Medical Imaging

Prof. Dr. Tobias Knopp

14. Dezember 2022

Institut für Biomedizinische Bildgebung

Non-Equidistant Fast Fourier Transform

Non-Equidistant Fast Fourier Transform

The non-equidistant fast Fourier transform (NFFT) is an approximative algorithm that performs the non-equidistant discrete Fourier transform (NDFT) efficiently.

Definition
The NDFT is defined as

$$f_j = f(x_j) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{f}_k \mathrm{e}^{-2\pi \mathrm{i} k x_j}, \quad j = 0, \ldots, M-1$$

It takes a sequence of N equidistantly distributed samples $\hat{f}_k$ and calculates the Fourier sum at M non-equidistant sampling nodes $x_j \in [-0.5, 0.5)$.

Non-Equidistant Fast Fourier Transform

Remarks

- Naive NDFT would require $\mathcal{O}(MN)$ arithmetic operations
- The FFT requires x_j to be equidistant, i.e.

$$x_j = -\frac{1}{2} + \frac{j}{N}, \qquad j = 0, \ldots, N-1$$

Key Idea of the NFFT

The key idea of the NFFT is to approximate the *anharmonic* complex exponential $\mathrm{e}^{-2\pi\mathrm{i}kx_j}$ by a sum of harmonic complex exponentials

$$\underbrace{\mathrm{e}^{-2\pi\mathrm{i}kx_j}}_{\text{anharmonic}} \approx \sum_{l=-\frac{L}{2}}^{\frac{L}{2}-1} \beta_{l,j}\, \underbrace{\mathrm{e}^{-2\pi\mathrm{i}\frac{kl}{L}}}_{\text{harmonic}}$$

Remarks

- Harmonic here is means that the frequency $\gamma_l = \frac{l}{L}$ is a multiple of the basis frequency $\gamma_1 = \frac{1}{L}$.
- Similar to other approximations (or interpolations) we use a base function and shift its frequency (c.f. spline interpolation).
- Most important point of the approximation is that the two indices k and j are not in the exponent anymore but they are factorized.

Derivation of the NFFT

We consider a general window function $\varphi \in L^1(\mathbb{R}) \cap L^2(\mathbb{R}) \cap \mathrm{BV}(\mathbb{R})$ and its periodization

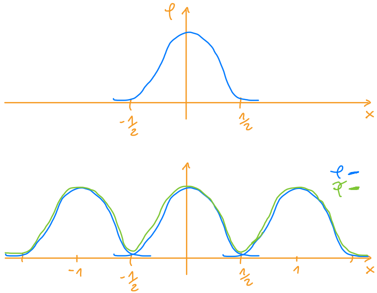
$$\tilde{\varphi}(\tilde{x}) := \sum_{p=-\infty}^{\infty} \varphi(\tilde{x} + p).$$

The function has a uniformly convergent Fourier series

$$\tilde{\varphi}(\tilde{x}) = \sum_{k=-\infty}^{\infty} c_k(\tilde{\varphi}) \mathrm{e}^{-2\pi \mathrm{i} k \tilde{x}}$$

with coefficients

$$c_k(\tilde{\varphi}) = \int_{-\frac{1}{2}}^{\frac{1}{2}} \tilde{\varphi}(\tilde{x}) \mathrm{e}^{2\pi \mathrm{i} k \tilde{x}} \, \mathrm{d}\tilde{x}$$



Note: The sign is flipped compared to the regular definition of the Fourier series.

Derivation of the NFFT

We now substitute $\tilde{x}$ by $\tilde{x} = x - x'$ with $x \in \mathbb{R}$ yielding with $\frac{\mathrm{d}\tilde{x}}{\mathrm{d}x'} = -1$

$$\begin{aligned}
c_k(\tilde{\varphi}) &= \int_{x+\frac{1}{2}}^{x-\frac{1}{2}} \tilde{\varphi}(x - x')\mathrm{e}^{2\pi\mathrm{i}k(x-x')}(-1)\,\mathrm{d}x' \\
&\overset{\text{periodicity}}{=} \int_{\frac{1}{2}}^{-\frac{1}{2}} \tilde{\varphi}(x - x')\mathrm{e}^{2\pi\mathrm{i}k(x-x')}(-1)\,\mathrm{d}x' \\
&= \int_{-\frac{1}{2}}^{\frac{1}{2}} \tilde{\varphi}(x - x')\mathrm{e}^{2\pi\mathrm{i}k(x-x')}\,\mathrm{d}x'
\end{aligned}$$

Derivation of the NFFT

Then we approximate the integral at equidistant nodes $\frac{l}{L}$ using a rectangular quadrature rule

$$c_k(\tilde{\varphi}) \approx \frac{1}{L} \sum_{l=-\frac{L}{2}}^{\frac{L}{2}-1} \tilde{\varphi}\left(x - \frac{l}{L}\right) \mathrm{e}^{2\pi \mathrm{i} k (x - \frac{l}{L})}$$

We will later chose $L = \alpha N$ where $\alpha > 1$ is the so-called *oversampling factor*. If $c_k(\tilde{\varphi}) \neq 0$ we obtain

$$\mathrm{e}^{-2\pi \mathrm{i} k x} \approx \frac{1}{\alpha N c_k(\tilde{\varphi})} \sum_{l=-\frac{\alpha N}{2}}^{\frac{\alpha N}{2}-1} \tilde{\varphi}\left(x - \frac{l}{\alpha N}\right) \mathrm{e}^{-2\pi \mathrm{i} k \frac{l}{\alpha N}} \tag{1}$$

Thus we can indeed approximate the anharmonic exponential by a sum of harmonic exponentials ($\beta_{l,j} = \frac{1}{\alpha N c_k(\tilde{\varphi})} \tilde{\varphi}\left(x - \frac{l}{\alpha N}\right)$).

Derivation of the NFFT

Since φ is a kernel function we know that only few of the summands in (1) have a significant contribution to the sum.

Key idea: Consider only significant summands such that the sum needs just be evaluated partially.

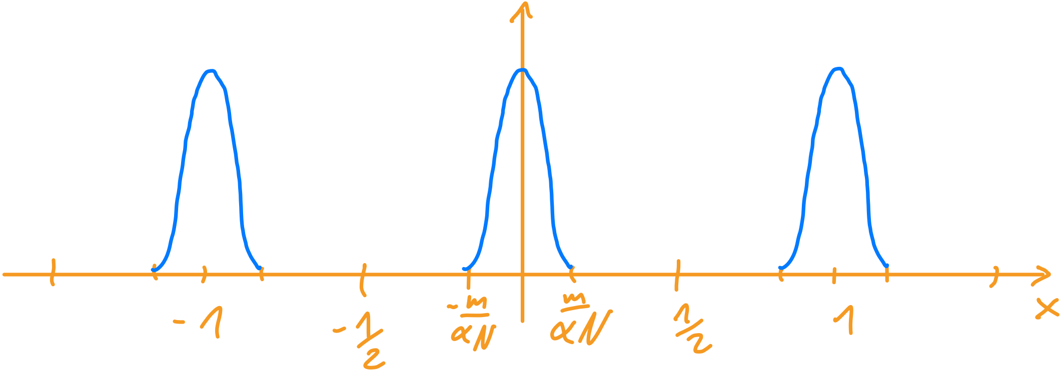
To this end we truncate the function φ at $\pm\frac{m}{\alpha N}$ and replace it by

$$\psi(x) := \begin{cases} \varphi(x) & \text{if } x \in \left[-\frac{m}{\alpha N}, \frac{m}{\alpha N}\right] \\ 0 & \text{else} \end{cases}.$$

With the periodization $\tilde{\psi}(\tilde{x}) := \sum_{p=-\infty}^{\infty} \psi(\tilde{x}+p)$ this yields our final approximation

$$\mathrm{e}^{-2\pi \mathrm{i} kx} \approx \frac{1}{\alpha N c_k(\tilde{\varphi})} \sum_{l=-\frac{\alpha N}{2}}^{\frac{\alpha N}{2}-1} \tilde{\psi}\left(x - \frac{l}{\alpha N}\right) \mathrm{e}^{-2\pi \mathrm{i} k \frac{l}{\alpha N}} \tag{2}$$

Illustration of the truncation:



Derivation of the NFFT

We now get back to the NDFT and replace the term $\mathrm{e}^{-2\pi\mathrm{i}kx}$ with our approximation yielding

$$
\begin{aligned}
f_j &= \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{f}_k \mathrm{e}^{-2\pi\mathrm{i}kx_j} \\
&\overset{(2)}{\approx} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{f}_k \frac{1}{\alpha N c_k(\tilde{\varphi})} \sum_{l=-\frac{\alpha N}{2}}^{\frac{\alpha N}{2}-1} \tilde{\psi}\left(x_j - \frac{l}{\alpha N}\right) \mathrm{e}^{-2\pi\mathrm{i}k\frac{l}{\alpha N}} \\
&= \underbrace{\sum_{l=-\frac{\alpha N}{2}}^{\frac{\alpha N}{2}-1} \tilde{\psi}\left(x_j - \frac{l}{\alpha N}\right) \underbrace{\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \underbrace{\frac{\hat{f}_k}{\alpha N c_k(\tilde{\varphi})}}_{\text{apodization}} \mathrm{e}^{-2\pi\mathrm{i}k\frac{l}{\alpha N}}}_{\text{DFT/FFT}}}_{\text{discrete convolution}}
\end{aligned}
$$

Algorithm 1 Pseudocode NFFT

input: $\hat{f}_k \in \mathbb{C}$, $k = -\frac{N}{2}, \ldots, \frac{N}{2} - 1$, $x_j \in [-\frac{1}{2}, \frac{1}{2})$, $j = 0, \ldots, M-1$, $\alpha > 1$ and $m \in \mathbb{N}$

output: $f_j \in \mathbb{C}$, $j = 0, \ldots, M-1$

1: **for** $k = -\frac{N}{2}, \ldots, \frac{N}{2} - 1$ **do**

2: $\quad \hat{g}_k = \frac{\hat{f}_k}{\alpha N c_k(\tilde{\varphi})}$

3: **end for**

4: compute the data $(g_l)_{l=-\frac{\alpha N}{2}}^{\frac{\alpha N}{2}-1}$ using an FFT of $(\hat{g}_k)_{k=-\frac{N}{2}}^{\frac{N}{2}-1}$.

5: **for** $j = 0, \ldots, M-1$ **do**

6: $$f_j = \sum_{l=-\frac{\alpha N}{2}}^{\frac{\alpha N}{2}-1} g_l \tilde{\psi}\left(x_j - \frac{l}{\alpha N}\right)$$

7: **end for**

Complexity Analysis

The three steps have an individual time complexity of

1. $\mathcal{O}(N)$
2. $\mathcal{O}(\alpha N \log(\alpha N))$
3. $\mathcal{O}(mM)$

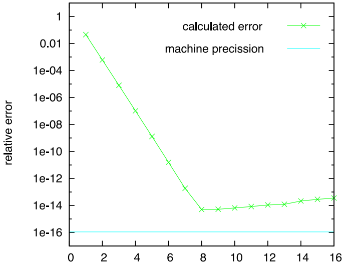
Thus, the total complexity of the NFFT is

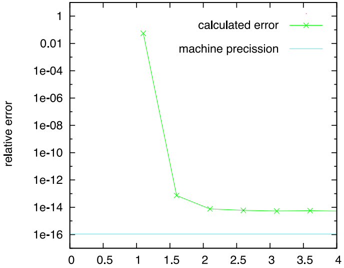
$$\mathcal{O}(\alpha N \log(\alpha N) + mM) \ll \mathcal{O}(NM)$$

Approximation Error

One can derive approximation error estimations for the NFFT. For specific φ (i.e. Kaiser-Bessel functions) one can show that the approximation error can be adjusted to be lower than tha floating point precission (64 bit $\rightarrow \alpha = 2$ and $m = 6$ for the Kaiser-Bessel window).
Since α and m are independent of N and M we end up with an algorithmic complexity of

$$\mathcal{O}(N \log N + M) \ll \mathcal{O}(NM)$$





Matrix-Vector Notation

The NFFT can also been expressed in matrix vector notation. Let

$$\hat{\boldsymbol{f}} := (\hat{f}_k)_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \in \mathbb{C}^N, \quad \boldsymbol{f} := (f_j)_{j=0}^{M-1} \in \mathbb{C}^M$$

and

$$\boldsymbol{A} := \left(\mathrm{e}^{-2\pi \mathrm{i} k x_j}\right)_{j=0,\ldots,M-1;\, k=-\frac{N}{2},\ldots,\frac{N}{2}-1} \in \mathbb{C}^{M\times N}.$$

then

$$\boldsymbol{f} = \boldsymbol{A}\hat{\boldsymbol{f}} \approx \underbrace{\boldsymbol{B}}_{\text{convolution matrix}} \underbrace{\boldsymbol{F}}_{\text{DFT matrix}} \underbrace{\boldsymbol{D}}_{\text{diagonal matrix}} \hat{\boldsymbol{f}}.$$

Here we note that $\boldsymbol{B}$ is a sparse matrix (i.e. has only few non-zero entries).

Implementation

The window function φ and $c_k(\tilde{\varphi})$ are usually expensive to calculate and should therefore be cached. There are two possibilities for φ

1. Create a lookup table for φ
2. Store $\boldsymbol{B}$ in a sparse matrix format (CRS / CCS)

Adjoint NFFT

In addition to the regular NFFT one often also needs the adjoint NFFT. It maps from non-equidistant samples to equidistant samples, wheras the NFFT is the other way around.

$$\hat{f}_k = \sum_{j=0}^{M-1} f_j \mathrm{e}^{2\pi \mathrm{i} k x_j} \quad k = -\frac{N}{2}, \ldots, \frac{N}{2} - 1$$

$$\approx \underbrace{\frac{1}{\alpha N c_k(\tilde{\varphi})} \underbrace{\sum_{l=-\frac{\alpha N}{2}}^{\frac{\alpha N}{2}-1} \underbrace{\left(\sum_{j=0}^{M-1} f_j \tilde{\psi}\left(x_j - \frac{l}{\alpha N} \right) \right)}_{\text{discrete convolution}} \mathrm{e}^{2\pi \mathrm{i} k \frac{l}{\alpha N}}}_{\text{DFT/FFT}}}_{\text{apodization}}$$

Adjoint NFFT

In matrix vector notation:

$$\hat{\boldsymbol{f}} = \boldsymbol{A}^{\mathrm{H}} \boldsymbol{f} \approx \boldsymbol{D}^{\mathrm{H}} \boldsymbol{F}^{\mathrm{H}} \boldsymbol{B}^{\mathrm{H}} \boldsymbol{f}$$

Window Function (Kaiser-Bessel)

There are various suitable window functions for which error estimations have been derived. The best is the Kaiser-Bessel window, which is defined as

$$\varphi(v) := \begin{cases} \frac{1}{2m} I_0\left(bm\sqrt{1-(\frac{\alpha N v}{m})^2}\right) & \text{falls} \quad |v| \leq \frac{m}{\alpha N} \quad (b := \pi(2-\frac{1}{\alpha})), \\ 0 & \text{falls} \quad |v| > \frac{m}{\alpha N} \end{cases}$$

where $I_k : \mathbb{C} \to \mathbb{C}$, $k \in \mathbb{N}_0$ is the modified Bessel function of the first kind:

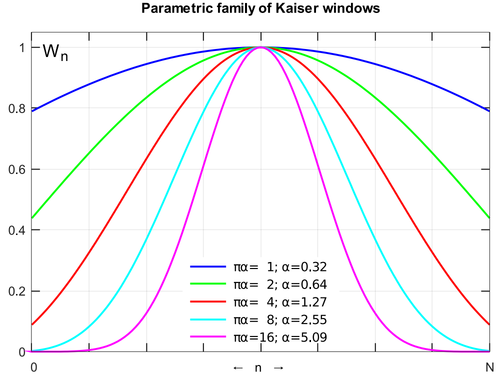
$$I_k(x) := \sum_{r=0}^{\infty} \frac{(\frac{x}{2})^{2r+k}}{(r+k)!r!}.$$

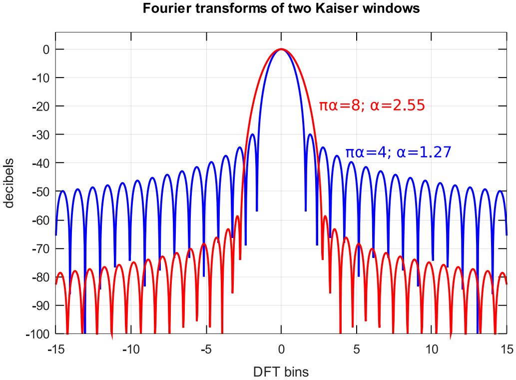
Window Function (Kaiser-Bessel)

The Fourier transform of the Kaiser-Bessel window can be shown to be

$$\begin{aligned}\hat{\varphi}(z) &= \frac{1}{\alpha N}\text{sinc}\left(\sqrt{\left(\frac{2\pi mz}{\alpha N}\right)^2 - b^2 m^2}\right) \\ &= \frac{1}{\alpha N}\sum_{n=0}^{\infty}\frac{1}{(2n+1)!}\left(b^2 m^2 - \left(\frac{2\pi mz}{\alpha N}\right)^2\right)^n.\end{aligned}$$

Window Function (Kaiser-Bessel)





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

Multidimensional NFFT

The NDFT and the NFFT can be also formulated / derived for multidimensional signals:

NFFT

$$f_j := \sum_{\boldsymbol{k} \in I_{\boldsymbol{N}}^d} \hat{f}_{\boldsymbol{k}} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_{\boldsymbol{j}}}, \qquad j = 0, \ldots M-1$$

Adjoint NFFT

$$\hat{f}_{\boldsymbol{k}} = \sum_{j=0}^{M-1} f_j \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_j}, \qquad \boldsymbol{k} \in I_{\boldsymbol{N}}^d$$

where the index set $I_{\boldsymbol{N}}^d$ with $\boldsymbol{N} = (N_0, \ldots, N_{d-1})^T \in \mathbb{N}^d$ is defined as

$$I_{\boldsymbol{N}}^d := \left\{-\frac{N_1}{2}, \ldots, \frac{N_1}{2} - 1\right\} \times \cdots \times \left\{-\frac{N_d}{2}, \ldots, \frac{N_d}{2} - 1\right\}$$

and d is the dimensionality of the tranform.

Inverse NFFT

In general the adoint NFFT is not (exactly) the inverse NFFT, i.e.

$$\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A} \neq \boldsymbol{I}$$

However, one can derive an approximation to the (pseudo)inverse quite efficiently.

To this end we first consider the NDFT

$$f(x_j) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{f}_k \mathrm{e}^{-2\pi \mathrm{i} k x_j}, \quad j = 0, \ldots, M-1$$

We now extend the sum to $\pm\infty$, which leads to the Fourier series

$$f(x_j) = \sum_{k=-\infty}^{\infty} \hat{f}_k \mathrm{e}^{-2\pi \mathrm{i} k x_j}, \quad j = 0, \ldots, M-1$$

where the coefficients $\hat{f}_k$ have been zero-padded.

Inverse NFFT

The Fourier coefficients $\hat{f}_k$ can be calculated by

$$\hat{f}_k = \int_{-\frac{1}{2}}^{\frac{1}{2}} f(x) \mathrm{e}^{2\pi \mathrm{i} k x_j} \, \mathrm{d}x \qquad k = -\frac{k}{2}, \dots, \frac{k}{2} - 1$$

Since f is only known at the sampling nodes x_j, we can only consider these when approximating the integral by a sum. When applying a rectangular quadrature rule, one obtains

$$\hat{f}_k \approx \sum_{j=0}^{M-1} w_j f(x_j) \mathrm{e}^{2\pi \mathrm{i} k x_j} \qquad k = -\frac{k}{2}, \dots, \frac{k}{2} - 1$$

where w_j, $j = 0, \dots, M-1$ are the quadrature weights. This is the adjoint NFFT with a pre-weighting. In matrix-vector notation this implies $\boldsymbol{A}^{\mathrm{H}} \boldsymbol{W} \approx \boldsymbol{A}^{+}$, i.e. $\boldsymbol{A}^{\mathrm{H}} \boldsymbol{W} \boldsymbol{A} \approx \boldsymbol{I}$.

Summary

- The NDFT is a generalization of the DFT
- The NFFT is an efficient implementation of the NDFT, which exploits a numerical approximation of the complex exponential
- The approximation error is known and can be adjusted to reach machine precision.
- In practice the convolution usually takes most of the computation time. With optimized parameters ($\alpha = 1.25$, $m = 2$) it is possible to make the convolution as fast as the FFT.
- There are various implementations of the NFFT. One reference implementation is the C library NFFT 3 (`https://github.com/NFFT/nfft`). Also a Julia package exists: `https://github.com/tknopp/NFFT.jl`