

CAVEAT: *I prepared this document mostly from the mathematical derivation found in the documentation for the NFFT (NUFFT) library provided by Chemnitz. I considered some of the explanations in that document to be insufficient, and some more complicated steps of their derivations were not included. In addition, the Chemnitz paper derives the NUFFT, and makes assumptions about the FFT, in such a way that what are typically referred to as time/space and frequency have their roles reversed; in their notation, the FFT takes what they call Fourier coefficients (and we call signal samples) and computes coefficients of trigonometric polynomials for the time/space representation (which we would normally consider to be the Fourier coefficients). Therefore, in an effort to foster a better understanding, I took the liberty of essentially re-organizing and re-writing their document to fix these problems, as well as include some information from other sources. To the best of my knowledge, what I have written here is an accurate revision of their paper, but it is quite possible that there are flaws in this document, so it should be taken with some discretion. Hopefully it will serve as a good reference on this topic and make the ideas behind the non-uniform DFT/FFT clear.*

*As an additional note, the last two sections on the inversion of the NUFFT is purposefully lacking in depth because the Chemnitz paper describes an algorithm for conjugate gradients in two cases without making all of their notation clear. Rather than guess at their errors or what some symbols may mean, since the majority of the people listening to the presentation have experience with CG, I will simply assume you would be able to see the application. For those that don't have experience with CG, I refer you to the Chemnitz paper, which I will send along with this document, and hope you have better luck with their notation than I. Alternatively, you can obtain a copy of An Introduction to the Conjugate Gradient Method Without the Agonizing Pain by Jonathan Richard Shewchuk and get a much better treatment of CG.*

## 1 Introduction to the NUDFT

In many scientific applications, it is useful to analyze a frequency representation of data that has been gathered in the time/space domain, rather than analyze the gathered data values themselves. For this reason, the Fourier Transform, and specifically the FFT, have become important tools in many fields. A severe disadvantage of Fourier techniques, however, is their dependence on uniformity in both domains; that is, traditional Fourier techniques require that data be measured on a uniform grid in the time/space domain, and transform that data into a representation on a uniform grid in the frequency domain. While this is acceptable for a large number of applications, it is occasionally useful to gather data that is unequally spaced in either domain, and in some applications, this is the only way data can be collected. Therefore, it is desirable to generalize Fourier techniques to handle non-uniform grids.

The most obvious way to do this is through simple interpolation, which is essentially what the non-uniform DFT (NUDFT) and FFT (NUFFT) employ. In the NUDFT, data is assumed to be collected at uniform locations in time/space and is transformed onto a non-uniform grid in the frequency domain through (essentially) interpolation. The NUDFT can be generalized as the NNFFT, which uses non-uniform grids in both time and frequency. This paper will explore the NUDFT and issues surrounding it.

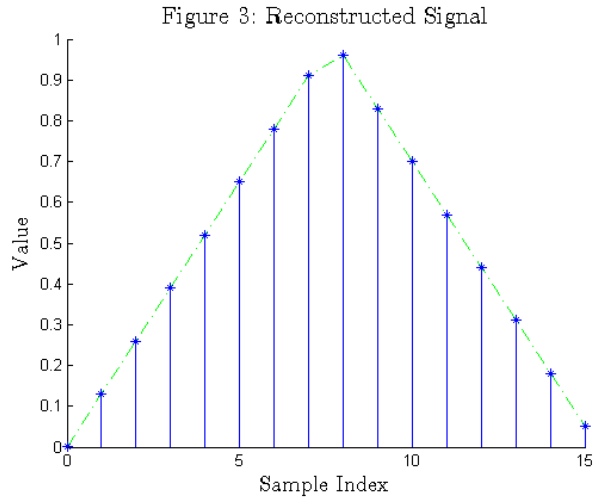
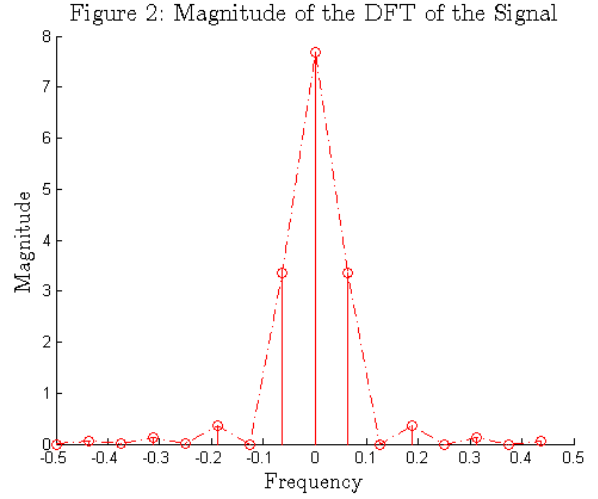
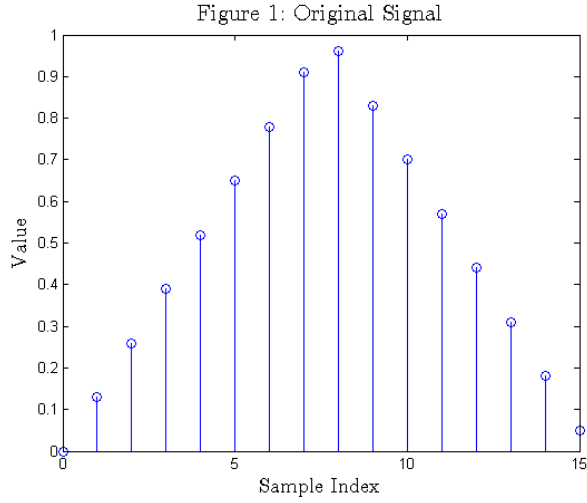
We begin by stating the general form of the NUDFT for equispaced time/space data in 1-dimension:

$$\hat{g}(f) = \sum_{k=0}^{N-1} g[k] e^{-2\pi i k f_j}$$

In this notation, the values  $g[k]$  are the sample values of  $g$  for  $k = 0, \dots, N-1$ , and the frequency  $f$  is the frequency dependence. In the standard DFT, the coefficients are obtained for equally-spaced frequency nodes. A result of this is that the adjoint of the Fourier matrix can be used to achieve a perfect reconstruction of the original sample values. In the NUDFT, however, if the frequencies for which the coefficients are obtained are not equally spaced, the NUDFT matrix  $\mathbf{A}$  will no longer satisfy  $\mathbf{A}^\perp \mathbf{A} = N\mathcal{I}_N$ .

It is instructive to think of the NUDFT as two processes that occur in serial, though in reality these processes occur simultaneously. In the first process, the function samples obtained from the uniformly spaced grid in time/space are used to compute the standard DFT, resulting in the determination of Fourier coefficients on a uniform grid in the frequency domain. These coefficients are then used to interpolate a continuous function in the frequency domain which is subsequently sampled at the desired frequency locations. It is easy to see how the quality of the reconstruction changes in light of this. If the frequency locations given to the NUDFT are equispaced, then the output of the DFT process requires no interpolation and is the total system output. When the frequency locations are not equispaced, then interpolation is required, which will result in an approximation of the Fourier coefficients at these locations. The quality of the approximation is determined by how far off from equidistant the spacing is and the behavior of the transformed function.

The effect of shifting the frequency sample locations can be easily illustrated. Suppose we begin with the signal shown in Figure 1. Figure 2 shows the NUDFT with equispaced frequency nodes, which is simply the DFT, and Figure 3 shows the approximation to the signal when the adjoint of the NUDFT matrix  $\mathbf{A}$  is applied, which is a perfect reconstruction since we are using the DFT.



Now, if we change the NUDFT matrix to contain unevenly spaced nodes, different results are seen. In Figure 4, the magnitude of the NUDFT of the original signal is shown, with the original DFT magnitude displayed in the dashed lines. The frequencies are the evenly spaced frequencies with the addition of random noise, uniformly distributed between 0 and 0.1. Figure 5 shows the reconstruction using the adjoint of the new NUDFT matrix. The signal is significantly different from the original signal, which is shown in a green line on the same figure.

Figure 2: Magnitude of the NUDFT of the Signal

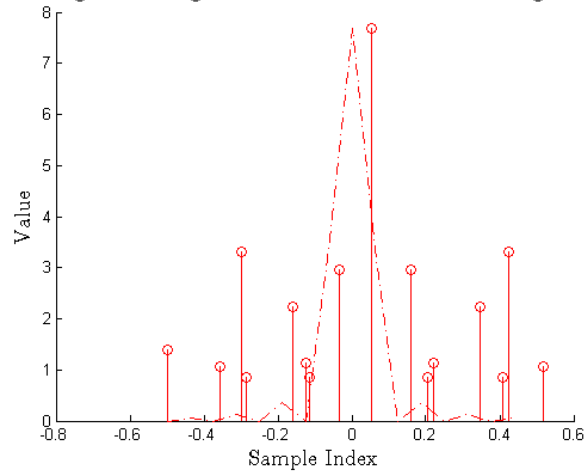
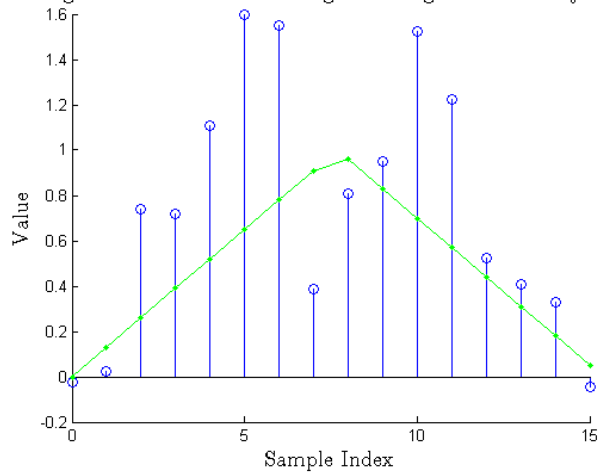


Figure 5: Reconstructed Signal Using NUDFT Adjoint



In this case, the node distribution was reasonably far from uniform, so it is not surprising that the reconstruction with the adjoint is so poor. As will be discussed later, the adjoint can still be used to achieve adequate (or perfect, depending on the case) reconstruction of the original signal using techniques such as the conjugate gradient method.

For additional examples and explanation on the general behavior in reconstructing a signal with the adjoint, refer to:

[http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/PIRODDI1/NUFT/node5.html](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/PIRODDI1/NUFT/node5.html)

## 2 Notation

Before developing the non-uniformly-spaced transform methods, we would like to establish some notation. To begin with, the difference in notating  $k$  as a vector or scalar will be established as:

$$k: \text{ scalar} \quad \mathbf{k}: \text{ vector}$$

In addition, the notation  $\mathbf{kx}$  will indicate the inner product between the vectors  $\mathbf{k}$  and  $\mathbf{x}$ , given by:

$$\mathbf{kx} = k_0x_0 + k_1x_1 + \cdots + k_{d-1}x_{d-1}$$

We will also use the notation  $\odot$  to indicate a component-wise product:

$$\mathbf{k} \odot \mathbf{x} = (k_0x_0, k_1x_1, \dots, k_{d-1}x_{d-1})$$

Finally, along the same lines, we will use the following notation:

$$\mathbf{n}^{-1} = \left( \frac{1}{n_0}, \frac{1}{n_1}, \dots, \frac{1}{n_{d-1}} \right)$$

## 3 Development of the NUFFT

In this section, we will investigate the mathematical derivation of the NUFFT. The connection between the NUFFT and the FFT will be made by considering what happens when the frequency nodes for the NUFFT are equally spaced.

### 3.1 Formal Definitions

We begin our definition of the NUFFT by noting that what we are trying to accomplish is a way of representing a function of  $d$  dimensions in terms of its frequency components at irregularly spaced intervals in the frequency domain. Thus, we begin by defining the range of the  $d$ -dimensional frequency space we will be expressing the function over and the range of  $d$ -dimensional time/space over which the function is measured.

Let  $\mathbb{T}^d$  be the collection of all  $\mathbf{f} \in \mathbb{R}^d$  such that  $-\frac{1}{2} \leq f_t < \frac{1}{2}$  for  $t = 0, \dots, d-1$ , where  $f_t$  is the  $t^{\text{th}}$  component of the vector  $\mathbf{f}$ . In other words,  $\mathbb{T}^d$  is the set of all  $d$ -dimensional frequency vectors whose individual components are all greater than  $-\frac{1}{2}$  and less than  $\frac{1}{2}$ . This space will serve as the subset of  $\mathbb{R}^d$  on which we describe the Fourier transform of the function  $g(\mathbf{x})$ , which is denoted as  $\hat{g}(\mathbf{f})$ . If we want to obtain the values of  $\hat{g}$  at a total of  $M$  points from this space, we have a set of frequencies  $\mathcal{F}$ , given by:

$$\mathcal{F} = \{\mathbf{f}_j \in \mathbb{T}^d : j = 0, \dots, M-1\}$$

Now, let  $I_N$  denote a set of indexes given by:

$$I_N = \left\{ \mathbf{k} = (k_0, k_1, \dots, k_{d-1}) \in \mathbb{Z}^d : -\frac{N_t}{2} \leq k_t < \frac{N_t}{2}, t = 0, \dots, d-1, N_t \in 2\mathbb{N} \right\}$$

The set  $I_N$  can be interpreted as all  $d$ -dimensional vectors containing the indices of a  $d$ -dimensional array. The elements of this array are the samples of the function  $g(\mathbf{x})$ , and the values of  $N_t$  for  $t = 0, 1, \dots, d-1$  indicate the number of samples of  $g(\mathbf{x})$  in each of the  $d$  dimensions.

Using these definitions, we can express the space of all  $d$ -variate one-periodic functions (denoted  $\hat{g} : \mathbb{T}^d \rightarrow \mathbb{C}$ ) with degree  $N_t$  in each of the  $t$  dimensions as a span of basis functions:

$$T_N = \text{SPAN} (e^{-2\pi i \mathbf{k} \mathbf{f}} : \mathbf{k} \in I_N)$$

This formulation should be intuitive; the Discrete Fourier Transform development leads to a similar result. Having this formulation complete, we see that a 1-periodic function  $\hat{g}(\mathbf{f})$  can be written as:

$$\hat{g}(\mathbf{f}) = \sum_{\mathbf{k} \in I_N} g_{\mathbf{k}} e^{-2\pi i \mathbf{k} \mathbf{f}}$$

where the sample values ( $g_{\mathbf{k}}$ ) serve as the weights for the trigonometric polynomials. In developing the NUFFT, we are looking for a way to determine (or approximate to a good degree of accuracy) the values of  $\hat{g}$  at the locations  $\mathbf{f}_j$  for each our samples when we are given samples of the function in time/space.. In other words, we are trying to solve

$$\hat{g}(\mathbf{f}_j) = \sum_{\mathbf{k} \in I_N} g_{\mathbf{k}} e^{-2\pi i \mathbf{k} \mathbf{f}_j}$$

for each of the nonequispaced frequency nodes  $\mathbf{f}_j \in \mathbb{T}^d$ .

We can write the NUDDFT in matrix notation as:

$$\hat{\mathbf{g}} = \mathbf{A} \mathbf{g}$$

where we have:

$$\hat{\mathbf{g}} = (\hat{g}(\mathbf{f}_0), \hat{g}(\mathbf{f}_1), \dots, \hat{g}(\mathbf{f}_{d-1}), ) \quad \mathbf{A} = (e^{-2\pi i \mathbf{k} \mathbf{f}_j})_{j=0, \dots, M-1; \mathbf{k} \in I_N} \quad \mathbf{g} = (g_{\mathbf{k}})_{\mathbf{k} \in I_N}$$

Note that, if we let  $P = |I_N|$ , then we see:

$$\text{SIZE}(\hat{\mathbf{g}}) = [M, 1] \quad \text{SIZE}(\mathbf{A}) = [M, P] \quad \text{SIZE}(\mathbf{g}) = [P, 1]$$

The order of the entries of  $A$  depend on the order of the indexed set  $I_N$ , and it is therefore not useful to write  $\mathbf{A}$  in terms of an expression for the  $i_j^{\text{th}}$  component. However, it is imperative that the entries of  $\mathbf{g}$ , which serve as the trigonometric weights (time samples) are properly aligned with the order of the indexed set.

Related to the NUDDFT (as will be discussed later) is the adjoint:

$$\mathbf{h} = \mathbf{A}^\perp \hat{\mathbf{g}} \quad \mathbf{h}_{\mathbf{k}} = \sum_{j=0}^{M-1} g_j e^{2\pi i \mathbf{k} \mathbf{f}_j}$$

where  $\perp$  indicates Hermitian (conjugate transpose).

### 3.2 Special Case: Equispaced Nodes

If  $M = N^d$ , and the nodes  $\mathbf{f}_j$  are equally spaced ( $\mathbf{f}_j = \frac{1}{N}\mathbf{j}, \mathbf{j} \in I_N$ ), then the formulation of the NUFFT reduces to the DFT, and will satisfy:

$$\mathbf{A}\mathbf{A}^\perp = \mathbf{A}^\perp\mathbf{A} = |I_N|\mathbf{I}$$

where  $|I_N|$  represents the dimension of the space of d-variate trigonometric polynomials (total number of sample points used).

## 4 Development of the NUFFT

Construction of the  $\mathbf{A}$  matrix for the NUFFT is not computationally efficient. In much the same way that the FFT is established to reduce the computational complexity of the DFT, we now establish the NUFFT to play an analogous role for the NUFFT.

### 4.1 Basic Formulation in 1-D

For simplified notation, we can explore the NUFFT in a single dimension. The NUFFT algorithm is approximative, and uses the design behind standard FFTs and window functions to achieve an overall computational complexity of  $\mathcal{O}(N \log(N) + \log(\frac{1}{\varepsilon})M)$ , with  $\varepsilon$  being the desired tolerance.

Assuming we have  $N$  sample points, in one dimension our earlier equation reduces to

$$\hat{g}(f) = \sum_{k \in I_N} g_k e^{-2\pi i k f}$$

which we consider at  $j$  arbitrary nodes ( $f_j \in \mathbb{T}, j = 0, \dots, M-1$ ). Our goal in designing the NUFFT is to approximate  $\hat{g}$  by a linear combination of shifted 1-periodic window functions, denoted  $\tilde{\varphi}$ . Assuming we have found the proper coefficients, then, our approximation is given by

$$s_1(f) = \sum_{q \in I_n} \hat{w}_q \tilde{\varphi}\left(f - \frac{q}{n}\right)$$

where the  $\hat{w}_q$  terms are the weights. If we assume that we have oversampled the data by some factor  $\sigma > 1$ , then we know  $n = \sigma N$ , and:

$$s_1(f) = \sum_{q \in I_n} \hat{w}_q \tilde{\varphi}\left(f - \frac{q}{\sigma N}\right)$$

#### 4.1.1 The Window Function

This section will clarify the role of the window function. Consider what the intuitive meaning of  $s_1(f)$  is; we are constructing an approximation of  $\hat{g}$  by weighting a series of window functions

centered at  $f = \frac{q}{n}$ . Therefore, what we are in effect doing is treating the weights as sample values at these locations and using an appropriate interpolation function. In choosing a window function, what we are trying to do is select a function that will give us the most accurate interpolation of the function  $\hat{g}$  from the known values at the sample locations.

If we assume a standard window function  $\varphi : \mathbb{R} \rightarrow \mathbb{R}$ , then its 1-periodic version  $\tilde{\varphi}$  is given by:

$$\tilde{\varphi}(f) = \sum_{r \in \mathbb{Z}} \hat{\varphi}(f + r)$$

Thus the 1-periodic version consists of the sum of all integer shifts of  $\varphi$ . Since  $\tilde{\varphi}$  is a periodic function, it can be expressed in a Fourier Series as

$$\tilde{\varphi}(f) = \sum_{k \in \mathbb{Z}} c_k(\tilde{\varphi}) e^{-2\pi i k f}$$

with Fourier coefficients:

$$c_k(\tilde{\varphi}) = \int_{\mathbb{T}} \tilde{\varphi}(f) e^{2\pi i k f} df = \varphi_k$$

#### 4.1.2 Cut-Off in the Time/Space Domain

Taking the definition of  $s_1(f)$  that we see:

$$s_1(f) = \sum_{q \in I_n} \hat{w}_q \tilde{\varphi}\left(f - \frac{q}{n}\right)$$

Now, using the Fourier Series we developed, consider the following:

$$\tilde{\varphi}\left(f - \frac{q}{n}\right) = \sum_{k \in \mathbb{Z}} c_k(\tilde{\varphi}) e^{-2\pi i k (f - \frac{q}{n})} = \sum_{k \in \mathbb{Z}} c_k(\tilde{\varphi}) e^{-2\pi i k f} e^{\frac{2\pi i k q}{n}}$$

Now, we can write this instead as a double sum:

$$\tilde{\varphi}\left(f - \frac{q}{n}\right) = \sum_{k \in I_n} \sum_{r \in \mathbb{Z}} c_{k+nr}(\tilde{\varphi}) e^{-2\pi i (k+nr) f} e^{\frac{2\pi i (k+nr) q}{n}}$$

This is further simplified as:

$$\tilde{\varphi}\left(f - \frac{q}{n}\right) = \sum_{k \in I_n} \sum_{r \in \mathbb{Z}} c_{k+nr}(\tilde{\varphi}) e^{-2\pi i (k+nr) f} e^{\frac{2\pi i k q}{n}} e^{2\pi i r q} = \sum_{k \in I_n} \sum_{r \in \mathbb{Z}} c_{k+nr}(\tilde{\varphi}) e^{-2\pi i (k+nr) f} e^{\frac{2\pi i k q}{n}}$$

This allows us to write  $s_1(f)$  as:

$$s_1(f) = \sum_{q \in I_n} \hat{w}_q \sum_{k \in I_n} \sum_{r \in \mathbb{Z}} c_{k+nr}(\tilde{\varphi}) e^{-2\pi i (k+nr) f} e^{\frac{2\pi i k q}{n}}$$

If we split this into two sums and change the order of the summations, we have:

$$s_1(f) = \sum_{k \in I_n} \sum_{q \in I_n} \hat{w}_q c_k(\tilde{\varphi}) e^{-2\pi i k f} e^{\frac{2\pi i k q}{n}} + \sum_{r \in \mathbb{Z} \setminus \{0\}} \sum_{k \in I_n} \sum_{q \in I_n} \hat{w}_q c_{k+nr}(\tilde{\varphi}) e^{-2\pi i (k+nr) f} e^{\frac{2\pi i k q}{n}}$$

Now, let us establish a new variable:

$$w_k = \sum_{q \in I_n} \hat{w}_q e^{\frac{2\pi i k q}{n}}$$

Then we have a full frequency representation for  $s_1(f)$  as:

$$s_1(f) = \sum_{k \in I_n} w_k c_k(\tilde{\varphi}) e^{-2\pi i k f} + \sum_{r \in \mathbb{Z} \setminus \{0\}} \sum_{k \in I_n} w_k c_{k+nr}(\tilde{\varphi}) e^{-2\pi i (k+nr) f}$$

While this may seem like an odd construct, it has an intuitive meaning: it is a representation of  $s_1(f)$  as a Fourier expansion.

At this juncture, we arrive at the first stage of approximation. If we assume that  $\hat{g}$  is well-localized in time, then we can truncate the coefficient for the time samples at the outside edges. Specifically, if we assume that  $c_k(\tilde{\varphi})$  is small for  $|k| \geq n - \frac{N}{2}$  we set  $w_k = 0$  for all  $k \in I_n \setminus I_N$ . For  $k \in I_N$ , we want the coefficients for  $s_1(f)$  to be identical to those of  $\hat{g}(f)$ , so we set  $w_k = \frac{g_k}{c_k(\tilde{\varphi})}$  for  $k \in I_N$ . Using this information in our equation for  $w_k$ , we can conclude that the weights for the shifted window functions are given by:

$$\hat{w}_q = \frac{1}{n} \sum_{k \in I_N} w_k e^{-2\pi i \frac{kq}{n}} \quad q \in I_n$$

Notice that this is a standard, uniform FFT of size  $n$ . Therefore, it can be evaluated with similar efficiency to standard FFT techniques.

This process results in an approximation because the truncation of the coefficients  $w_k$  for those  $k \in I_n \setminus I_N$  will cause a cut-off in the time domain that results in signal truncation.

### 4.1.3 Cut-off in Frequency Domain

In addition to our truncated time/space approximation, we would also like for our window function to have compact support in the frequency domain. If we assume that the non-periodic  $\varphi$  is reasonably well localized between  $-\frac{m}{n}$  and  $\frac{m}{n}$  ( $m \ll n$ ,  $m \in \mathbb{N}$ ), then we can approximate it by a function  $\psi$  given by:

$$\hat{\psi}(f) = \hat{\varphi}(f) \hat{\chi}_{[-\frac{m}{n}, \frac{m}{n}]}(f)$$

This assures us that  $\text{SUPP}(\hat{\psi}) = [-\frac{m}{n}, \frac{m}{n}]$ . Using the same technique as before, we can write the one periodic version as:

$$\tilde{\psi}(f) = \sum_{r \in \mathbb{Z}} \hat{\psi}(f+r)$$

If the values of  $n$  and  $m$  are fixed, we can define an index set as a function of  $f_j \in \mathbb{T}$  by:

$$I_{n,m}(f_j) = \{q \in I_n : nf_j - m \leq q \leq nf_j + m\}$$

The reason for establishing this set can be seen by using the function  $\psi$  to approximate our earlier expression for  $s_1$ :

$$s_1(f_j) = \sum_{q \in I_n} \hat{w}_q \tilde{\varphi}\left(f_j - \frac{q}{n}\right) \approx \sum_{q \in I_n} \hat{w}_q \tilde{\varphi}\left(f_j - \frac{q}{n}\right) \hat{\chi}_{[-\frac{m}{n}, \frac{m}{n}]}(f_j - \frac{q}{n})$$

Notice that if  $\left|f_j - \frac{q}{n}\right| > \frac{m}{n}$  for any of the  $q$  terms, its contribution will be 0. Therefore, we can determine that the only values of  $q$  that will contribute non-zero terms to the sum are given by:

$$-\frac{m}{n} \leq f_j - \frac{q}{n} \leq \frac{m}{n}$$

$$nf_j - m \leq q \leq nf_j + m$$

Therefore, rather than summing over  $q \in I_n$ , we can instead write our approximation to  $s_1(f_j)$  as:

$$s_1(f_j) = \sum_{q \in I_{n,m}(f_j)} \hat{w}_q \tilde{\varphi}\left(f_j - \frac{q}{n}\right) \hat{\chi}_{[-\frac{m}{n}, \frac{m}{n}]}\left(f_j - \frac{q}{n}\right) = \sum_{q \in I_{n,m}(f_j)} \hat{w}_q \tilde{\psi}\left(f_j - \frac{q}{n}\right)$$

We can see from our definition of  $I_{n,m}(f_j)$  that for a fixed  $f_j$ , there will be at most  $(2m+1)$  non-zero terms in the sum.

The cut-off we impose in the frequency domain causes further error to our signal via aliasing.

## 4.2 Generalization to Higher Dimensions

The development for the 1-D NUFFT can be generalized to higher dimensions. To do so, we design a  $d$ -variate window function as the product of  $d$  univariate window functions:

$$\hat{\varphi}(\mathbf{f}) = \hat{\varphi}(f_0) \hat{\varphi}(f_1) \cdots \hat{\varphi}(f_{d-1})$$

Since each function in this product is univariate, the Fourier coefficients are given by

$$c_{\mathbf{k}}(\tilde{\varphi}) = c_{k_0}(\tilde{\varphi}_0) c_{k_1}(\tilde{\varphi}_1) \cdots c_{k_{d-1}}(\tilde{\varphi}_{d-1})$$

where we are reminded that  $\mathbf{k}$  is a vector in this multidimensional extension. Using the same general procedure as in 1-D, we begin with

$$s_1(\mathbf{f}) = \sum_{\mathbf{q} \in I_{\mathbf{n}}} \hat{w}_{\mathbf{q}} \tilde{\varphi}(\mathbf{f} - (\mathbf{n}^{-1} \odot \mathbf{q}))$$

where  $\mathbf{n}$  is now a vector indicating the FFT size, and is given by  $\mathbf{n} = \sigma \odot \mathbf{N}$ , where  $\sigma$  is a vector of oversampling factors for each dimension. In higher dimensions, the frequency cut-off approximation is expressed by:

$$w_{\mathbf{k}} = \begin{cases} \frac{g_{\mathbf{k}}}{c_{\mathbf{k}}(\tilde{\varphi})} & \mathbf{k} \in I_{\mathbf{N}} \\ 0 & \mathbf{k} \in I_{\mathbf{n}} \setminus I_{\mathbf{N}} \end{cases}$$

This, in turn, allows for the values  $\hat{w}_{\mathbf{q}}$  to be obtained by a standard multivariate FFT of size  $[n_0, n_1, \dots, n_{d-1}]$  in the  $d$  dimensions:

$$\hat{w}_{\mathbf{q}} = \frac{1}{|I_{\mathbf{n}}|} \sum_{\mathbf{k} \in I_{\mathbf{N}}} w_{\mathbf{k}} e^{-2\pi i \mathbf{k}(\mathbf{n}^{-1} \odot \mathbf{q})} \quad \mathbf{q} \in I_{\mathbf{n}}$$

If we denote Cartesian product of  $[-\frac{m}{n}, \frac{m}{n}]$  in each of the  $d$  dimensions as  $[-\frac{m}{n}, \frac{m}{n}]^d = [-\frac{m}{n}, \frac{m}{n}] \times [-\frac{m}{n}, \frac{m}{n}] \times \cdots \times [-\frac{m}{n}, \frac{m}{n}]$ , then we approximate the window function with a function having compact support by:

$$\hat{\psi}(\mathbf{f}) = \hat{\varphi}(\mathbf{f}) \hat{\chi}_{[-\frac{m}{n}, \frac{m}{n}]^d}(\mathbf{f})$$

Then, if we define the multi-index set in multiple dimensions as

$$I_{\mathbf{n}, \mathbf{m}}(\mathbf{f}_j) = \{\mathbf{q} \in I_{\mathbf{n}} : \mathbf{n} \odot \mathbf{f}_j - m\mathbf{1} \leq \mathbf{q} \leq \mathbf{n} \odot \mathbf{f}_j + m\mathbf{1}\}$$

where  $\mathbf{1}$  represents a vector of length  $d$  having 1 for all of its components. This allows us to write our final approximation  $s$  in multiple dimensions as:

$$s(\mathbf{f}_j) = \sum_{\mathbf{q} \in I_{\mathbf{n}, \mathbf{m}}(\mathbf{f}_j)} \hat{w}_{\mathbf{q}} \tilde{\psi}(\mathbf{f}_j - \mathbf{n}^{-1} \odot \mathbf{q})$$

### 4.3 The NUFFT Algorithm

The NUFFT uses a set of Fourier coefficients obtained at known frequencies to determine an approximate value of the function that generated those coefficients. Now that the mathematics of the NUFFT algorithm have been developed, it can be stated in a few simple steps. The inputs to the algorithm are  $d$ ,  $M \in \mathbb{N}$ ,  $\mathbf{N} \in 2\mathbb{N}^d$  (a  $d$ -length vector of even integers),  $\mathbf{f}_j \in [-\frac{1}{2}, \frac{1}{2}]^d$ ,  $j = 0, \dots, M-1$ , and  $g_{\mathbf{k}} \in \mathbb{C}$ , with  $\mathbf{k} \in I_{\mathbf{N}}$ . Having these inputs, the algorithm is as follows:

1. For  $\mathbf{k} \in I_{\mathbf{N}}$ , compute  $w_{\mathbf{k}} = \frac{g_{\mathbf{k}}}{|I_{\mathbf{N}}| c_{\mathbf{k}}(\tilde{\varphi})}$
2. For  $\mathbf{q} \in I_{\mathbf{n}}$ , compute by use of the  $d$ -variate FFT  $\hat{w}_{\mathbf{q}} = \sum_{\mathbf{k} \in I_{\mathbf{N}}} w_{\mathbf{k}} e^{-2\pi i \mathbf{k}(\mathbf{n}^{-1} \odot \mathbf{q})}$
3. For  $j = 0, \dots, M-1$ , compute  $\hat{g}(\mathbf{f}_j) = \hat{g}_j = \sum_{\mathbf{q} \in I_{\mathbf{n}, \mathbf{m}}(\mathbf{f}_j)} \hat{w}_{\mathbf{q}} \tilde{\psi}(\mathbf{f}_j - \mathbf{n}^{-1} \odot \mathbf{q})$

The output produced by this is an approximation of the values of  $\hat{g}$  at the nodes  $\mathbf{f}_j$  for  $j = 0, \dots, M-1$ , and its complexity is  $\mathcal{O}(|\mathbf{N}| \log(|\mathbf{N}|) + M)$ .

The algorithm can also be expressed in matrix notation as  $\mathbf{A}\mathbf{g} \approx \mathbf{B}\mathbf{F}\mathbf{D}\mathbf{g}$ . In this notation, the matrix  $\mathbf{B}$  is a sparse matrix of dimension  $M \times |I_{\mathbf{n}}|$  holding the window values, and is given by:

$$\mathbf{B} = \left( \tilde{\psi}(\mathbf{f}_j - \mathbf{n}^{-1} \odot \mathbf{q}) \right)_{j=0, \dots, M-1; \mathbf{q} \in I_{\mathbf{n}}}$$

$\mathbf{F}$  plays the role of the Fourier matrix of size  $|I_{\mathbf{n}}| \times |I_{\mathbf{n}}|$ , which performs the FFT described in the development of the NUFFT. The matrix  $\mathbf{D}$  is designed so that the product  $\mathbf{D}\hat{\mathbf{f}}$  holds the truncated sample values that result from the time cut-off approximation, and is given by:

$$\mathbf{D} = \bigotimes_{t=0}^{d-1} \left( \mathbf{0}_t \left| \text{DIAG} \left( \frac{1}{c_{k_t}(\tilde{\varphi}_t)} \right)_{k_t \in I_{N_t}} \right| \mathbf{0}_t \right)^T$$

where  $\mathbf{0}_t$  represents a zero matrix of appropriate size and  $\bigotimes$  represents the Kronecker product ([http://en.wikipedia.org/wiki/Kronecker\\_product](http://en.wikipedia.org/wiki/Kronecker_product)). It should be noted that the entries of  $\mathbf{D}$  are all real, as are the entries of  $\mathbf{B}$ .

## 4.4 The Adjoint NUFFT and Its Algorithm

Recall that in the development of the NUDFT, we mentioned a related matrix to the NUDFT matrix that we referred to as the adjoint NUDFT. This is given by:

$$\mathbf{h} = \mathbf{A}^\perp \hat{\mathbf{g}} \quad \mathbf{h}_{\mathbf{k}} = \sum_{j=0}^{M-1} \hat{g}_j e^{2\pi i \mathbf{k} \mathbf{f}_j}$$

Having developed the NUFFT, we can state in a similar fashion an algorithm for its adjoint. Starting with the matrix notation, the adjoint of the NUFFT will be approximated by:

$$\mathbf{A}^\perp \hat{\mathbf{g}} \approx \mathbf{D}^T \mathbf{F}^\perp \mathbf{B}^T \hat{\mathbf{g}}$$

We define a new multi-index set:

$$I_{\mathbf{n}, \mathbf{m}}^T(\mathbf{q}) = \{j = 0, \dots, M-1 : \mathbf{q} - m\mathbf{1} \leq \mathbf{n} \odot \mathbf{f}_j \leq \mathbf{q} + m\mathbf{1}\}$$

This index set will allow us to state the adjoint NUFFT algorithmically. If we assume that for the adjoint NUFFT, we are given as inputs the dimensions ( $d$ ), the number of frequency points ( $M$ ), the sample length vector ( $\mathbf{N} \in 2\mathbb{N}^d$ ), frequency node locations ( $\mathbf{f}_j \in [-\frac{1}{2}, \frac{1}{2}]^d$ ), and the Fourier coefficient values at the nodes ( $\hat{g}(\mathbf{x}_j) = \hat{g}_j \in \mathbb{C}$ ,  $j = 0, \dots, M_1$ ), then the algorithm reads as:

1. For  $\mathbf{q} \in I_{\mathbf{n}}$ , compute  $\hat{w}_{\mathbf{q}} = \sum_{j \in I_{\mathbf{n}, \mathbf{m}}^T(\mathbf{q})} \hat{g}_j \tilde{\psi}(\mathbf{f}_j - \mathbf{n}^{-1} \odot \mathbf{q})$
2. For  $\mathbf{k} \in I_{\mathbf{N}}$ , use the inverse d-variate FFT to compute  $w_{\mathbf{k}} = \sum_{\mathbf{q} \in I_{\mathbf{n}}} \hat{w}_{\mathbf{q}} e^{2\pi i \mathbf{k}(\mathbf{n}^{-1} \odot \mathbf{q})}$
3. For  $\mathbf{k} \in I_{\mathbf{N}}$ , compute  $h_{\mathbf{k}} = \frac{w_{\mathbf{k}}}{|I_{\mathbf{n}}| c_{\mathbf{k}}(\tilde{\varphi})}$

This will produce our outputs, the approximate sample values in time/space  $h_{\mathbf{k}}$  for  $\mathbf{k} \in I_{\mathbf{N}}$ , and has the same complexity as the NUFFT.

## 4.5 Window Function Choices

There is no theoretically best choice of a window function for the NUFFT or its adjoint. As is the case with most applications calling for a window function, the choice of window will vary by application based on needs for computational efficiency or specific tailoring in the frequency domain. However, certain window functions have become commonly implemented in NUFFT applications:

1. *Dilated Gaussian*

$$\hat{\varphi}(f) = (\pi b)^{-\frac{1}{2}} e^{-\frac{(nf)^2}{b}} \quad \text{with } b = \frac{2\sigma}{2\sigma - 1} \frac{m}{\pi}$$

$$\varphi(k) = \frac{1}{n} e^{-b\left(\frac{\pi k}{n}\right)^2}$$

2. *Cardinal Central B-Splines* (<http://en.wikipedia.org/wiki/B-spline>)

$$\hat{\varphi}(f) = M_{2m}(nf) \quad \text{with } M_{2m}: \text{ order } 2m \text{ centered cardinal B-spline}$$

$$\varphi(k) = \frac{1}{n} \left( \text{SINC} \left( \frac{k\pi}{n} \right) \right)^{2m}$$

3. *Sinc*

$$\hat{\varphi}(f) = \frac{N(2\sigma - 1)}{2m} \left( \text{SINC} \left( \frac{\pi N f (2\sigma - 1)}{2m} \right) \right)^{2m}$$

$$\varphi(k) = M_{2m} \left( \frac{2mk}{(2\sigma - 1)N} \right)$$

4. *Kaiser-Bessel*

$$\hat{\varphi}(f) = \frac{1}{\pi} \begin{cases} \frac{\text{SINH} \left( b\sqrt{m^2 - n^2 f^2} \right)}{\sqrt{m^2 - n^2 f^2}} & |f| \leq \frac{m}{n} \quad b = \pi \left( 2 - \frac{1}{\sigma} \right) \\ \frac{\text{SINH} \left( b\sqrt{n^2 f^2 - m^2} \right)}{\sqrt{n^2 f^2 - m^2}} & \text{otherwise} \end{cases}$$

$$\varphi(k) = \frac{1}{n} \begin{cases} I_0 \left( m\sqrt{b^2 - \left( \frac{2\pi k}{n} \right)^2} \right) & k = -n \left( 1 - \frac{1}{2\sigma} \right), \dots, n \left( 1 - \frac{1}{2\sigma} \right) \\ 0 & \text{otherwise} \end{cases}$$

where  $I_0$  is the modified zero-order Bessel function

([http://en.wikipedia.org/wiki/Bessel\\_function#Modified\\_Bessel\\_functions.:I.CE.B1.2C.K.CE.B1](http://en.wikipedia.org/wiki/Bessel_function#Modified_Bessel_functions.:I.CE.B1.2C.K.CE.B1))

For these particular functions, it has been shown that the approximation error introduced by the NUFFT's use of these window functions decays exponentially with the number  $m$  of summands, though the complexity of the NUFFT increases with  $m$ .

## 5 Inversion

The NUDFT has limited utility if it cannot be adequately inverted. In the task of inverting the NUDFT, we seek to solve the linear system  $\mathbf{A}\mathbf{g} \approx \hat{\mathbf{g}}$ , for  $\mathbf{g}$  given  $\mathbf{A}$  and  $\mathbf{y}$ . The approximation sign is used to acknowledge the fact that if the frequency nodes are not equally spaced, the best we can do is approximate  $\hat{\mathbf{g}}$ .

It is typical for the number of frequency samples  $M$  and the dimension of the polynomial space (the total number of time/space samples) to be unequal, which causes  $\mathbf{A}$  to be rectangular and not square. As has already been discussed, there is no simple formulaic inverse to the matrix  $\mathbf{A}$  when the frequency nodes are unequally spaced, so the problem reduces to finding an appropriate pseudo-inverse  $\mathbf{A}^\dagger$  to apply to the Fourier vector  $\hat{\mathbf{g}}$ .

### 5.1 Over-determined Case: Least Squares

When attempting to perform an inversion, if the number of Fourier coefficients  $M$  is larger than the number of sample locations  $|I_{\mathbf{N}}|$ , the system is over-determined. In this case, the solution is of the form of a weighted least-squares problem, given by:

$$\check{\mathbf{g}} = \underset{\mathbf{g}}{\operatorname{argmin}} \|\hat{\mathbf{g}} - \mathbf{A}\mathbf{g}\|_{\hat{\sigma}}$$

where  $\hat{\sigma}$  is a diagonal frequency weight matrix. The standard notation  $\hat{\mathbf{W}}$  is not used to avoid confusion with the sum weights  $w$  used in the development of the NUFFT. In matrix notation, this problem is equivalent to the weighted normal equation

$$\mathbf{A}^\perp \hat{\sigma} \mathbf{A} \mathbf{g} = \mathbf{A}^\perp \hat{\sigma} \hat{\mathbf{g}}$$

and can be solved with the conjugate gradient method.

### 5.2 Under-determined

When the polynomial degree  $|I_{\mathbf{N}}|$  greater than the number of Fourier coefficients  $M$ , the problem is under-determined, and can be represented as a damped minimization problem:

$$\check{\mathbf{g}} = \underset{\mathbf{g}}{\operatorname{argmin}} \|\mathbf{g}\|_{\sigma^{-1}} \quad \text{subject to} \quad \mathbf{A}\mathbf{g} = \hat{\mathbf{g}}$$

In solving this, we would like a smooth solution in the time/space domain, i.e., the sample values will decay near the edges. This is an interpolation problem that can be expressed as a damped normal equation:

$$\mathbf{A}\sigma\mathbf{A}^\perp\mathbf{g} = \hat{\mathbf{g}}$$

This problem can also be solved with the use of conjugate gradients.

## 6 High-Level Description of Further Generalization

A generalization of the NUDFT is achieved by allowing both the time/space locations and the frequency locations to be non-equispaced. This algorithm is referred to as the NNDFFT in literature, and while its specifics will not be explored in this paper it can be described in a similar manner to the NUDFT. If the NNDFFT is viewed as a series of processes, as the NUDFT was described, the only difference between the NNDFFT and the NUDFT is that an extra interpolation step is involved. If the NNDFFT is fed sample values obtained on a known but non-uniform grid, the first process is to use the samples to interpolate a continuous time function and then sample this function at uniform nodes. The result of this process can then be fed into the same processes as the NUDFT to achieve the output at desired frequency nodes. Again, in the development of the NNDFFT, these steps occur simultaneously or in two distinct steps, rather than as separate processes. It should also be noted that there is an even greater amount of error introduced as two interpolations must be done rather than one. As a result, the inversion of the process is made into a difficult problem.

A twist on the NUDFT as it is described in literature is to allow the time/space nodes to be unevenly spaced and maintain the even spacing of the frequency nodes. The development of the NUDFT in this sense is very similar, and the same method of development can be followed. From the process point of view, rather than interpolate after the standard FFT, the unevenly spaced function samples are used to interpolate the function at evenly spaced samples in time/space, as was described for the NNDFFT.

The development of the NUDFT can be generalized to other non-uniformly spaced transforms, as well. Specific examples of these include the NUFCT/NUFST (non-uniform fast cosine/sine transform), the NUSFFT (non-equispaced sparse fast Fourier transform), the NUFPT (non-uniform fast polynomial transform), and the NUFST (non-uniform fast spherical transform). The development of these transforms is of a similar nature to that of the NUDFT, with differences mainly occurring from divergence in the development of the standard versions of these transforms from the development of the standard FFT.