

Fast Fourier Transformations

In ROOT

Paris J. Huth | 19. Jan 2024

Fourier Transformation

- Ermöglicht die Transformation zwischen Domänen einer kontinuierlichen Funktion $f(x)$

- $$\hat{f}(k) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi kx} dx$$

- $$f(x) = \int_{-\infty}^{\infty} \hat{f}(k) e^{i2\pi kx} dk$$

Discrete Fourier Transformation

Umgang mit diskreten Werten

- Ansatz für die Transformation von diskreten Daten

- $$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi \frac{k}{N} n}$$

- Komplexität $O(n^2)$

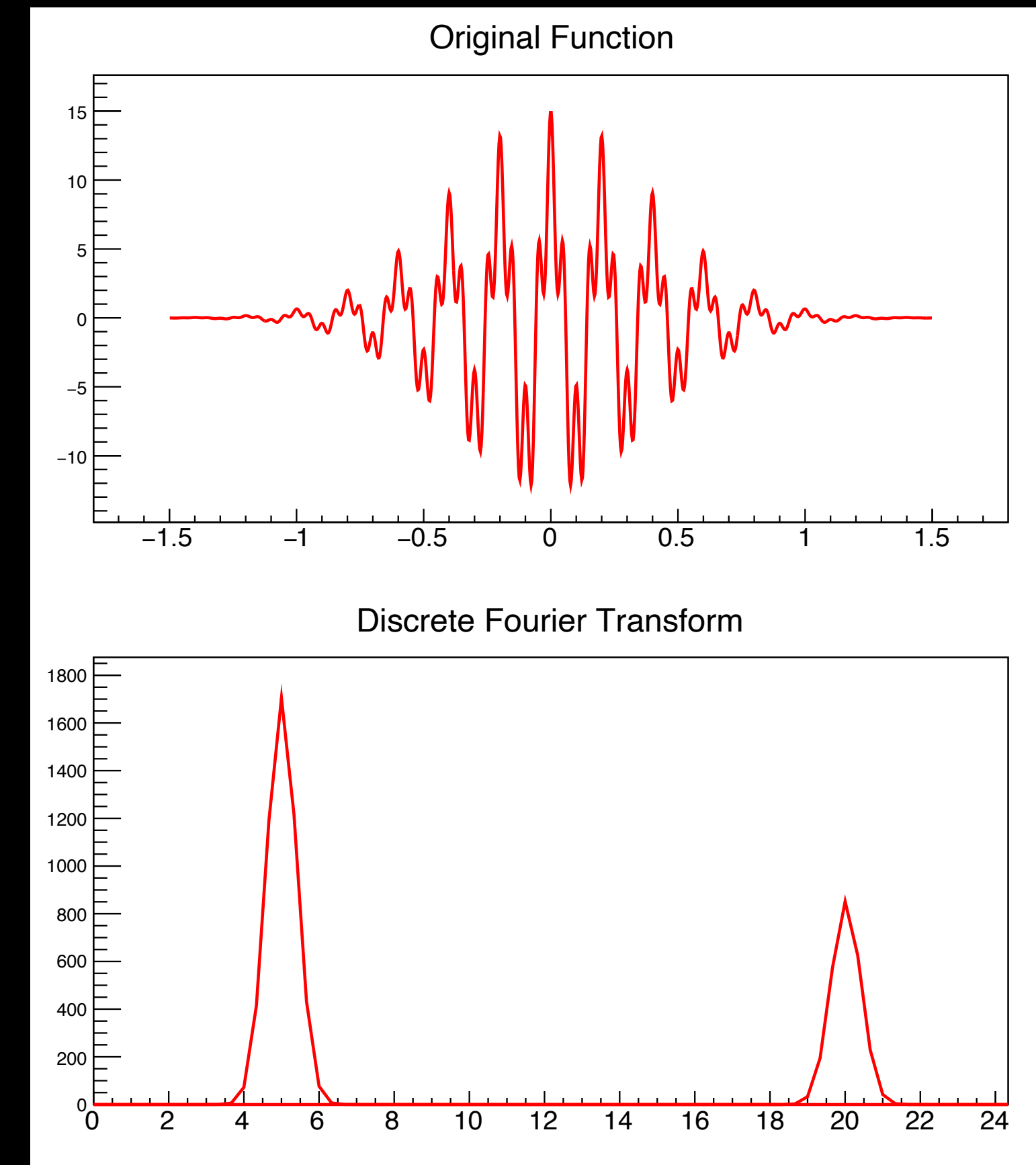
Discrete Fourier Transform Implementation

```
void dft(double *input, complex<double> *output, int size) {  
    for (int k=0; k< size; k++) {  
        output[k] = 0.0;  
        for (int n=0; n < (size-1); n++) {  
            double angle = -2.0 * pi * k * n /size;  
            output[k] += input[n] * complex<double>(cos(angle), sin(angle));  
        }  
    }  
}
```

Darstellung 1. Implementation einer DFFT

Simple Beispiel:

$$f(x) = \left[10 \cdot \cos(2\pi(5x)) + 5 \cdot \cos(2\pi(20x)) \right] \cdot e^{-\pi x^2}$$



Darstellung 2. DFFT

Fast Fourier Transformation

- Algorithmus nach James Cooley und John W. Tukey:

$$\bullet f_m = \sum_{k=0}^{2n-1} x_k e^{-\frac{2\pi i}{2n}mk} = \sum_{k=0}^{n-1} x_{2k} e^{-\frac{2\pi i}{2n}m(2k)} + \sum_{k=0}^{n-1} x_{2k+1} e^{-\frac{2\pi i}{2n}m(2k+1)}$$

- Komplexität $O(n \ln n)$

Fast Fourier Transformation

Implementation

```
void fft(complex<double> *input, complex<double> *output, int size, int step)
{
    if (size == 1) {
        output[0] = input[0];
        return;
    }

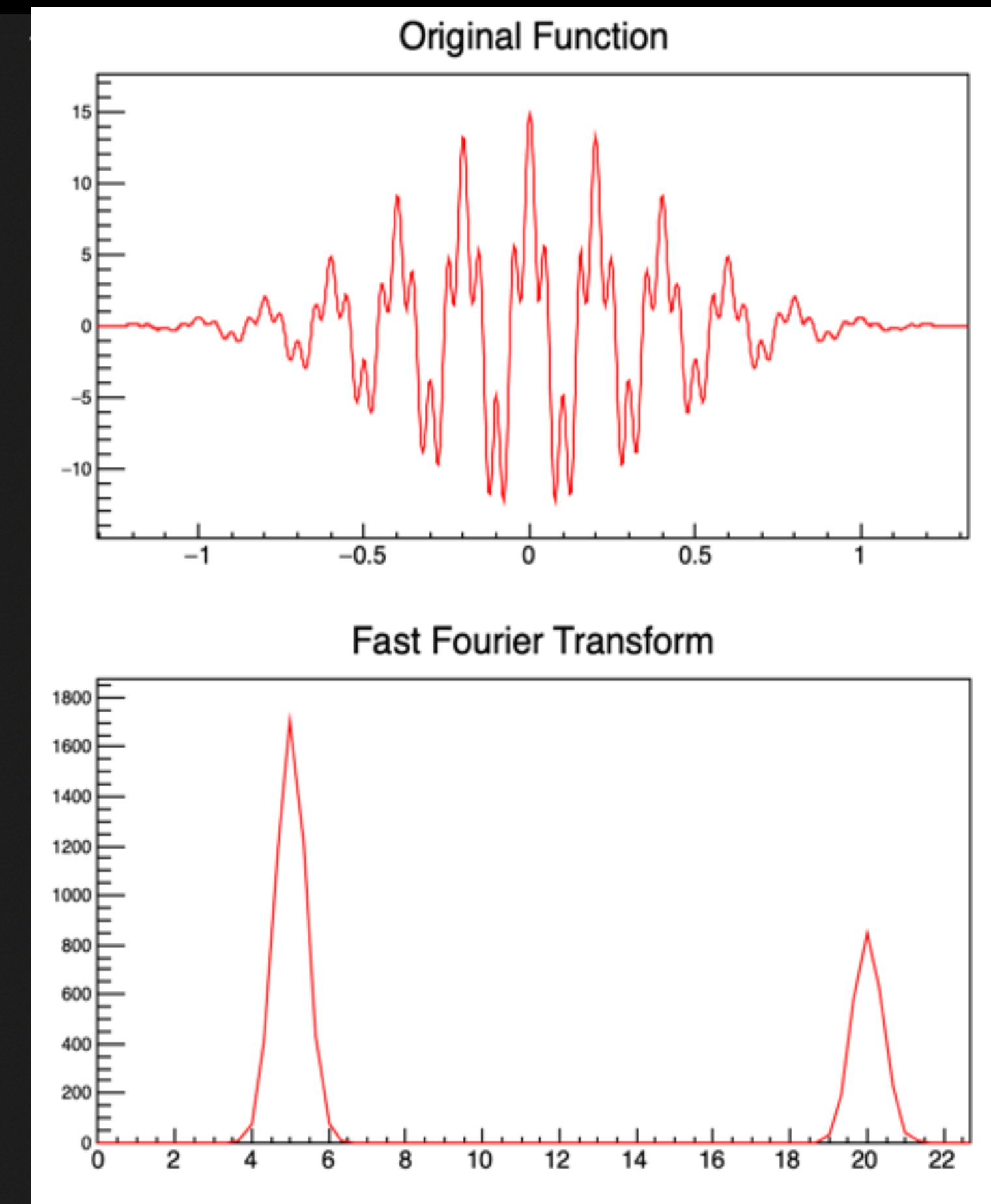
    int halfSize = size / 2;
    fft(input, output, halfSize, 2 * step);
    fft(input + step, output + halfSize, halfSize, 2 * step);

    for (int i = 0; i < halfSize; i++) {
        complex<double> even = output[i];
        complex<double> odd = output[i + halfSize];
        complex<double> t = polar(1.0, -2.0 * pi * i / size) * odd;

        output[i] = even + t;
        output[i + halfSize] = even - t;
    }
}

void fft(complex<double> *input, complex<double> *output, int size) {
    fft(input, output, size, 1);
}
```

Darstellung 3. Implementation FFT



Darstellung 4. FFT

FFT's in ROOT

Verfügbare Klassen

- TFFTRealComplex, TFFTReal, TFFTComplex, TFFTComplexReal
 - Simple Interface für FFTW (Fastest Fourier Transform in the West)
 - Z.B. nur für reellwertige Eingabe- und komplexwertige Ausgabewerte
- TVirtualFFT
 - Verschiedenen FFT implementationen (Default: FFTW)
 - Erbt von allen FFT implementation in ROOT

TFFTRealComplex

Workflow

- "ES" (from "estimate") - no time in preparing the transform, but probably sub-optimal performance
- "M" (from "measure") - some time spend in finding the optimal way to do the transform
- "P" (from "patient") - more time spend in finding the optimal way to do the transform
- "EX" (from "exhaustive") - the most optimal way is found

Darstellung 5. Init flags

- TFFTRealComplex **fftInstance**(size);
- fftInstance.**Init**(someFlags);
- fftInstance.**SetPoints**(data);
 - fftInstance.**SetPoint**(data);
 - fftInstance.**SetPointComplex**(data);
 - ...
- fftInstance.**Transform**();
- fftInstance.**GetPoints**();
 - fftInstance.**GetPoint**();
 - ...

TVirtualFFT

Workflow

- TVirtualFFT *fftInstance =
TVirtualFFT::FFT(dim, size, "flags");
- fftInstance ->SetPoints(data);
- fftInstance ->Transform();
- fftInstance ->GetPoints();

```
SetDefaultFFT();
```

Darstellung 6. Library wechseln

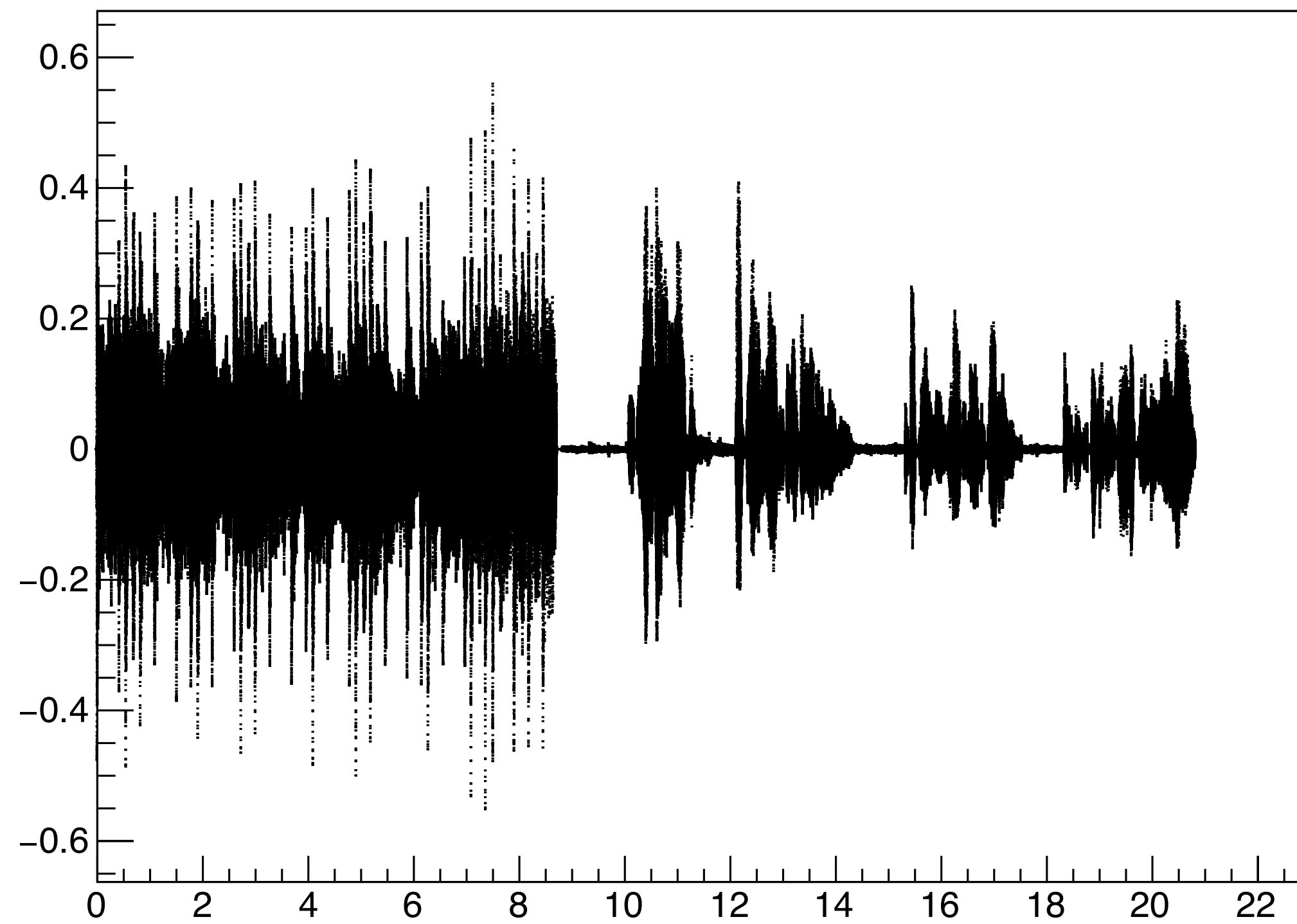
```
C2CFORWARD  
C2CBACKWARD  
R2C  
C2R  
R2HC  
HC2R  
DHT
```

Darstellung 7. Verfügbare
Transformationen

Anwendung

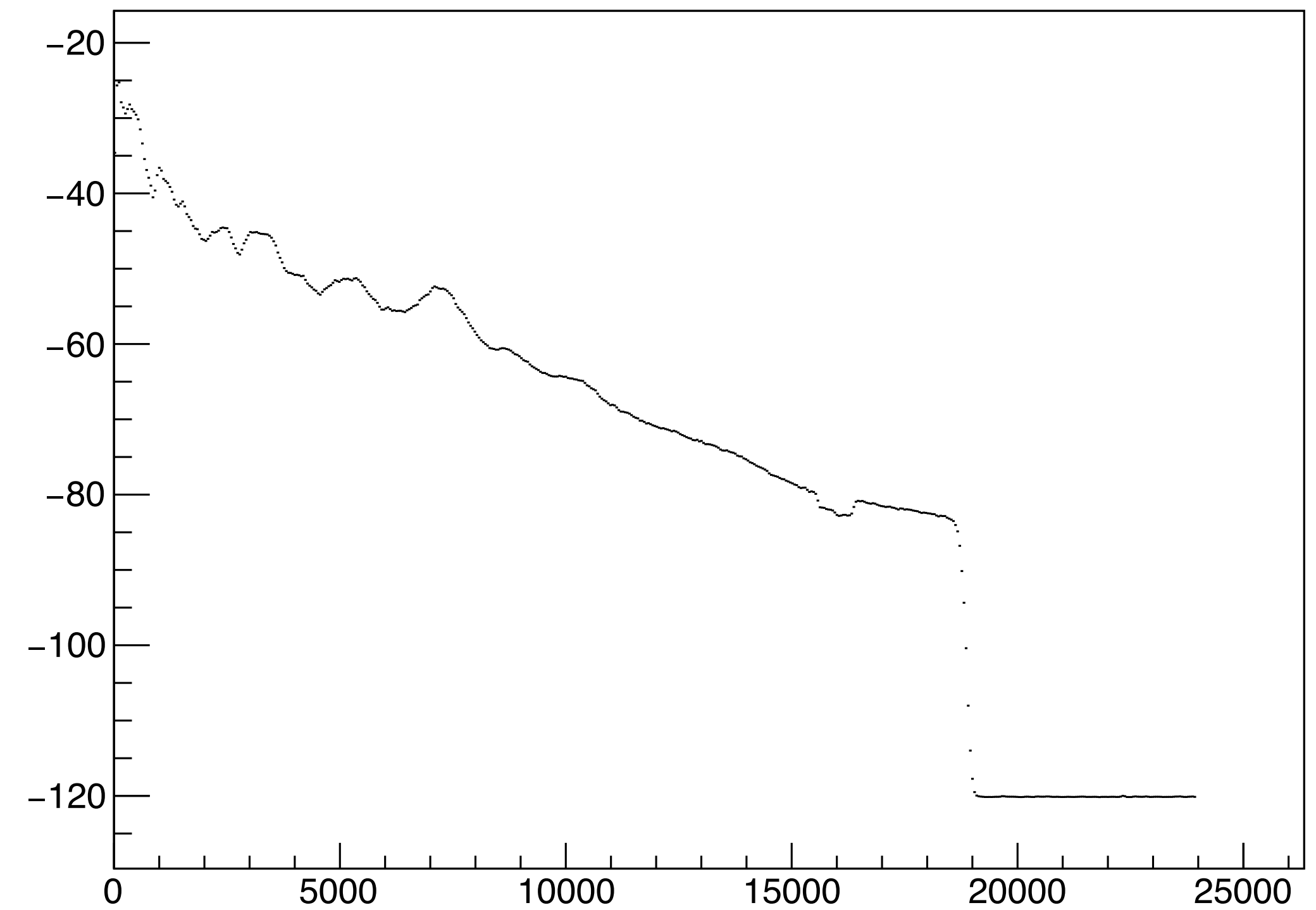
Audio

Audio Signal



Darstellung 8. Input Signal

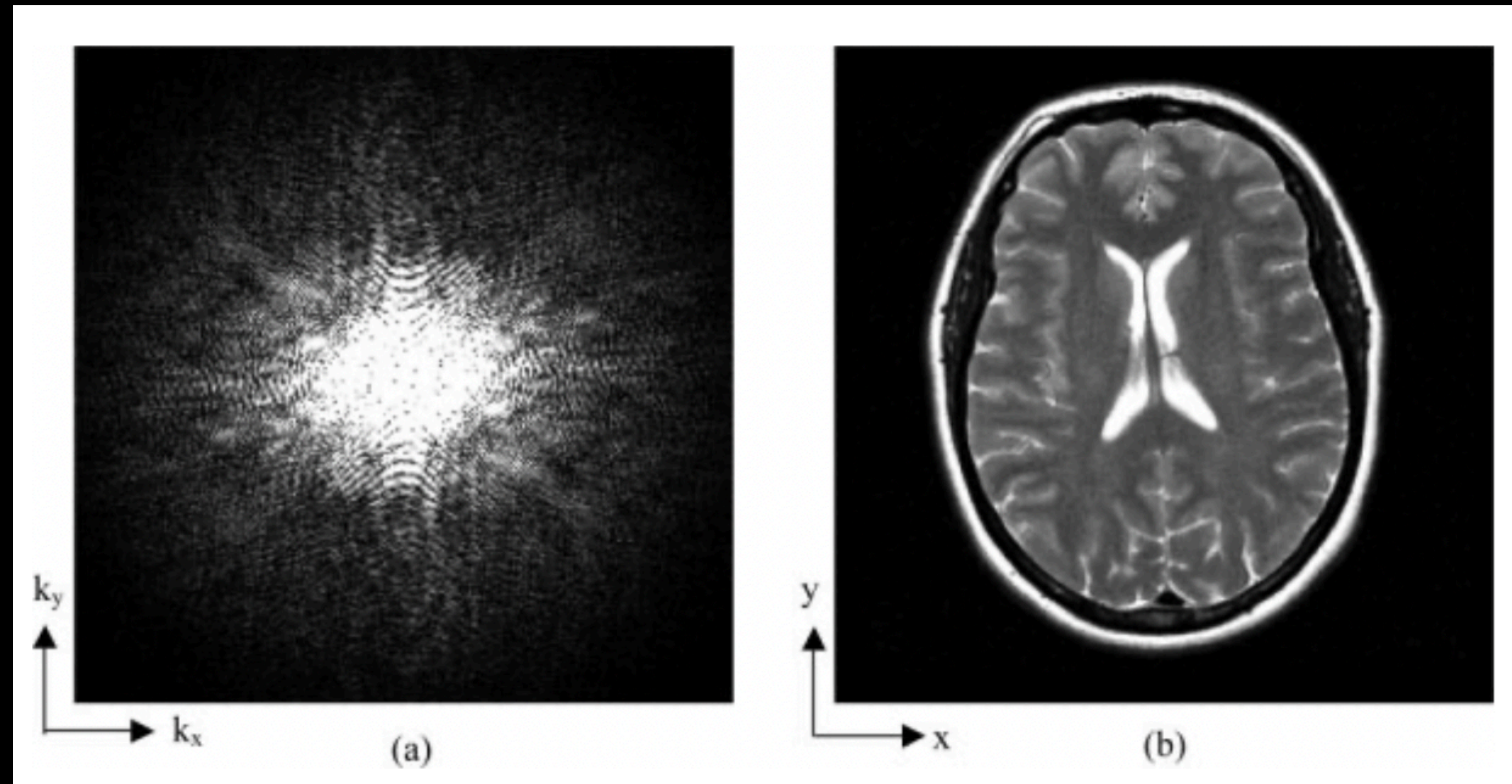
Fourier Transformation



Darstellung 9. Transformiertes Signal

2-dim FFT

Beispiel: MRT



Bildgebung in der
Magnetresonanztomographie.
Transformation vom
k-Raum in den
Bildraum.

Darstellung 10. Bildgebung im MRT (Paschal, Morris 2004)

Quellen

<https://root.cern.ch/doc/master/classTFFTRealComplex.html>

<https://root.cern.ch/doc/master/classTVirtualFFT.html>

https://youtu.be/A91ji_RTNLU?si=eHTw-1uIIWJgdimx

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

<https://courses.cs.washington.edu/courses/cse373/16au/slides/23-Divide-and-Conquer-the-FFT-6up.pdf>

https://en.wikipedia.org/wiki/Cooley–Tukey_FFT_algorithm

Paschal CB, Morris HD. 2004. K-Space in the Clinic, JOURNAL OF MAGNETIC RESONANCE IMAGING 19:145–159