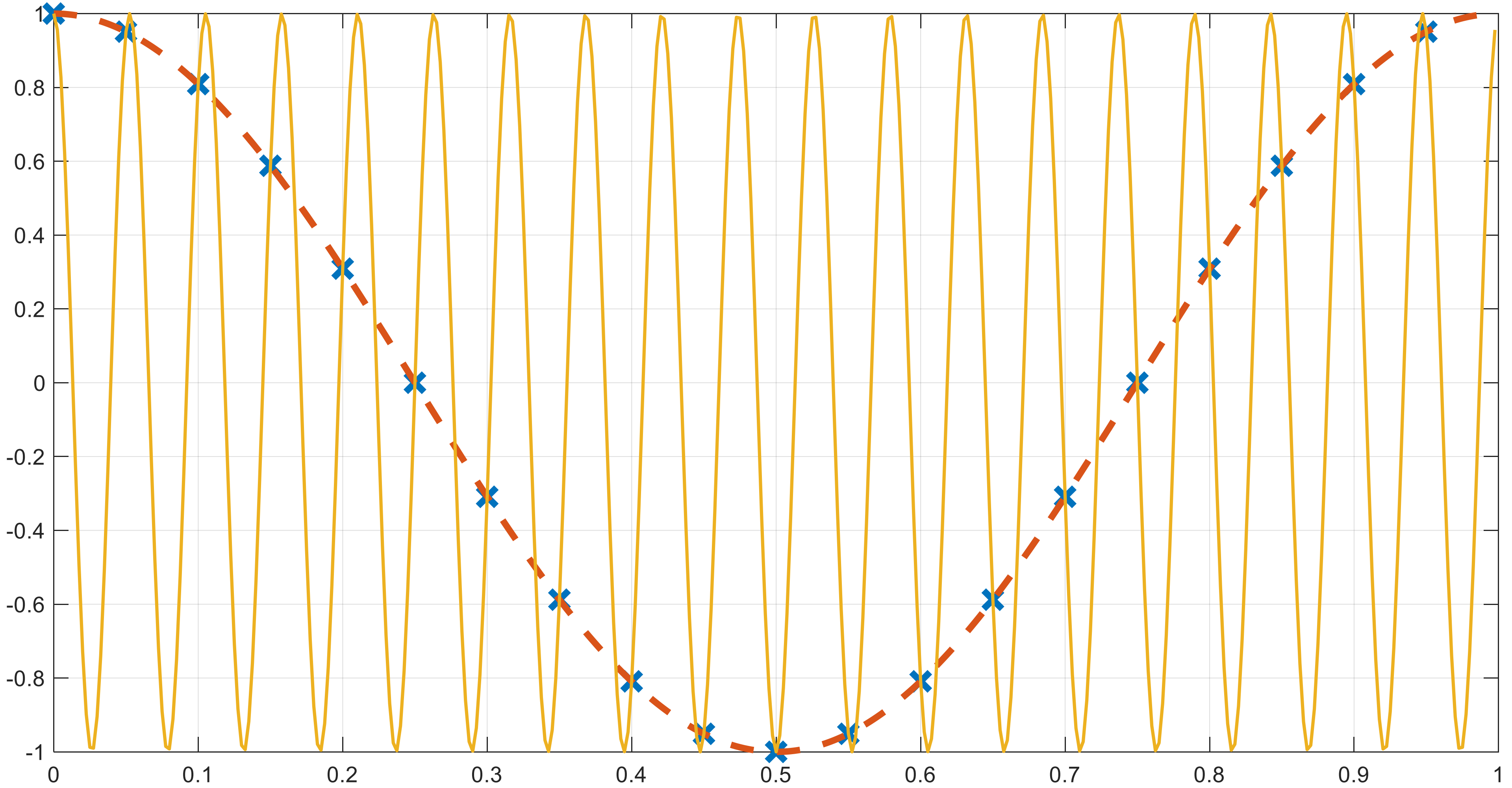
$$\sin(2\pi x) = \frac{1}{n} \left(\frac{n}{2} i \exp(-2\pi i x) - \frac{n}{2} i \exp(2\pi i x) \right)$$

Aber negative Argumente kommen in unserer Darstellungen nicht vor!

Michael Möller – michael.moeller@uni-siegen.de

Fourier Transformation

$$\exp(-2\pi i x) = \exp(2\pi i(n-1)x) \qquad \forall x \in \{0, 1/n, \ldots, (n-1)/n\}$$



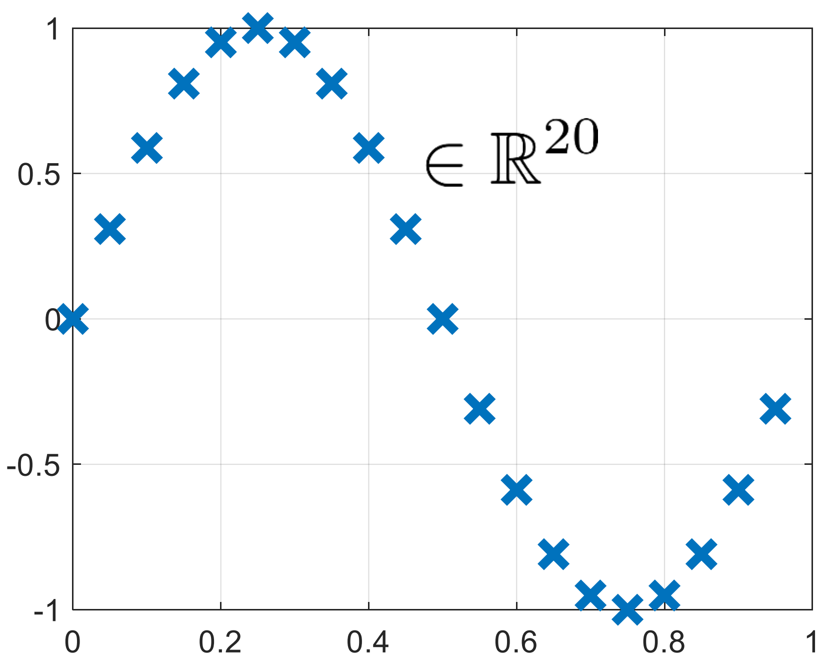
Michael Möller – michael.moeller@uni-siegen.de

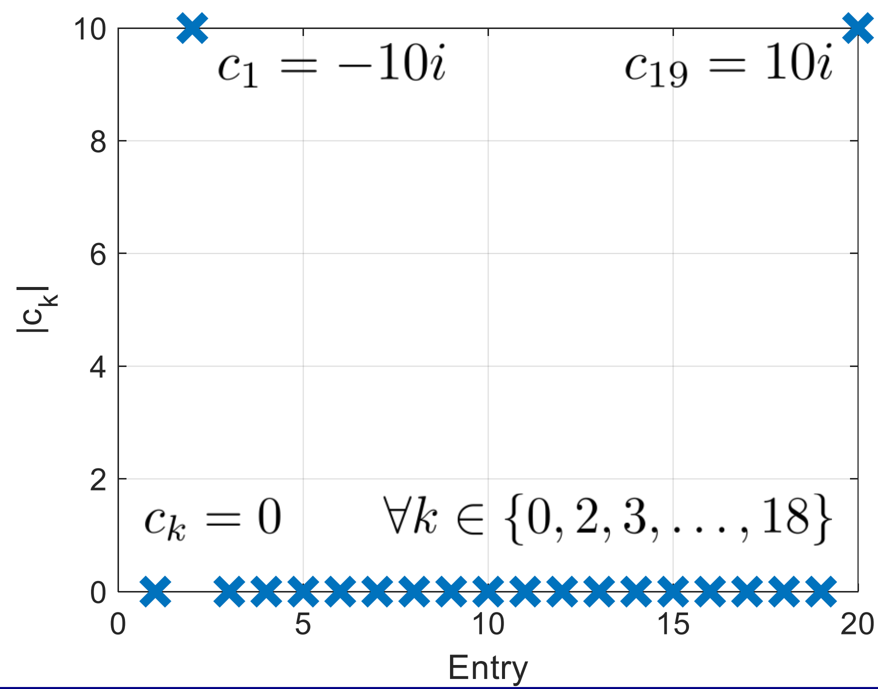
Fourier Transformation

Für $x \in \{0, 1/n, \dots, (n-1)/n\}$ gilt

$$\sin(2\pi x) = \frac{1}{n}\left(\left(\frac{n}{2}i\right)\exp(2\pi i(n-1)x) + \left(-\frac{n}{2}i\right)\exp(2\pi i x)\right)$$

c_{n-1} c_1





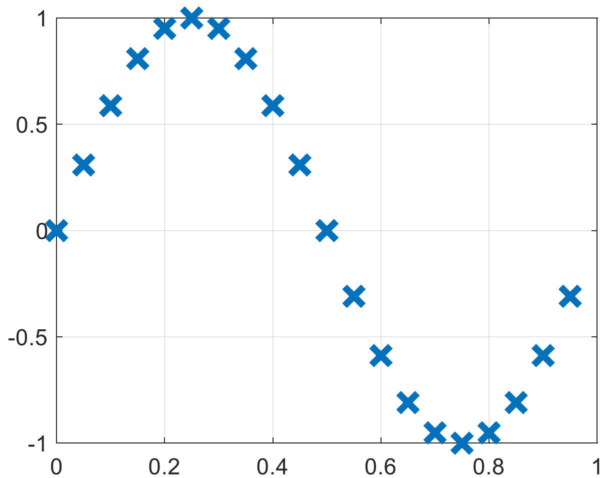
Fourier Transformation

Auch im Allgemeinen gilt

$$\exp(-2\pi i s x) = \exp(2\pi i (n-s)x) \qquad \forall x \in \{0, 1/n, \ldots, (n-1)/n\}$$

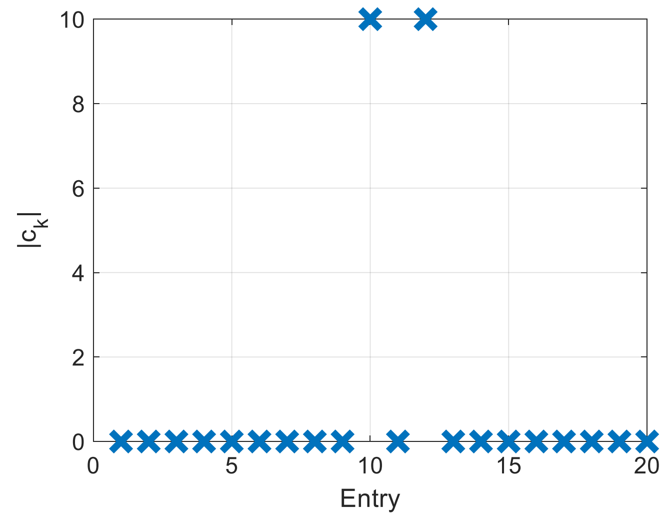
Mit anderen Worten, die Fouriertransformierte $\vec{c} \in \mathbb{C}^n$ muss zyklisch verstanden werden!

Eine typische Visualisierung ist daher, den Koeffizienten c_0 „in die Mitte" zu schieben. Dann gilt: „Je weiter am Rand desto höher die dargestellte Frequenz."



Fourier Transformation

mit Shift



Fourier Transformation

Des Weiteren kann man zeigen, dass

$$f \in \mathbb{R}^n \qquad \Rightarrow \qquad c_k = -c_{n-k} \qquad \forall 1 \leq k \leq n/2$$

(der Imaginärteil muss sich herausheben)

Um recheneffizient zu sein und mit weniger Koeffizienten rechnen zu müssen, gibt es in numpy extra eine Funktion **np.fft.rfft** neben der üblichen **np.fft.fft**

\>> Einige Beispiele zeigen!

Michael Möller – michael.moeller@uni-siegen.de

Fourier Transformation – 2d

Und wie steht es mit Bildern?

$f \in \mathbb{R}^{n_y \times n_x}$

$$f = \begin{pmatrix} f_{0,0} & f_{0,1} & \cdots & f_{0,n_x-2} & f_{0,n_x-1} \\ f_{1,0} & f_{1,1} & \cdots & f_{1,n_x-2} & f_{1,n_x-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ f_{n_y-2,0} & f_{n_y-2,1} & \cdots & f_{n_y-2,n_x-2} & f_{n_y-2,n_x-1} \\ f_{n_y-1,0} & f_{n_y-1,1} & \cdots & f_{n_y-1,n_x-2} & f_{n_y-1,n_x-1} \end{pmatrix}$$

Wenn man den 1d Fall verstanden hat, ist die 2d Fouriertransformation geradlinig:

Wir transformieren erst die Spalten von f und erhalten ein „Zwischenbild“ $\tilde{f}$.

Dann transformieren wir die Zeilen des Zwischenbildes $\tilde{f}$ und erhalten die vollständige Fouriertransformierte $c \in \mathbb{C}^{n_y \times n_x}$.

Fourier Transformation – 2d

Erinnerung aus dem 1d Fall:

$$c_k = \sum_{l=0}^{n-1} f_l \exp\left(-2\pi i \frac{lk}{n}\right)$$ **Fouriertransformation**

In Formeln:

$$\tilde{f}_{k_1,l_2} = \sum_{l_1=0}^{n_y-1} f_{l_1,l_2} \exp\left(-2\pi i \frac{l_1 k_1}{n_y}\right)$$

$$c_{k_1,k_2} = \sum_{l_2=0}^{n_x-1} \tilde{f}_{k_1,l_2} \exp\left(-2\pi i \frac{l_2 k_2}{n_x}\right)$$

$$= \sum_{l_2=0}^{n_x-1} \sum_{l_1=0}^{n_y-1} f_{l_1,l_2} \exp\left(-2\pi i \frac{l_1 k_1}{n_y}\right) \exp\left(-2\pi i \frac{l_2 k_2}{n_x}\right)$$

Wie man sieht spielt die Reihenfolge, „erst Spalten dann Zeilen“ oder umgekehrt, keine Rolle: Die Summen kommutieren!

Michael Möller – michael.moeller@uni-siegen.de

Fourier Transformation – 2d

Fouriertransformation in 2d

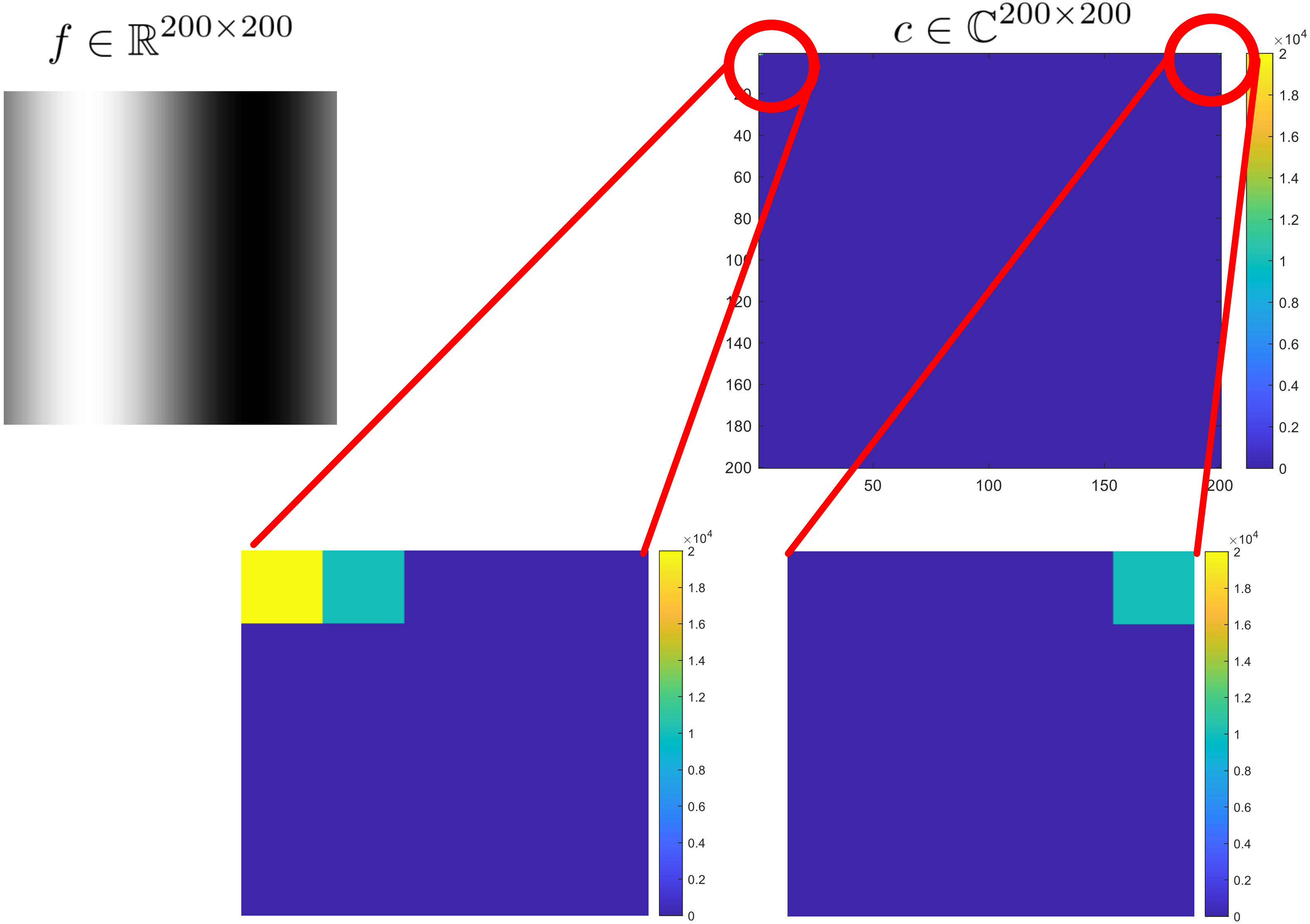
$$c_{k_1,k_2} = \sum_{l_2=0}^{n_x-1} \sum_{l_1=0}^{n_y-1} f_{l_1,l_2} \exp\left(-2\pi i \frac{l_1 k_1}{n_y}\right) \exp\left(-2\pi i \frac{l_2 k_2}{n_x}\right)$$

Inverse Fouriertransformation in 2d

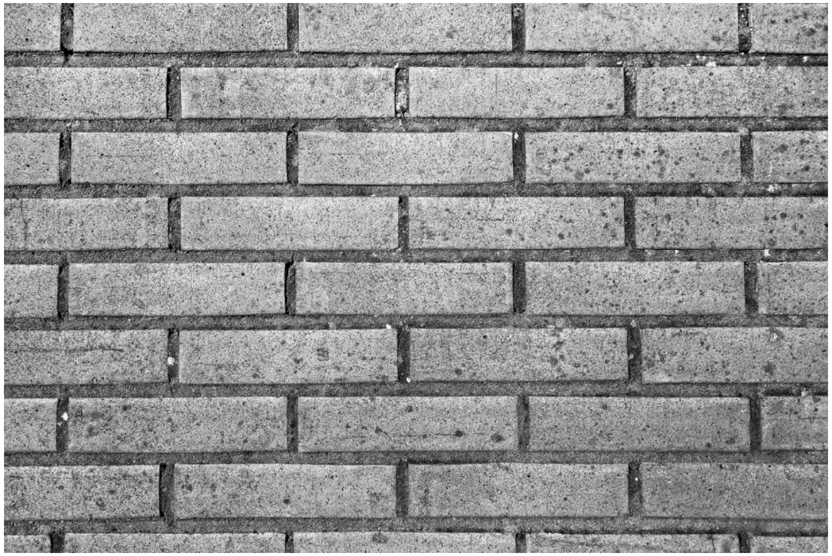
$$f_{l_1,l_2} = \frac{1}{n_y n_x} \sum_{k_1=0}^{n_y-1} \sum_{k_2=0}^{n_x-1} c_{k_1,k_2} \exp\left(2\pi i \frac{l_1 k_1}{n_y}\right) \exp\left(2\pi i \frac{l_2 k_2}{n_x}\right)$$

Fourier Transformation – 2d

$f \in \mathbb{R}^{200\times 200}$ $c \in \mathbb{C}^{200\times 200}$

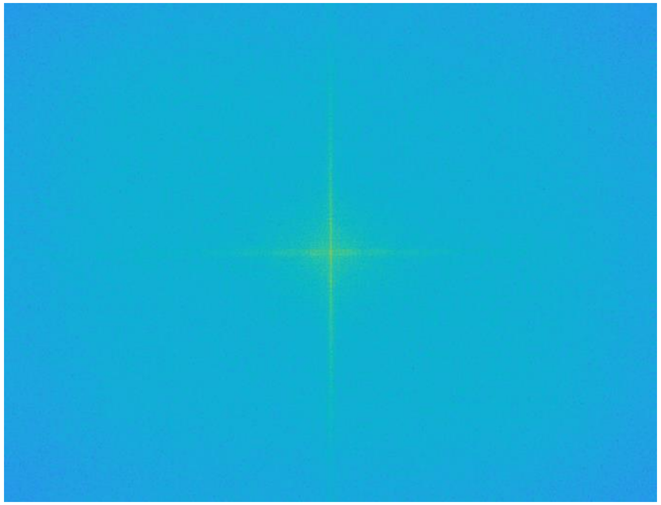


Fourier Transformation – 2d



log(|Fourier Transform|)

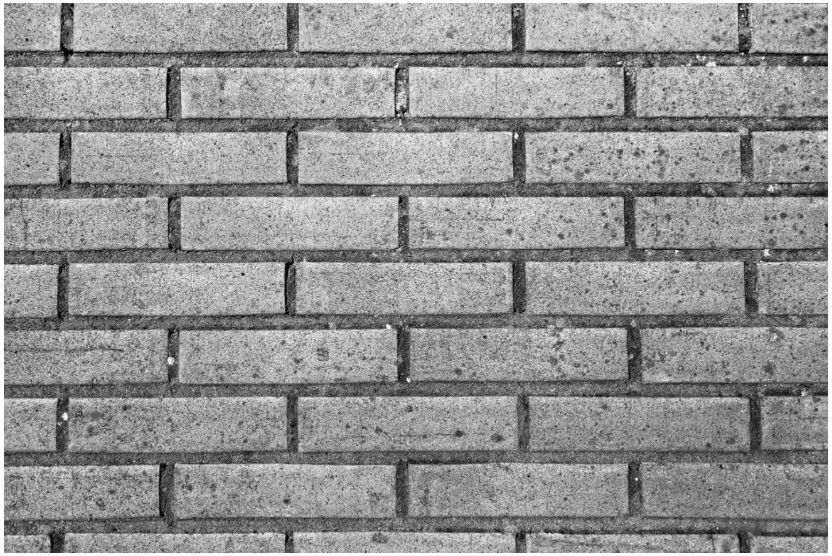
sift(log(|Fourier Transform|))



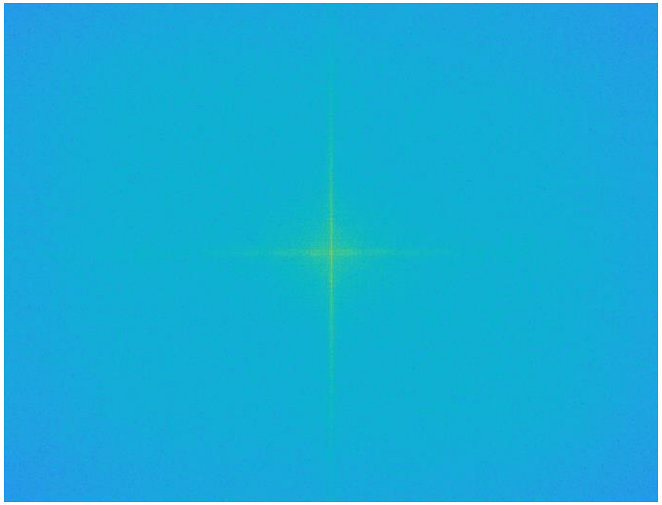
Übliche Art mit zyklischer Periodizität umzugehen: Schiebe $c_{0,0}$ in die Mitte!

Michael Möller – michael.moeller@uni-siegen.de

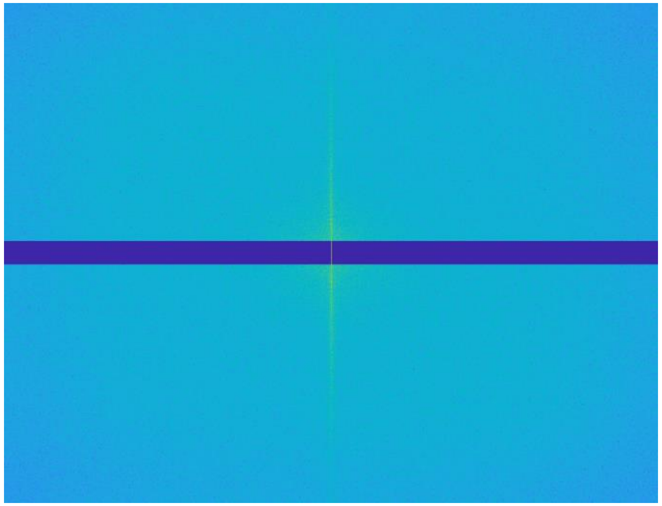
Fourier Transformation – 2d

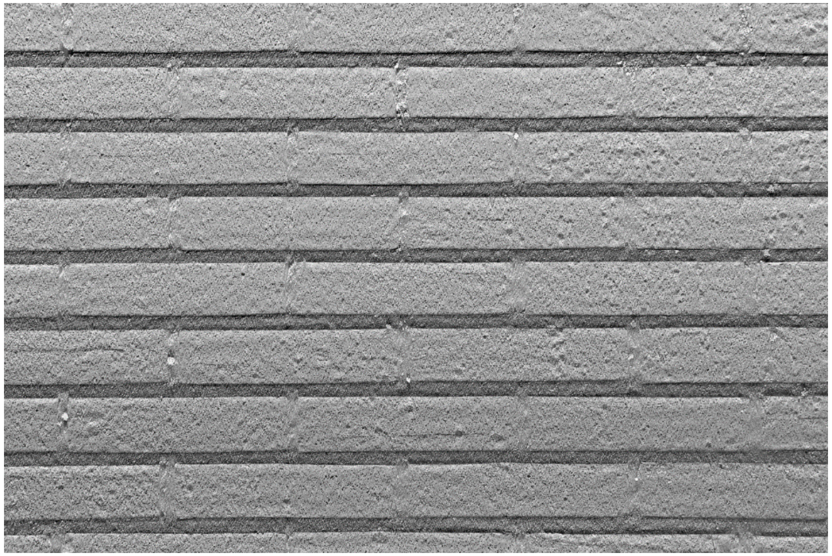


sift(log(|Fourier Transform|))

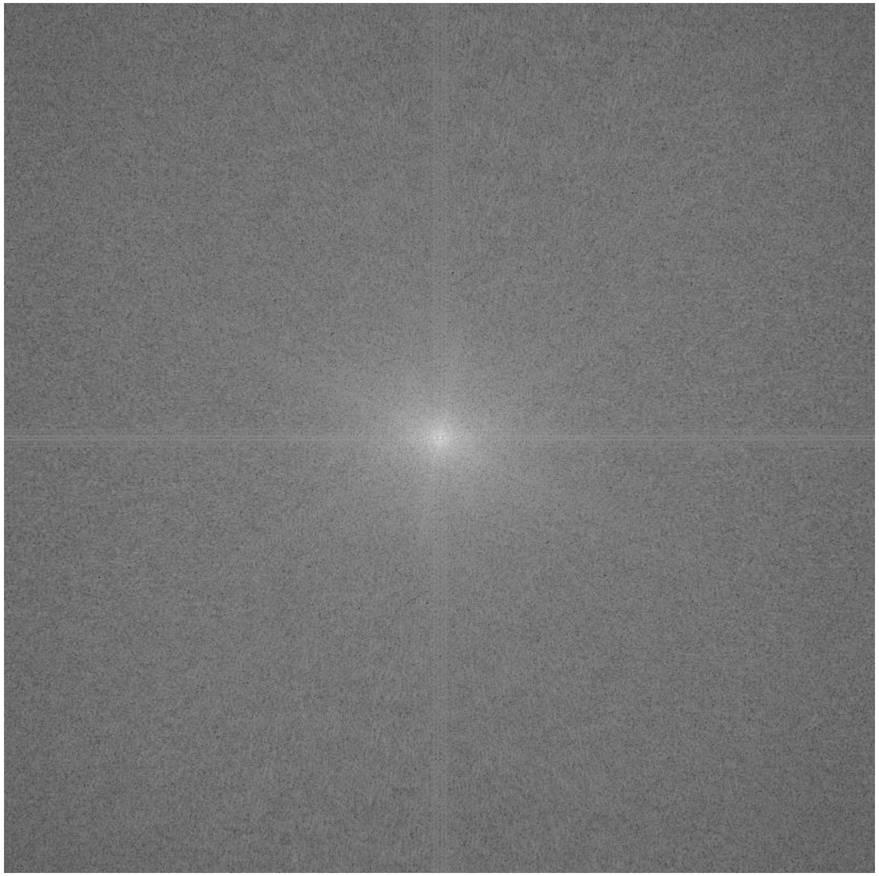


sift(log(|Fourier Transform|))

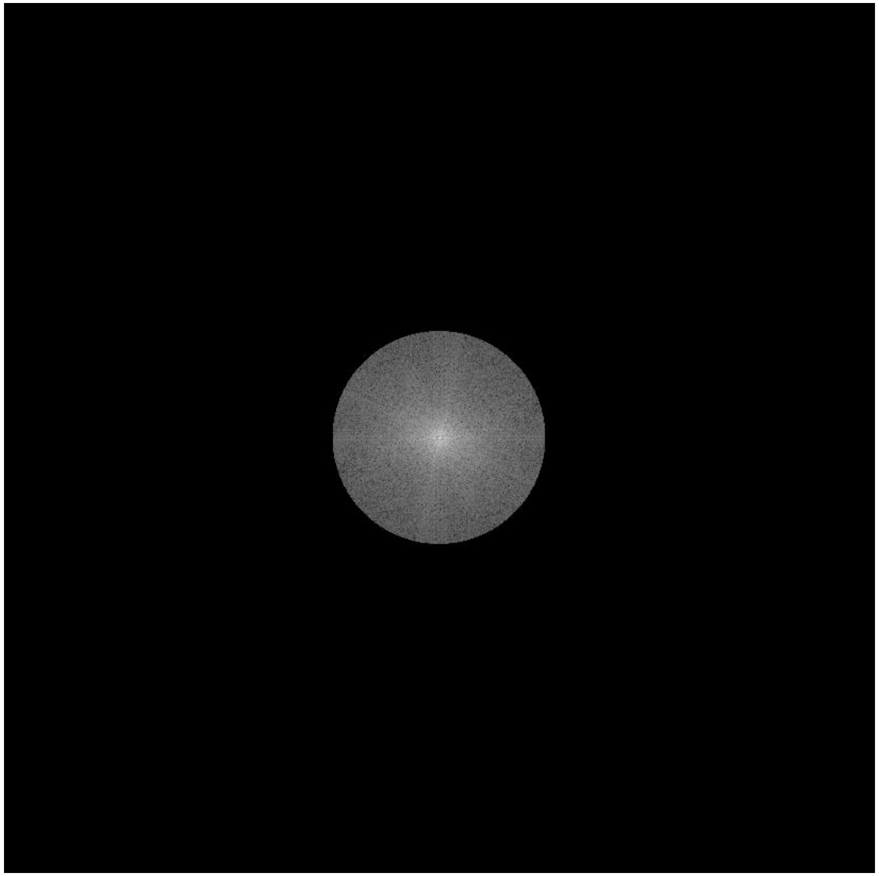




Fourier Transformation – 2d

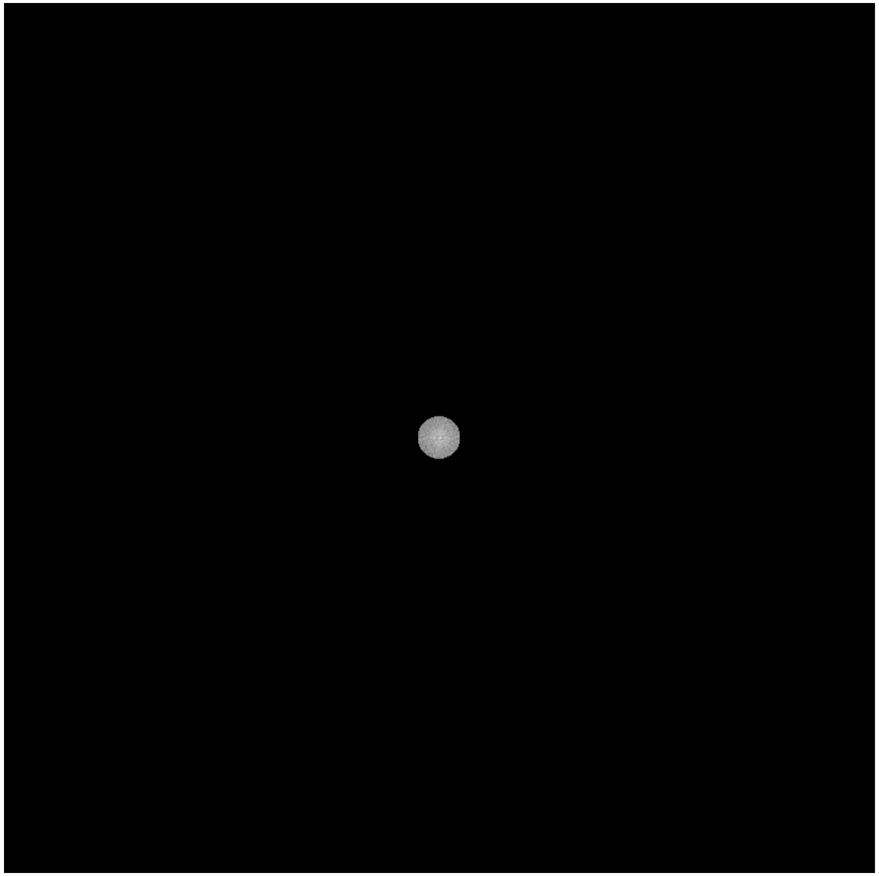


Fourier Transformation – 2d

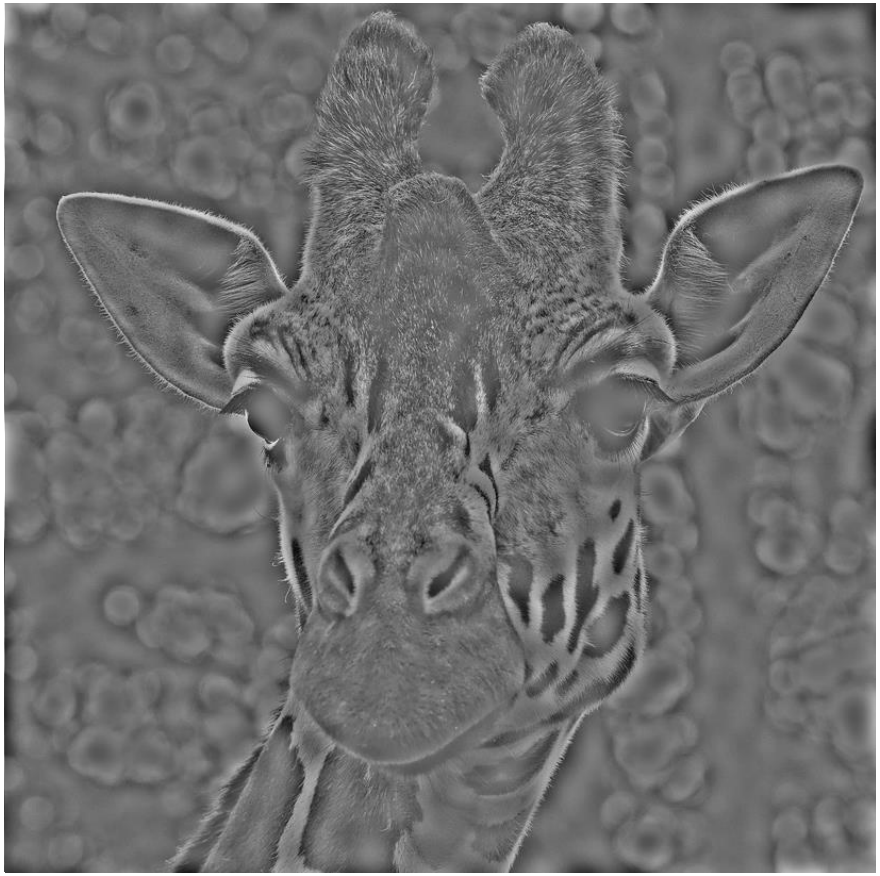


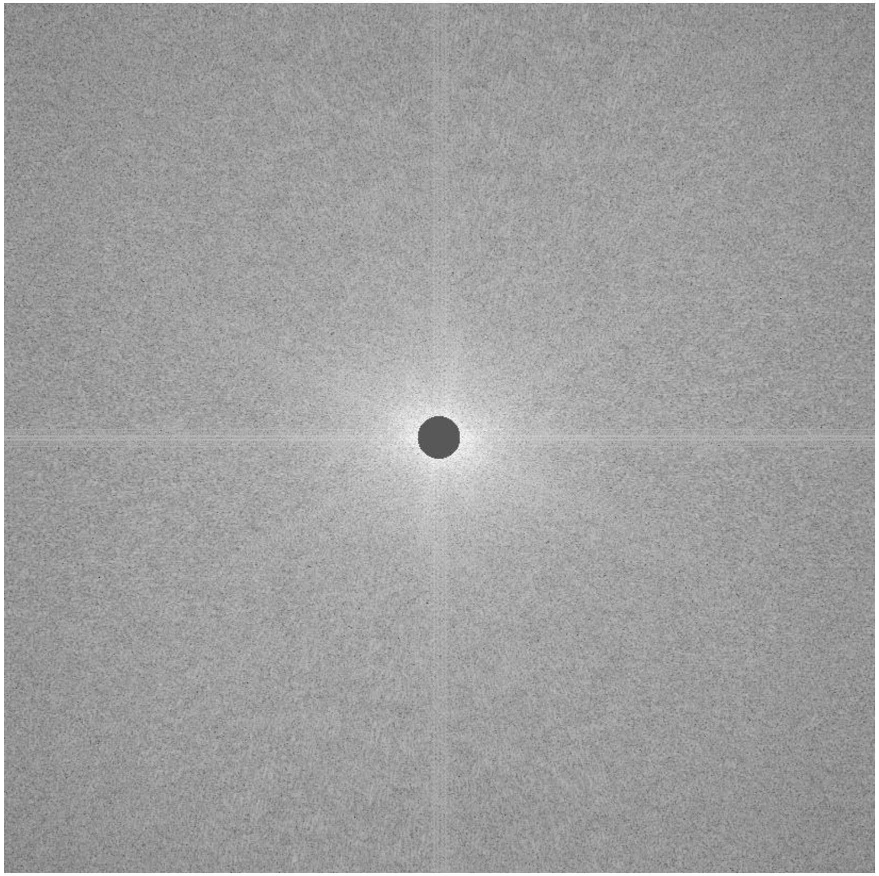
Michael Möller – michael.moeller@uni-siegen.de

Fourier Transformation – 2d

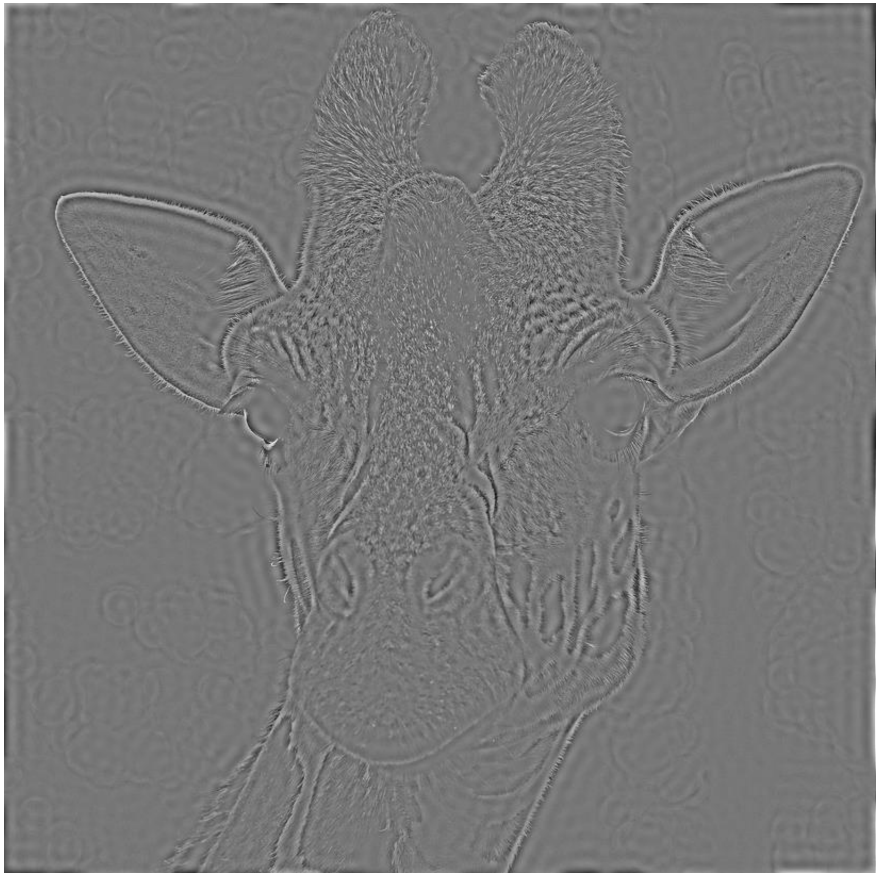


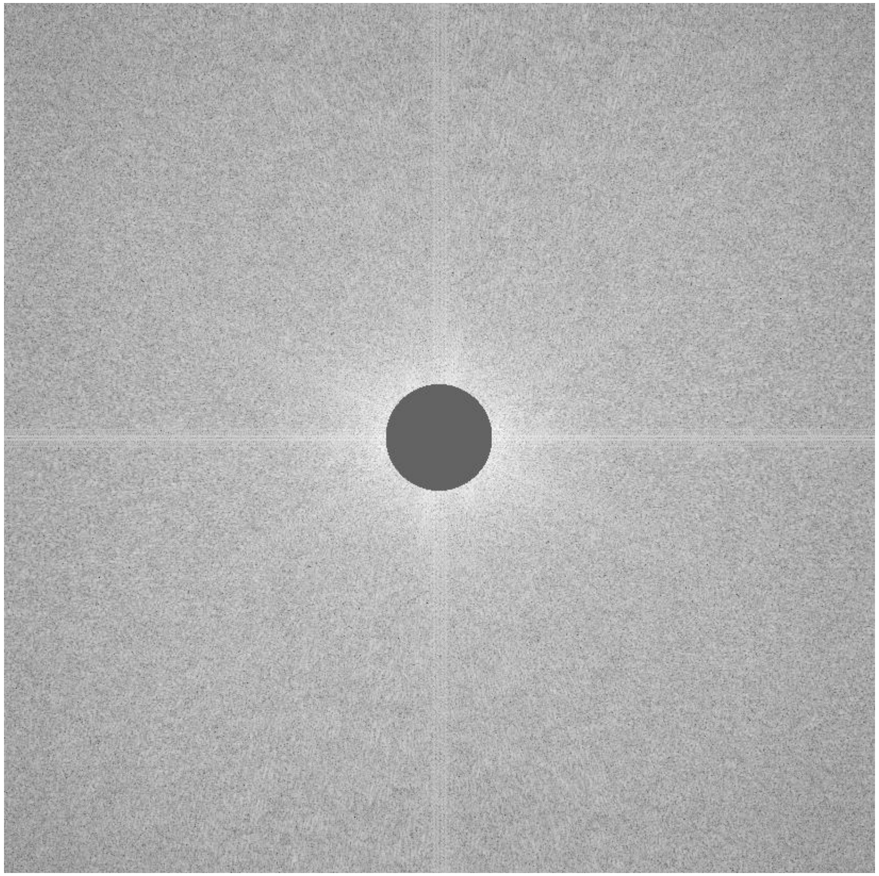
Fourier Transformation – 2d





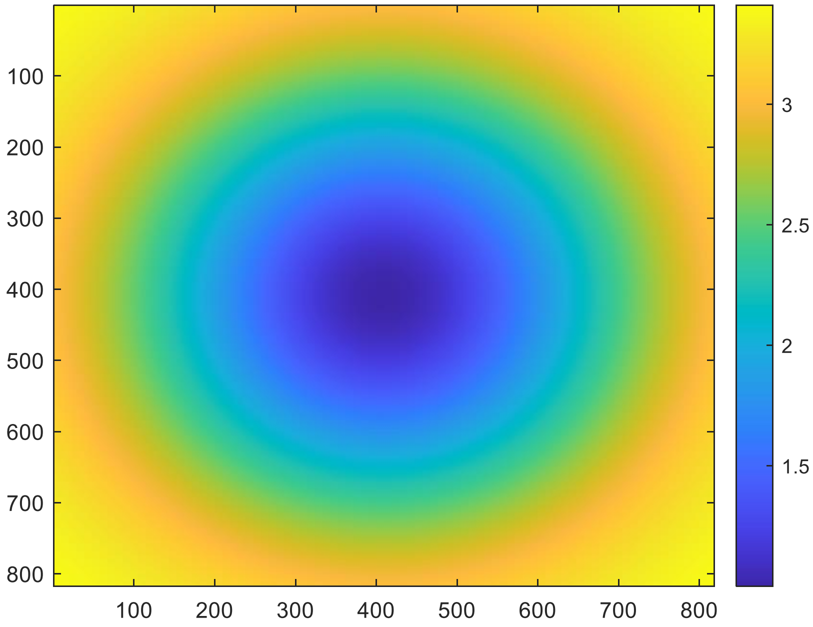
Fourier Transformation – 2d





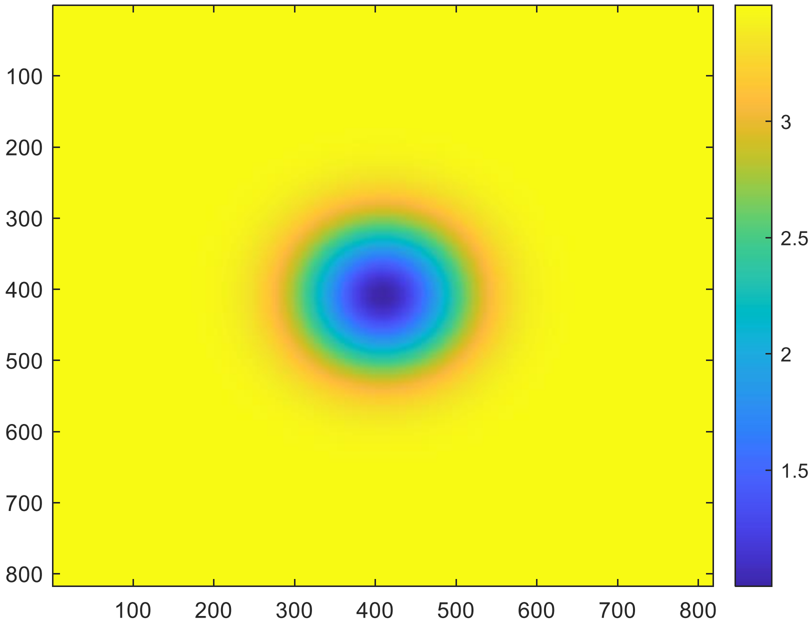
Fourier Transformation – 2d

Multipliziere FT punktweise mit



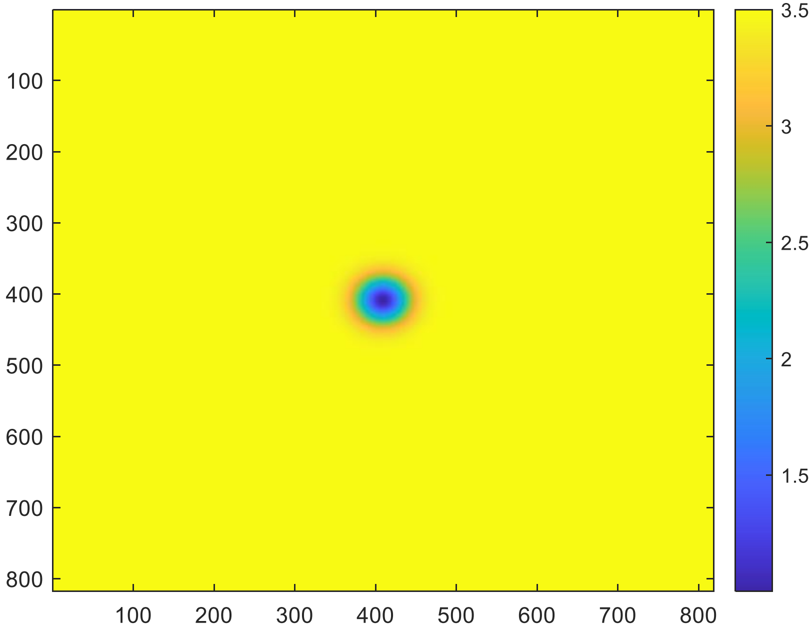
Fourier Transformation – 2d

Multipliziere FT punktweise mit



Fourier Transformation – 2d

Multipliziere FT punktweise mit



Faltungssatz

Nun hat die ganz Motivation damit angefangen, dass ich behauptet habe wir könnten Faltungen anders interpretieren. Was hat das nun mit Fourier-Transformationen zu tun?

Faltungssatz

Seien $f,\ g \in \mathbb{C}^{n_y \times n_x}$ und $\mathcal{F}(f),\ \mathcal{F}(g) \in \mathbb{C}^{n_y \times n_x}$ ihre Fouriertransformierten.

Die Faltung $f * g$ mit zyklischen Randbedingungen entspricht dem punktweisen Produkt im Fourierraum – mit anderen Worten, es gilt

$$f * g = \mathcal{F}^{-1}\left(\mathcal{F}(f) \odot \mathcal{F}(g)\right)$$

punktweises Produkt

Ausrechnen der Fourier-Trafo

Neben der schönen Interpretation – ergeben sich noch andere Vorteile aus dem Faltungssatz?

Antwort: Der Faltungssatz kann auch eine effiziente Möglichkeit sein Faltungen auszurechnen! Die Fouriertransformation lässt sich mithilfe eines algorithmischen Tricks extrem schnell implementieren!

Die sogenannte *schnelle Fourier-Transformation* (*FFT* – Fast Fourier Transform) rechnet die Fouriertransformation eines Signals der Länge n mit $\mathcal{O}(n\log(n))$ Rechenoperation aus!

Aus Wikipedia:

> *In 1994, Gilbert Strang described the FFT as "the most important numerical algorithm of our lifetime",[3][4] and it was included in Top 10 Algorithms of 20th Century by the IEEE magazine Computing in Science & Engineering.*

https://en.wikipedia.org/wiki/Fast_Fourier_transform