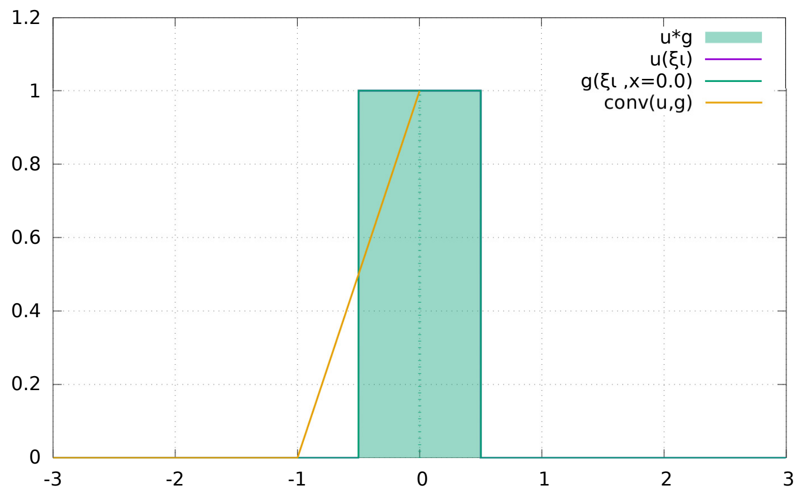
Lösung 9
Fouriertransformation

Aufgabe 1: Filterung eines Signals

(a) Es muss eine Dreiecksfunktion sein: (Hier nur zur Hälfte gezeichnet.)

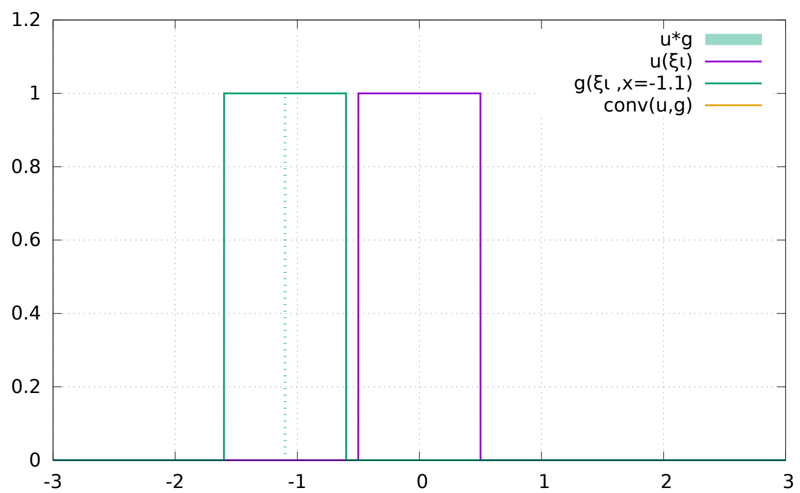


Veranschaulichung mit Animation aus Vorlesung!

(b)

$$\tilde{u} = \int_{-\infty}^{\infty} u(\xi) g(x-\xi) d\xi \qquad (1)$$

1. Fall: $x < -1$:

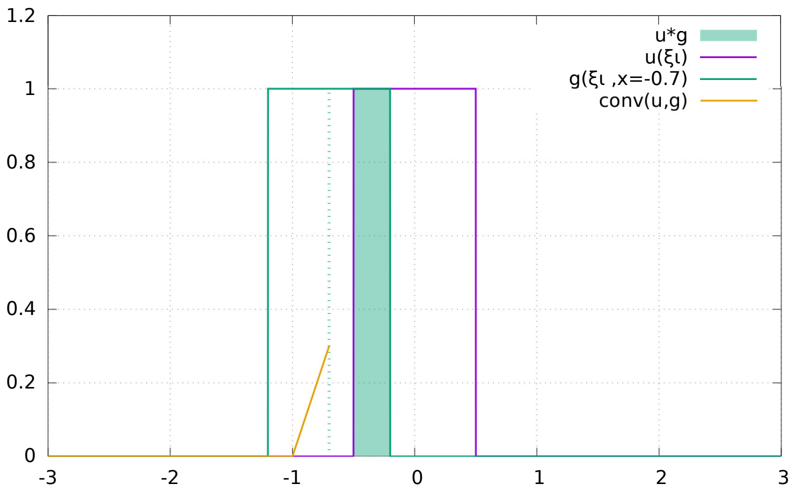


$$\tilde{u} = 0 \tag{2}$$

2. Fall: $x > 1$:

$$\tilde{u} = 0 \tag{3}$$

3. Fall: $-1 < x < 0$:



$$\tilde{u}(x) = \int_{-\frac{1}{2}}^{x+\frac{1}{2}} u(\xi) g(x-\xi) \mathrm{d}\xi = \tag{4}$$

$$= \int_{-\frac{1}{2}}^{x+\frac{1}{2}} 1 \cdot 1 \mathrm{d}\xi \tag{5}$$

$$= [\xi]_{-\frac{1}{2}}^{x+\frac{1}{2}} = x + 1 \tag{6}$$

$$\tag{7}$$

4. Fall: $0 < x < 1$:

$$\tilde{u}(x) = \int_{x-\frac{1}{2}}^{\frac{1}{2}} u(\xi) g(x-\xi) \mathrm{d}\xi \tag{8}$$

$$= \int_{x-\frac{1}{2}}^{\frac{1}{2}} 1 \cdot 1 \mathrm{d}\xi \tag{9}$$

$$= [\xi]_{x-\frac{1}{2}}^{\frac{1}{2}} = -x + 1 \tag{10}$$

$$\tag{11}$$

(c)

$$\hat{u}(\kappa) = \int_{-\infty}^{\infty} u(x)\exp(-2\pi i\kappa x)\mathrm{d}x \tag{12}$$
$$= \int_{-\frac{1}{2}}^{\frac{1}{2}} 1\exp(-2\pi i\kappa x)\mathrm{d}x \tag{13}$$
$$= \left[-\frac{1}{2\pi i\kappa}\exp(-2\pi i\kappa x)\right]_{-\frac{1}{2}}^{\frac{1}{2}} \tag{14}$$
$$= -\frac{1}{2\pi i\kappa}\left[\exp(-\pi i\kappa) - \exp(\pi i\kappa)\right] \tag{15}$$
$$= \frac{1}{\pi\kappa}\frac{1}{2i}\left[-\exp(-\pi i\kappa) + \exp(\pi i\kappa)\right] \tag{16}$$
$$= \frac{1}{\pi\kappa} sin(\pi\kappa) = sinc(\pi\kappa) \tag{17}$$

(d)

$$\hat{\tilde{u}}(\kappa) = \hat{u}(\kappa)\cdot\hat{g}(\kappa) = \mathrm{sinc}^2(\pi\kappa) \tag{18}$$

(e)

$$\tilde{u}(x) = \mathscr{F}^{-1}\left\{\hat{\tilde{u}}(\kappa)\right\} \tag{19}$$
$$= \mathscr{F}^{-1}\left\{\mathrm{sinc}^2(\pi\kappa)\right\} \tag{20}$$
$$= 1 - |x| \tag{21}$$

- Die selbe Lösung wie in Aufgabeteil 2!

Aufgabe 2: Anwendungsbeispiel einer DFT

Python Code durchgehen.

Gehen Sie den bereitgestellten Python-Code durch:

dft.py def getDFTSqMatr(size, inverse=False):
Berechnet die Transformationsmatrix entsprechend der Vorschrift:

$$A_{nm} = \frac{1}{N}e^{\frac{-2\pi inm}{N}}$$

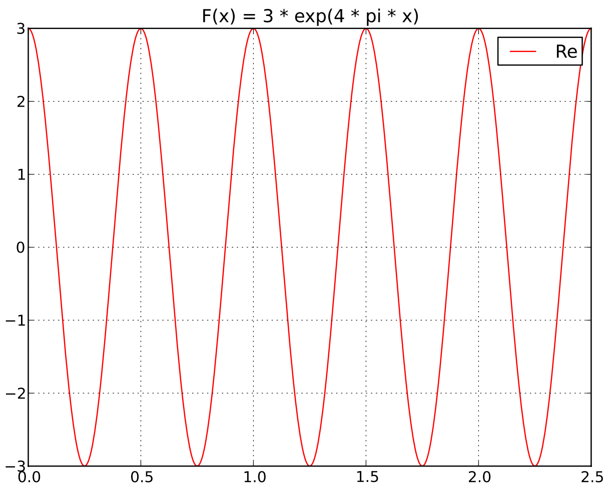
dft.py def dft(y,inverse=False):
Führt die Matrixmultiplikation durch und gibt den transformierten diskreten Datensatz aus

$$\hat{F_m} = A_{nm}F_n$$

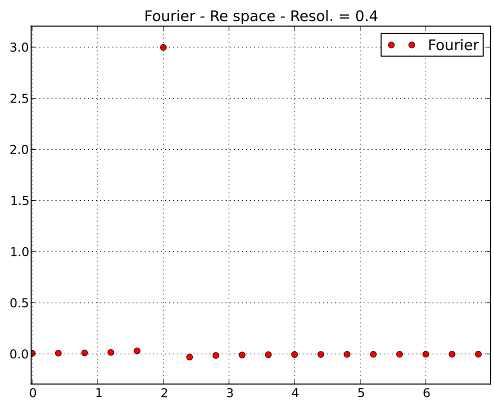
dft.py def getImageSpace(x):
Berechnet die Diskretisierung im Fourierraum

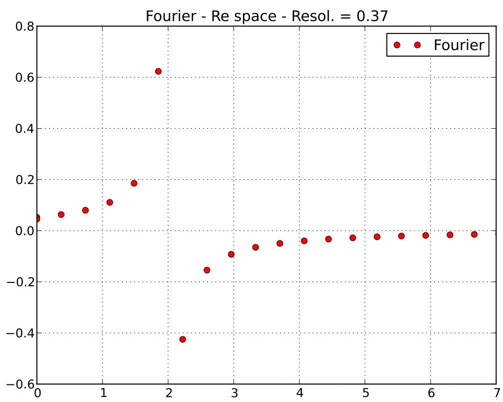
apply.py
Dieses Skript erlaubt die Anwendung der Fouriertramsformation auf verschiedene Testfunktionen. Kommentieren Sie verschieden Funktionen ein und aus, variieren Sie die Auflösung des physikalischen Raums (Zeile 4) und des Bildraums (Zeile 7).

Beispiel:
Berechnung von $u = 3. * \exp(4i * \pi * x)$



Vergleich der Auflösung von $\kappa = 0.4$ und $\kappa = 0.37$:





Verfeinerung der Auflösung: Peak nicht mehr getroffen.