IET Research Journals

ISSN 1751-8644
doi: 0000000000
www.ietdl.org

Efficient Type 4 and 5 Nonuniform FFT Methods in the One-Dimensional Case

J. Selva

[1] The author is with the Dept. of Physics, Systems Engineering and Signal Theory (DFISTS), University of Alicante, P.O.Box 99, E-03080 Alicante, Spain (e-mail: jesus.selva@ua.es).

Abstract: The so-called non-uniform FFT (NFFT) is a family of algorithms for efficiently computing the Fourier transform of finite-length signals, whenever the time or frequency grid is nonuniformly spaced. Among the five usual NFFT types, types 4 and 5 involve an inversion problem, and this makes them the most intensive computationally. The usual efficient methods for these last types are either based on a finite multipole (FM) or on an iterative conjugate gradient (CG) method. The purpose of this paper is to present efficient methods for these type 4 and 5 NFFTs in the one-dimensional case that just require three NFFTs of types 1 or 2 plus some additional FFTs. Fundamentally, they are based on exploiting the Lagrange formula structure. The proposed methods roughly provide a factor-ten improvement on the FM and CG alternatives in computational burden. The paper includes several numerical examples in double precision, in which the proposed and the Gaussian elimination (GE), CG and FM methods are compared, both in terms of round-off error and computational burden.

1 Introduction

As is well known, the FFT computes the spectrum of a finite-length discrete signal with complexity $O(P \log P)$, where P is the signal's length. However, the FFT requires regular grids in both the time (or spatial) and frequency domains. This constraint is a shortcoming in various applications in which the sampling or evaluation grid is nonuniform. The so-called nonuniform FFT (NFFT) is a family of algorithms for performing the same transform as the FFT but with irregular input or output grids. The NFFT has been developed during the last decades for various applications [1–3]. There exist several surveys and tutorials on this topic [4–7].

Following the classification in [8], there are five basic NFFT types. The first three types have the same complexity order as the FFT and are non-iterative, while types 4 and 5 require either an iterative procedure like the conjugate gradient (CG) method, [9, Sec. 11.3], or a Finite Multipole (FM) method [10]. Type 1 performs the same operation as the FFT but with input samples taken nonuniformly, while type 2 is similar to the inverse FFT but with nonuniform output instants (or positions). Type 3 is a combination of types 1 and 2 and can be viewed as a method to compute the spectrum of a nonuniform delta train at nonuniform frequencies. Finally, types 4 and 5 are the inverses of types 1 and 2 respectively, and are far more expensive than the first three types computationally, given that they involve the inversion of a linear system. For problems of any dimension, these inverse types can be computed through the conjugate gradient (CG) method [11]. However, in the one-dimensional case, a more direct approach consists of exploiting the properties of the Lagrange interpolation formula in order to derive an efficient evaluation procedure. This alternative approach was followed in [10]. In short, in this last reference it was shown that the Lagrange formula can be viewed as a nonuniform convolution which can be implemented through the FM method. However, a drawback of this last method is its expensive initialization, [11, p. 48], [4, p. 27].

The purpose of this paper is to take up again the Lagrange formula approach for the evaluation of the NFFT types 4 and 5 in the one-dimensional case. Fundamentally, the methods proposed for these NFFT types consist of evaluating the Lagrange formula by means of type 1 and type 2 NFFTs and simpler operations. As shown in the numerical examples, the proposed methods roughly provide a factor-ten improvement in computational burden over the state-of-the-art methods, while having the same round-off error performance. This significant improvement is due to the fact that the proposed methods neither require an expensive initialization nor an iterative process, as is the case for state-of-the-art methods like FM and CG. Note that the extension of the proposed methods to multiple dimensions seems difficult, given that there is no general Lagrange formula in several variables [12].

The paper has been organized as follows. In the next section, we shortly recall the five NFFT types. Then, the detailed derivations of the type-5 and type-4 NFFT methods are presented in Secs. 3 and 4 respectively. Fundamentally, in these two sections it is shown how a specific form of the Lagrange formula can be evaluated in order to obtain the type-5 and type-4 NFFTs. [Specifically, the formula is (20).] For clarity, these two sections consist of several sub-sections in which a specific transition in the diagrams in Figs. 1 and 2 is proved. The proposed methods depend on two parameters, the attenuation and oversampling factors, which are analyzed in Sec. 5. Finally, Sec. 6 contains a numerical assessment of the methods' round-off error and complexity.

1.1 Notation

In the rest of the paper, we will use the following notation:

- Definitions will be introduced using the symbol "$\equiv$".
- "$O(\cdot)$" will be the big-O notation.
- P, Q, and R will denote positive integers.
- The notation $\{\cdot\}_p^P$ will represent the vector formed by evaluating the expression inside curly braces for $p = 0, \ldots, P-1$. Thus, for a function $f(x)$, $\{f(p)\}_p^P$ will be the vector $[f(0), f(1), \ldots, f(P-1)]$.
- The operators "DFT" and "IDFT" will respectively denote the DFT and IDFT of a vector. Thus, given a sequence v_q, $\mathrm{DFT}\{v_q\}_q^P$ is the P-length vector whose element at position $p+1$ is

$$\sum_{q=0}^{P-1} v_q e^{-j2\pi pq/P}. \tag{1}$$

- "$\odot$" will represent the element-by-element product of two vectors,

$$\{v_p\}_p^P \odot \{w_p\}_p^P = \{v_p w_p\}_p^P. \tag{2}$$

• The operator "Coef" will extract the coefficient vector of a given trigonometric polynomial,

$$\text{Coef} \sum_{p=0}^{P-1} F_p e^{j2\pi pt} = \{F_p\}_p^P. \tag{3}$$

• "$\|\cdot\|$" will refer to the quadratic norm,

$$\|\{v_p\}_p^P\| = \sqrt{\sum_{p=0}^{P-1} |v_p|^2}. \tag{4}$$

• δ_p will be Kronecker's delta: $\delta_0 = 1$ and $\delta_p = 0$ if $p \neq 0$.
• The arrow "$\to$" will denote a replacement in a given expression.

2 Basic NFFT types

The NFFT methods perform the same operations as the FFT or IFFT but allow for nonuniform sampling or evaluation grids. Basically, there exist five NFFT types, that depend on which grid is nonuniform. In the next sub-sections, we briefly define these methods for the one-dimensional case following the classification in [8].

2.1 *Type-1 NFFT*

The type-1 NFFT transforms a nonuniform into a uniform grid, and can be viewed as a method to sample regularly the spectrum of a nonuniform delta train like

$$\alpha(t) \equiv \sum_{p=1}^{Q} a_p \delta(t - t_p), \tag{5}$$

where $\{a_{p+1}\}_p^Q$ and $\{t_{p+1}\}_p^Q$ are a complex and real vector respectively, with $0 \leq t_p < 1$. More precisely, if $A(f)$ denotes the spectrum of $\alpha(t)$,

$$A(f) \equiv \sum_{p=1}^{Q} a_p e^{-j2\pi f t_p}, \tag{6}$$

the type-1 NFFT computes $\{A(p)\}_p^P$ from $\{t_{p+1}\}_p^Q$ and $\{a_{p+1}\}_p^Q$ with complexity $O(P \log P + Q)$. Typically, it involves one gridding operation and one FFT, both with oversampling factor two. Fundamentally, the gridding operation consists of replacing the deltas in (5) with band-limited discrete pulses.

A basic application of the type-1 NFFT is the following method to compute nonuniform convolutions.

2.1.1 *Efficient computation of nonuniform convolutions:*

The type-1 NFFT allows one to efficiently evaluate nonuniform convolutions of the form

$$\gamma(t) \equiv \sum_{p=1}^{Q} a_p \lambda(t - t_p) \tag{7}$$

in a regular grid $\{q/P\}_q^P$, where $\lambda(t)$ is a trigonometric polynomial

$$\lambda(t) \equiv \sum_{p=0}^{R-1} \Lambda_p e^{j2\pi pt}, \tag{8}$$

and R is an integer multiple of P, ($R \equiv \eta P$ with integer $\eta \geq 1$). To see the relation between $\gamma(t)$ and the type-1 NFFT, note that $\gamma(t)$ is the convolution of $\lambda(t)$ with the delta train in the type-1 NFFT in (5),

$$\gamma(t) = \lambda(t) * \sum_{p=1}^{Q} a_p \delta(t - t_p) = \lambda(t) * \alpha(t), \tag{9}$$

Thus, we have that $\gamma(t)$ is the polynomial

$$\gamma(t) \equiv \sum_{p=0}^{R-1} \Lambda_p A(p) e^{j2\pi pt}, \tag{10}$$

in which $\{A(p)\}_p^R$ can be obtained by means of one type-1 NFFT with $P \to R$. So, we have from (10) that $\{\gamma(q/P)\}_q^P$ is the output of one inverse DFT,

$$\{\gamma(q/P)\}_q^P = P\,\text{IDFT}\Big\{ \sum_{r=0}^{\eta-1} \Lambda_{Pr+p} A(Pr+p) \Big\}_p^P. \tag{11}$$

The type-1 NFFT is a basic tool in spectral estimation from nonuniform samples [13] and, additionally, has various application fields like synthetic aperture radar (SAR) imaging [1], graphics processing [2], and magnetic resonance imaging (MRI) [14].

2.2 *Type-2 NFFT*

The type-2 NFFT is complementary to the type-1 NFFT in the sense that it converts a uniform into a nonuniform grid. Specifically, given a trigonometric polynomial of the form

$$s(t) \equiv \sum_{p=0}^{P-1} S_p e^{j2\pi pt}, \tag{12}$$

where $\{S_p\}_p^P$ is a complex vector, the type-2 NFFT computes $s(t)$ at Q arbitrary instants $\{t_{p+1}\}_p^Q$ with complexity $O(P \log P + Q)$. Usually, it involves one weighting of the vector $\{S_p\}_p^P$, followed by one inverse FFT, and one final interpolation operation. This last operation consists of a short summation extended to the samples close to the interpolation instants t_{p+1}, and all the processing is performed with oversampling factor two. As (12) shows, its basic use is signal interpolation from spectral samples. It has applications in antenna design [15], computational electromagnetics [16], array processing [17], among other fields.

A basic operation based on this NFFT type is the computation of derivatives, as shown in the sequel.

2.2.1 *Efficient derivative computation:*

Since the derivative of $s(t)$ in (12) has coefficients $j2\pi pS_p$,

$$s'(t) = \sum_{p=9}^{P-1} j2\pi p S_p e^{j2\pi pt} \tag{13}$$

we may compute $s'(t_q)$ through the type-2 NFFT of the sequence $\{j2\pi pS_p\}_p^P$.

2.3 *Type-3 NFFT*

The type-3 NFFT is a combination of the previous two types and is a nonuniform to nonuniform transformation. In short, given the instants $\{t_{p+1}\}_p^Q$ and coefficients $\{a_{p+1}\}_p^Q$ in (5), the type-3 NFFT computes $\{A(f_{r+1})\}_r^R$ for a given set of frequencies $\{f_{r+1}\}_r^R$ with complexity $O(P \log P + Q + R)$. It is applied in heat flow computation and MRI [18].

2.4 *Type-4 NFFT*

The type-4 NFFT is the inverse of the type-1 NFFT assuming $P = Q$, and therefore it consists of inverting the linear system

$$A(p) = \sum_{q=1}^{P} a_q e^{-j2\pi p t_q},\ p = 0, \ldots, P-1, \tag{14}$$

with unknowns $\{a_{q+1}\}_q^P$. The efficient computation of this NFFT has been addressed in [8, 10] using two different methods. In [8], the method is based on the a CG algorithm in which each iteration is implemented through the FFT. And in [10], the method consists of writing the Lagrange interpolation formula in terms of nonuniform convolutions, which are then computed through the Fast Multipole (FM) method. The CG method turns out to be the more efficient due to the computational cost of the FM method initialization [11, p. 48], [4, p. 27].. This NFFT type as well as type 5 in the next sub-section appear in computational electromagnetics [3], MRI [14], and spectral estimation [13], among other fields.

2.5 *Type-5 NFFT*

The type-5 NFFT is the inverse of the type-2 NFFT for $P = Q$, i.e, it computes the coefficients $\{S_p\}_p^P$ from the samples $\{s(t_{q+1})\}_q^P$ by inverting the linear system

$$s(t_q) = \sum_{p=0}^{P-1} S_p e^{j2\pi p t_q}. \tag{15}$$

The methods available for this type are fundamentally same as for the type-4 NFFT (CG and FM methods), given that this last linear system is the dual (Hermitian) of that in (14).

3 Computation of the type-5 NFFT

In this section, we present the detailed derivation of the type-5 NFFT method, split in several sub-sections corresponding to each of the transitions in Fig. 1. In each sub-section the corresponding transition is proved, starting with the last one and proceeding backward. Fig. 1 fully specifies the type-5 NFFT computation, given that its inputs are the sampling instants $\{t_{p+1}\}_p^P$ and sample values $\{s(t_{p+1})\}_p^P$ (framed vectors), and its output is the desired coefficients vector $\{S_p\}_p^P$. This figure specifies a method to evaluate the Lagrange formula in (20) simultaneously at all the instants in a regular grid, where ja is an imaginary shift of the t variable, $a > 0$. The function of this last shift is to avoid the poles in two nonuniform convolutions and, at the same time, make the approximation of the corresponding convolution kernels by band-limited functions possible.

3.1 *Coefficients S_p from samples $s(q/P + ja)$*

The final output $\{S_p\}_p^P$ can be computed from the sequence $\{s(q/P + ja)\}_q^P$ through one DFT. This is so because $s(t + ja)$ is just the output of passing $s(t)$ through an invertible filter, as the following equation reveals

$$s(t + ja) = \sum_{p=0}^{P-1} S_p e^{-2\pi p a} e^{j2\pi p t}. \tag{16}$$

The coefficients S_p are given by

$$\{S_p\}_p^P = \frac{1}{P}\{e^{2\pi p a}\}_p^P \odot \mathrm{DFT}\{s(q/P + ja)\}_q^P, \tag{17}$$

and the problem comes down to computing $\{s(q/P + ja)\}_q^P$.

$$\begin{tikzcd}
& \boxed{\{t_{p+1}\}_p^P} \arrow[d, "\text{type-1 NFFT, (29)}"] \\
& \Big\{\sum_{p=1}^{P} \log(1 - e^{j2\pi(q/P - t_p + ja)})\Big\}_q^P \arrow[d, "(25)"] \\
& \{L(e^{j2\pi(q/P+ja)})\}_q^P \arrow[d, "\text{(33), type-2 NFFT}"] \arrow[ddd, bend right=80, "(20)"'] \\
\boxed{\{s(t_{p+1})\}_p^P} \arrow[dr] & \{L'(e^{j2\pi t_p})\}_p^P \arrow[d, "\text{(41), type-1 NFFT}"] \\
& \Big\{\sum_{p=1}^{P} \frac{s(t_p) h(-Pt_p + Pja)}{L'(e^{j2\pi t_p}) e^{j2\pi t_p}} h_1(q/P - t_p)\Big\}_q^P \arrow[d, "(20)"] \\
& \{s(q/P + ja)\}_q^P \arrow[d, "(17)"] \\
& \{S_p\}_p^P
\end{tikzcd}$$

Fig. 1: Flow diagram of type-5 NFFT method.

3.2 *Sequence $s(q/P + ja)$ from Lagrange formula factors*

In order to compute $\{s(q/P + ja)\}_q^P$, consider the Lagrange formula

$$s(t) = \sum_{p=1}^{P} s(t_p) \frac{L(e^{j2\pi t})}{L'(e^{j2\pi t_p})(e^{j2\pi t} - e^{j2\pi t_p})}, \tag{18}$$

where

$$L(e^{j2\pi t}) \equiv \prod_{p=1}^{P} (e^{j2\pi t} - e^{j2\pi t_p}). \tag{19}$$

(18) for $t = q/P + ja$ can be written in the following way,

$$\begin{aligned} s(q/P + ja) &= L(e^{j2\pi(q/P+ja)}) \\ &\cdot \sum_{p=1}^{P} \frac{s(t_p)}{L'(e^{j2\pi t_p})(e^{j2\pi(q/P+ja)} - e^{j2\pi t_p})}. \end{aligned} \tag{20}$$

Note that if we replace q/P by the t variable in this expression, then it is the product of two terms and none of them has any poles or zeros for real t due to the shift ja. Thus, it is possible to compute $\{s(q/P + ja)\}_q^P$ by multiplying element-wise the sequences

$$\{L(e^{j2\pi(q/P+ja)})\}_q^P \tag{21}$$

and

$$\Big\{\sum_{p=1}^{P} \frac{s(t_p)}{L'(e^{j2\pi t_p})(e^{j2\pi(q/P+ja)} - e^{j2\pi t_p})}\Big\}_q^P. \tag{22}$$

In the next sub-section, we show how to compute (21). Then, in Sec. 3.5 we address the computation of the samples $\{L'(e^{j2\pi t_p})\}_p^P$ appearing in (22), and in sub-section 3.6 the computation of the whole sequence (22).

3.3 *Kernel samples* $L(e^{j2\pi(q/P+ja)})$ *from a nonuniform convolution.*

Let us analyze the kernel $L(e^{j2\pi(t+ja)})$ in order to compute (21). The logarithm of this kernel is the following,

$$\begin{aligned}\log L(e^{j2\pi(t+ja)}) &= \sum_{p=1}^{P} \log(e^{j2\pi(t+ja)} - e^{j2\pi t_p}) \\ &= jP\pi + j2\pi \sum_{p=1}^{P} t_p + \sum_{p=1}^{P} \log(1 - e^{j2\pi(t-t_p+ja)}).\end{aligned} \tag{23}$$

Let $v(t)$ denote the summation in this last expression,

$$v(t) \equiv \sum_{p=1}^{P} \log(1 - e^{j2\pi(t-t_p+ja)}). \tag{24}$$

From this definition and (23), we have that the kernel samples can be computed from the sequence $\{v(q/P)\}_q^P$ through the equation

$$L(e^{j2\pi(q/P+ja)}) = \exp\Big(jP\pi + j2\pi \sum_{p=1}^{P} t_p + v(q/P)\Big). \tag{25}$$

3.4 *Computation of* $v(q/P)$.

Let us analyze the definition of $v(t)$ in (24) in order to compute $\{v(q/P)\}_q^P$. The summation in that definition can be written as the convolution of a delta train with the kernel $\log(1 - e^{j2\pi(t+ja)})$,

$$\begin{aligned} v(t) &= \log(1 - e^{j2\pi(t+ja)}) * \sum_{p=1}^{P} \delta(t - t_p) \\ &= \log(1 - e^{j2\pi(t+ja)}) * z(t), \end{aligned} \tag{26}$$

where

$$z(t) \equiv \sum_{p=1}^{P} \delta(t - t_p). \tag{27}$$

Besides, the shift ja introduces an exponential decay in the Fourier series coefficients of $\log(1 - e^{j2\pi(t+ja)})$, and this fact allows us to truncate this last series at a sufficiently large index R. Specifically, we have

$$\begin{aligned}\log(1 - e^{j2\pi(t+ja)}) &= -\sum_{p=1}^{\infty} \frac{1}{p} e^{-2\pi pa} e^{j2\pi pt} \\ &\approx -\sum_{p=1}^{R-1} \frac{1}{p} e^{-2\pi pa} e^{j2\pi pt}.\end{aligned} \tag{28}$$

Therefore, in (26) we may replace $\log(1 - e^{j2\pi(t+ja)})$ with its truncated Fourier series, obtaining

$$v(t) \approx g(t) * z(t), \tag{29}$$

where

$$g(t) \equiv -\sum_{p=1}^{R-1} \frac{1}{p} e^{-2\pi pa} e^{j2\pi pt}, \tag{30}$$

and R is the index of the first neglected coefficient. For simplicity, we take R equal to an integer multiple of P, $R = \eta P$, (integer $\eta \geq 1$). (29) is a nonuniform convolution of the form in (7). Thus, we may compute $\{v(q/P)\}_q^P$ using the procedure already described for (7) with $\lambda(t) \to g(t)$, $a_p \to 1$, and $R \to \eta P$.

3.5 *Derivative samples* $L'(e^{j2\pi t_p})$ *from kernel samples* $L(e^{j2\pi(q/P+ja)})$.

Let $\{L_p\}_p^{P+1}$ denote the set of coefficients of $L(e^{j2\pi t})$ and note the following straight-forward equations

$$\text{Coef } L'(e^{j2\pi t}) = \{(p+1)L_{p+1}\}_p^P \tag{31}$$

$$\text{Coef } L(e^{j2\pi(t+ja)}) = \{e^{-2\pi pa} L_p\}_p^{P+1}. \tag{32}$$

The last equation implies that the DFT of $\{L(e^{j2\pi(q/P+ja)})\}_q^P$ gives the coefficients $\{L_p\}_p^P$ with a weighting. More precisely, we have to take the DFT of $\{L(e^{j2\pi(q/P+ja)})\}_q^P$, remove the aliasing at $p = 0$, and compensate the factor $e^{-2\pi pa}$ in (32). The result is the following equation

$$\begin{aligned}\{L_p\}_p^P = \frac{1}{P}\Big(&\text{DFT}\{L(e^{j2\pi(q/P+ja)})\}_q^P \\ &- Pe^{-2\pi Pa}\{\delta_p\}_p^P\Big) \odot \{e^{2\pi pa}\}_p^P.\end{aligned} \tag{33}$$

Once the coefficients $\{L_p\}_p^P$ are known, and noting that $L_P = 1$, we may obtain $\{L'(e^{j2\pi t_p})\}_p^P$ by applying a type-2 NFFT to $\{(p+1)L_{p+1}\}_p^P$, due to (31).

3.6 *Summation in (22) assuming known samples* $L'(e^{j2\pi t_p})$

In order to compute the summation in (22), let us first define the kernel

$$h(t) \equiv \frac{1}{e^{j2\pi t} - 1}, \tag{34}$$

and re-write the summation in that equation in the following way

$$\begin{aligned}&\sum_{p=1}^{P} \frac{s(t_p)}{L'(e^{j2\pi t_p})e^{j2\pi t_p}} h(q/P - t_p + ja) \\ &= \Big[h(t+ja) * \sum_{p=1}^{P} \frac{s(t_p)}{L'(e^{j2\pi t_p})e^{j2\pi t_p}} \delta(t - t_p)\Big]_{t=q/P}.\end{aligned} \tag{35}$$

Note that this expression resembles that in Sec. 3.4, Eq. (26), for the computation of the samples $\{v(q/P)\}_q^P$. In it, we have the convolution of a signal $h(t + ja)$ with a nonuniform delta train. Besides, $h(t + ja)$ has infinite bandwidth, as can be readily seen in its Fourier series

$$h(t + ja) = -\sum_{r=0}^{\infty} e^{-2\pi ra} e^{j2\pi rt}, \tag{36}$$

and the delta train coefficients are known, given that $\{L'(e^{j2\pi t_p})\}_p^P$ has been pre-computed in Sec. 3.5. However, (35) is a simpler case because $h(t + ja)$ can be replaced by a band-limited kernel, and this allows us to employ a type-1 NFFT without any truncation. To see this point, let us insert a term $h(q - Pt_p + Pja)$ in the summand in (35) and operate as follows:

$$\begin{aligned}&\frac{s(t_p)}{L'(e^{j2\pi t_p})e^{j2\pi t_p}} h(q/P - t_p + ja) \\ &= \frac{s(t_p)h(q - Pt_p + Pja)}{L'(e^{j2\pi t_p})e^{j2\pi t_p}} \cdot \frac{h(q/P - t_p + ja)}{h(q - Pt_p + Pja)}.\end{aligned} \tag{37}$$

Note that in the first fraction we may simplify

$$h(q - Pt_p + Pja) = h(-Pt_p + Pja), \tag{38}$$

and the second fraction is equal to $h_1(q/P - t_p)$, where $h_1(t)$ is a new band-limited pulse, defined by

$$\begin{aligned} h_1(t) \equiv \frac{h(t+ja)}{h(P(t+ja))} &= \frac{e^{j2\pi P(t+ja)} - 1}{e^{j2\pi(t+ja)} - 1} \\ &= \sum_{p=0}^{P-1} e^{-2\pi pa} e^{j2\pi pt}. \end{aligned} \tag{39}$$

So, we have the following formula for the vector in (22)

$$\begin{aligned} &\Big\{ \sum_{p=1}^{P} \frac{s(t_p)}{L'(e^{j2\pi t_p})(e^{j2\pi(q/P+ja)} - e^{j2\pi t_p})} \Big\}_q^P \\ &= \Big\{ \sum_{p=1}^{P} \frac{s(t_p)h(-Pt_p + Pja)}{L'(e^{j2\pi t_p})e^{j2\pi t_p}} h_1(q/P - t_p) \Big\}_q^P. \end{aligned} \tag{40}$$

This is a non-uniform convolution that can be computed through a type-1 NFFT, if we identify in (7)

$$\begin{array}{rcl} R & \to & P \\ a_p & \to & \dfrac{s(t_p)h(-Pt_p + Pja)}{L'(e^{j2\pi t_p})e^{j2\pi t_p}} \\ \lambda(t) & \to & h_1(t). \end{array} \tag{41}$$

4 Computation of the type-4 NFFT

In the type-4 NFFT, we set $Q = P$ in (5) and (6) and the objective is to compute $\{a_{q+1}\}_q^P$ given $\{A(p)\}_p^P$. In order to derive the type-4 NFFT method, note first that there is a convolution similar to (7) in the derivation of the type-5 NFFT, specifically, in the computation of the sequence

$$\Big\{ \sum_{p=1}^{P} \frac{s(t_p)h(-Pt_p + Pja)}{L'(e^{j2\pi t_p})e^{j2\pi t_p}} h_1(q/P - t_p) \Big\}_q^P; \tag{42}$$

(see Fig. 1). Let us relate this sequence with the known spectral samples $\{A(p)\}_p^P$. For this, consider the unique signal $s(t)$ of the form in (12) such that

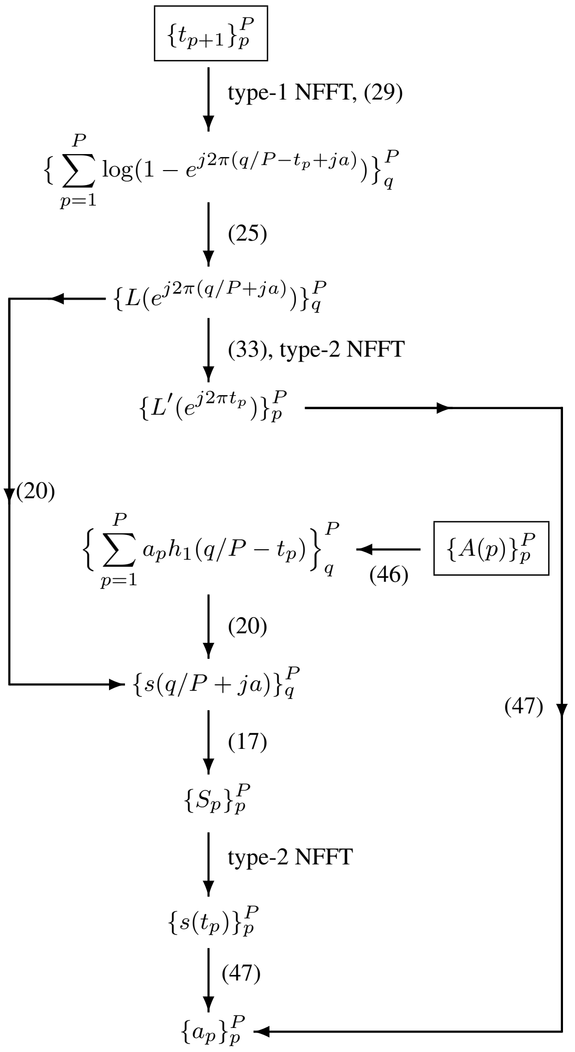
$$\{s(t_{p+1})\}_p^P = \Big\{ \frac{a_{p+1}L'(e^{j2\pi t_{p+1}})e^{j2\pi t_{p+1}}}{h(-Pt_{p+1} + Pja)} \Big\}_p^P, \tag{43}$$

where $\{a_{p+1}\}_p^P$ is the unknown sequence. For this signal, (42) takes the form

$$\Big\{ \sum_{p=1}^{P} a_p h_1(q/P - t_p) \Big\}_q^P. \tag{44}$$

Let us apply the DFT to this sequence. Recalling (39), we have

$$\begin{aligned} &\mathrm{DFT}\Big\{ \sum_{p=1}^{P} a_p h_1(q/P - t_p) \Big\}_q^P \\ &\quad = \sum_{p=1}^{P} a_p \mathrm{DFT}\{h_1(q/P - t_p)\}_q^P \\ &\quad = \sum_{p=1}^{P} a_p \{P e^{-2\pi ra} e^{-j2\pi t_p r}\}_r^P \\ &\quad = P\Big\{ e^{-2\pi ra} \sum_{p=1}^{P} a_p e^{-j2\pi t_p r} \Big\}_r^P \\ &= P\{e^{-2\pi ra} A(r)\}_r^P = P\{e^{-2\pi ra}\}_r^P \odot \{A(r)\}_r^P. \end{aligned} \tag{45}$$



$$\begin{tikzcd} & \boxed{\{t_{p+1}\}_p^P} \arrow[d, "\text{type-1 NFFT, (29)}"] & \\ & \{ \sum_{p=1}^{P} \log(1 - e^{j2\pi(q/P - t_p + ja)}) \}_q^P \arrow[d, "(25)"] & \\ & \{L(e^{j2\pi(q/P+ja)})\}_q^P \arrow[d, "\text{(33), type-2 NFFT}"] \arrow[ddd, bend right=60, "(20)"'] & \\ & \{L'(e^{j2\pi t_p})\}_p^P \arrow[ddddd, bend left=60, "(47)"] & \\ & \Big\{ \sum_{p=1}^{P} a_p h_1(q/P - t_p) \Big\}_q^P \arrow[d, "(20)"] & \boxed{\{A(p)\}_p^P} \arrow[l, "(46)"] \\ & \{s(q/P + ja)\}_q^P \arrow[d, "(17)"] & \\ & \{S_p\}_p^P \arrow[d, "\text{type-2 NFFT}"] & \\ & \{s(t_p)\}_p^P \arrow[d, "(47)"] & \\ & \{a_p\}_p^P & \end{tikzcd}$$

Fig. 2: Flow diagram of type-4 NFFT method.

Thus, applying the IDFT, we have

$$\begin{aligned} &\Big\{ \sum_{p=1}^{P} a_p h_1(q/P - t_p) \Big\}_q^P \\ &\qquad = P\,\mathrm{IDFT}\Big(\{e^{-2\pi ra}\}_r^P \odot \{A(r)\}_r^P \Big). \end{aligned} \tag{46}$$

Using this equation, we may first compute (44) from $\{A(r)\}_r^P$ without actually knowing $\{a_{p+1}\}_p^P$. Then, we may follow the diagram in Fig. 1, successively computing $\{s(q/P + ja)\}_q^P$ and $\{S_p\}_p^P$. Once $\{S_p\}_p^P$ is available, we may compute $\{s(t_{p+1})\}_p^P$ through a type-2 NFFT. And finally, we may obtain $\{a_{p+1}\}_p^P$ by inverting (43),

$$\{a_{p+1}\}_p^P = \Big\{ \frac{s(t_{p+1})h(-Pt_{p+1} + Pja)}{L'(e^{j2\pi t_{p+1}})e^{j2\pi t_{p+1}}} \Big\}_p^P. \tag{47}$$

Fig 2 shows the flow diagram of this type-4 NFFT method.

5 Selection of a and η and refined methods

The oversampling factor η in (30) was selected as an integer multiple of P. Note that this constraint is not a limitation in practice, given that the DFTs in the proposed methods work in the most efficient way if their size is a multiple of P. This is so because both methods operate on the regular time-domain grid $\{p/P + ja\}_p^P$.

Regarding the final accuracy, it is clear that the only error source is the truncation of (28) assuming infinite working precision. However, for finite precision an incorrect selection of either a or η may completely spoil the final result. We can see this point by analyzing the computation of $\{v(q/P)\}_q^P$ in Sec. 3.4. On the one hand, the truncation of (28) requires a negligible ratio between the first and last summand of the truncated series. Thus, if μ denotes this last ratio, defined by

$$\mu \equiv \frac{e^{-2\pi(\eta P-1)a}}{\eta P-1}, \tag{48}$$

then a and η should selected to ensure μ is close to the working precision. So, we may see from (48) that, in rough terms, the product $aP\eta$ must be sufficiently large, and this can be achieved by either increasing a or η. However, an increase in a produces a strong attenuation in the computation of the coefficients $\{L_p\}_p^P$ in (33) due to the vector $\{e^{2\pi pa}\}_p^P$. And an increase in η seems suitable for truncating (28) while reducing the damping effect in (33), but there is a corresponding increase in the computational burden, because the computation of $\{v(q/P)\}_q^P$ involves a type-1 NFFT of size ηP.

A simple way to overcome this situation consists of selecting a and μ that produce an inaccurate result, and then applying the method twice, first to the input sequences $\{t_{p+1}\}_p^P, \{a_{p+1}\}_p^P$, and then to $\{t_{p+1}\}_p^P$ and the residual error, which can be computed through a type-1 or type-2 NFFT. More precisely, suppose we require to compute the type-4 NFFT, but the method available produces an error $\{\Phi_{0,p}\}_p^P$,

$$\{t_{p+1}\}_p^P, \{a_{p+1}\}_p^P \xrightarrow{\text{type-4 method}} \{A(p)+\Phi_{0,p}\}_p^P. \tag{49}$$

Also, suppose that the accuracy of this method is $\epsilon < 1$ in the sense that

$$\|\{\Phi_{0,p}\}_p^P\| < \epsilon \, \|\{A(p)\}_p^P\| \tag{50}$$

for any possible input sequence $\{a_{p+1}\}_p^P$. Then, we may obtain a more accurate approximation of $\{A(p)\}_p^P$ in the following steps:

1. Compute the type-1 NFFT of the right-hand side of (49),

$$\{t_{p+1}\}_p^P, \{A(p)+\Phi_{0,p}\}_p^P \xrightarrow{\text{type-1 NFFT}} \{a_{p+1}+\phi_{0,p+1}\}_p^P,$$

where $\{\phi_{0,p+1}\}_p^P$ is the type-1 NFFT of $\{t_{p+1}\}_p^P$ and $\{\Phi_{0,p}\}_p^P$.
2. Subtract $\{a_{p+1}\}_p^P$ to the last output to obtain $\{\phi_{0,p+1}\}_p^P$.
3. Apply the type-4 method to $\{\phi_{0,p+1}\}_p^P$,

$$\{\phi_{0,p+1}\}_p^P \xrightarrow{\text{type-4 method}} \{\Phi_{0,p}+\Phi_{1,p}\}_p^P$$

where $\{\Phi_{1,p}\}_p^P$ is a new residual error. Now we have

$$\|\{\Phi_{1,p}\}_p^P\| < \epsilon^2 \, \|\{A(p)\}_p^P\|. \tag{51}$$

4. Subtract the last sequence from the output of (50). The result is $\{A(p)-\Phi_{1,p}\}_p^P$ and we have doubled the accuracy due to (51).

A similar refinement can be applied to the type-5 NFFT.

In practice, there are two basic setups for the proposed methods, as can be inferred from the numerical examples in the next section:

1. NFFT method with $\eta = 1$ or 2: These choices yield single precision roughly.
2. R-NFFT method with $\eta = 1$: It yields double precision.

6 Numerical examples

In this section, we evaluate the proposed NFFT methods in terms of round-off error and computational burden using double precision.

Operation	Flops
Real sum	1
Complex sum	2
Real multiplication	1
Complex multiplication	6
Sine, cosine	4
Complex exponential, log.	7
Tangent, cotangent	8
Size-N FFT, IFFT	$5N\log_2 N$

Table 1 Flop counts for basic operations.

We only present results for the type-4 NFFT, though they also represent the corresponding performance of the type-5 NFFT. Actually, the figures that follow and those for the type-5 NFFT are identical and, therefore, it is redundant to repeat them in the present paper. This identical performance is due to the relation between the linear systems in (14) and (15. As can be readily inferred from these last equations, the linear system solved by the types 4 and 5 NFFT form a dual pair, i.e, if we take the Hermitian of the type-4 linear system matrix we obtain the corresponding type-5 matrix. A consequence of this duality is that methods like Gaussian elimination and conjugate gradient have identical round-off error performance and computational burden for both types. And this is also true for the NFFT methods proposed in this paper. Specifically, as can be easily deduced from Figs. 1 and 2, the type-4 and type-5 evaluation procedures are the same except for a small modification that involves no change in either the round-off error performance or computational burden.

We have evaluated the performance of the type-4 (and type-5) NFFT method in the following setup:

• *Monte Carlo trials.* The figures have been generated from just 10 Monte Carlo trial, given that the large values of P produce low-variance round-off error estimates. In these trials, the sampling instants $\{t_{p+1}\}_p^P$ where obtained by shifting the elements of a regular grid with spacing $1/P$. These shifts were independent and had uniform distribution in the interval $[0, 0.6/P]$. The amplitudes $\{a_{p+1}\}_p^P$ were independent complex Gaussian samples of zero mean and variance one.
• *Numerical methods.* We have used the following methods:
◦ GE: Computation based on inverting the associated linear system through Gaussian elimination, [9, Ch. 3].
◦ CG: The same inversion but using the conjugate gradient method [11, p. 73].
◦ FM: Method in [10] based on evaluating the Lagrange formula. In it, one nonuniform convolution is computed through the FM method and two additional convolutions are directly evaluated.
◦ NFFT: Method proposed in this paper in Sec. 4.
◦ R-NFFT: Previous method with refinement (Sec. 5).
• *Computational burden.* We have measured the computational burden in floating-point operations (flops), following the counts in Table 1 for basic operations.
• *Round-off error measure.* Given a true vector $\{A(p)\}_p^P$ and an interpolated vector $\{\tilde{A}_p\}_p^P$, the error measure has been

$$\frac{\|\{A(p)-\tilde{A}_p\}_p^P\|}{\|\{A(p)\}_p^P\|}. \tag{52}$$

Fig. 3 shows the round-off error of the NFFT method versus the attenuation μ in (48) for several oversampling factors η and $P = 1024$. This figure also includes the round-off error of the GE, CG, and FM methods as benchmarks. Note that with $\eta = 1$, the achievable round-off error is around -130 dB, which is a value significantly larger that the GE, CG benchmarks. However, this error decreases with η. With $\eta = 6$ it is around -220 dB and can be reduced by increasing η to a value only 10 dB above the previous benchmarks. As Fig. 4 shows, this decrease is obtained at the expense of a higher computational burden but, by far, the

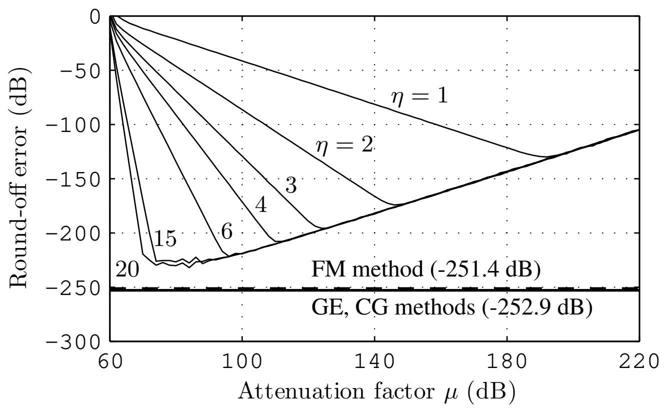


Fig. 3: Round-off error versus attenuation factor (μ) for several oversampling factors (η) for the type-4 NFFT method.

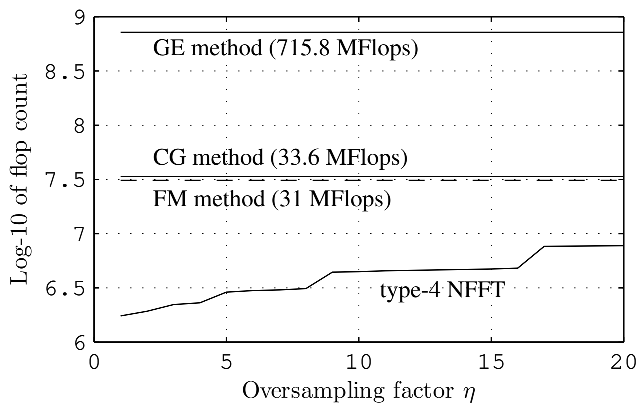


Fig. 4: Computational burden of type-4 NFFT method, measured in floating point operations (flops), versus oversampling factor η. The vertical axis shows the base-10 logarithm of the flop count. For instance, for the GE method the y abscissa is $\log_{10}(751.8 \cdot 10^6) = 8.8761$.

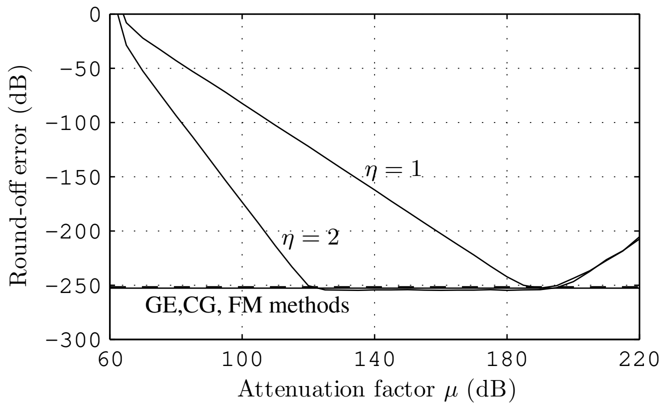


Fig. 5: Round-off error versus attenuation factor (μ) for two oversampling factors (η) for the iterated type-4 NFFT method.

type-4 NFFT is the cheapest computationally. Actually, its computational burden is more than factor 10 smaller that the CG method's complexity for $\eta = 6$.

Fig 5 shows the round-off error of the type-4 R-NFFT method versus the attenuation factor μ for $\eta = 1, 2$. Note that without any oversampling ($\eta = 1$), we may select an attenuation μ for which its round-off error is similar to that of the GE, CG or FM methods. Fig. (6) shows the same round-off error but versus P for $\eta = 1$. For each

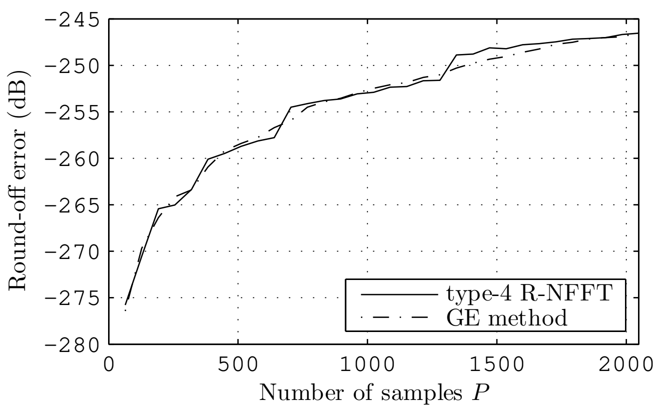


Fig. 6: Round-off error versus number of samples P for the type-4 R-NFFT method.

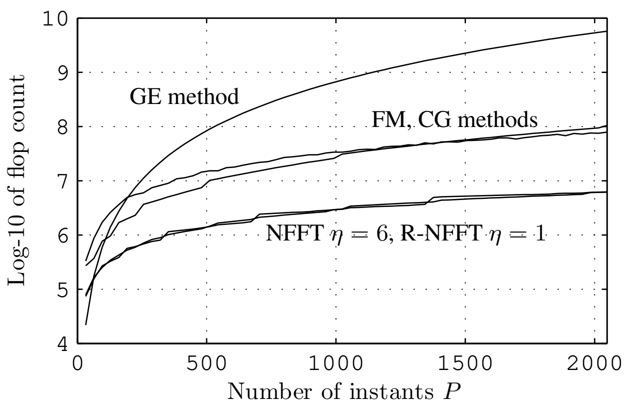


Fig. 7: Flop counts of four methods: GE, CG, FM, type-4 NFFT with $\eta = 6$, and type-4 R-NFFT with $\eta = 1$.

abscissa in this figure, μ has been selected to minimize the round-off error. We can see that the type-4 R-NFFT method reaches the GE benchmark for typical P values.

Finally, Fig. 7 shows the flop count of five methods: CG, GE, FM, type-4 NFFT with $\eta = 6$, and type-4 R-NFFT with $\eta = 1$. Note that the last two methods roughly have the same flop count but, as shown in Fig. 3, the NFFT method is slightly above the GE, CG and FM benchmarks, while the R-NFFT method reaches it (Fig. 6).

7 Conclusions

We have presented non-iterative methods for the type 4 and 5 NFFTs in the one-dimensional case, which are significantly less expensive computationally that the state-of-the-art methods like the conjugate gradient (CG) or finite multipole (FM). The methods are based on expressing the Lagrange formula in terms of nonuniform convolutions that can be efficiently evaluated using the type 1 and 2 NFFTs. The paper contains several numerical examples in which the proposed methods are compared with the CG, FM, and Gaussian elimination (GE) methods in terms of round-off error and computational burden.

8 References

1 Andersson, F., Moses, R., Natterer, F.: 'Fast Fourier methods for synthetic aperture radar imaging', *IEEE Transactions on Aerospace and Electronic Systems*, 2012, **48**, (1), pp. 215–229

2 Kunis, S., Kunis, S.: 'The nonequispaced FFT on graphics processing units', *PAMM*, 2012, **12**, (1), pp. 7–10

3 Zhou, L., Fan, Z.H., Mo, L., Chen, R.S., Wang, D.X. 'A fast inverse NUFFT algorithm for computational electromagnetics'. In: IEEE Antennas and Propagation Society International Symposium. vol. 4B. (, 2005. pp. 160–163

4 Keiner, J., Kunis, S., Potts, D.: 'Using NFFT 3—a software library for various nonequispaced fast Fourier transforms', *ACM Transactions on Mathematical Software*, 2009, **36**, (4), pp. 1–30

5 Potts, D., Steidl, G., Tasche, M. Fast Fourier Transforms for Nonequispaced Data: A tutorial. In: 'Modern sampling theory'. (Birkhauser Boston, 2001. pp. 245–270

6 Benedetto, J.J.: 'Wavelets: mathematics and applications'. vol. 13. (CRC press, 1993)

7 Duijndam, A.J.W., Schonewille, M.A.: 'Nonuniform Fast Fourier Transform', *Geophysics*, 1999, **64**, (2), pp. 539–551

8 Dutt, A., Rokhlin, V.: 'Fast Fourier transforms for nonequispaced data', *SIAM Journal on Scientific computing*, 1993, **14**, (6), pp. 1368–1393

9 Golub, G.H., Loan, C.F.V.: 'Matrix Computations'. 4th ed. (The Johns Hopkins University Press, 2013)

10 Dutt, A., Rokhlin, V.: 'Fast Fourier transforms for nonequispaced data, II', *Applied and Computational Harmonic Analysis*, 1995, **2**, (1), pp. 85–100

11 Kunis, S.: 'Nonequispaced FFT, generalisation and inversion'. (Technisch-Naturwissenschaftlichen Fakultat, Universitat zu Lubeck, 2006)

12 Gasca, M., Sauer, T.: 'Polynomial interpolation in several variables', *Advances in Computational Mathematics*, 2001, **12**, (4), pp. 377–410

13 Babu, P., Stoica, P.: 'Spectral analysis of nonuniformly sampled data–a review', *Digital Signal Processing*, 2010, **20**, (2), pp. 359–378

14 Fessler, J.A., Sutton, B.P.: 'Nonuniform fast Fourier transform using mini-max interpolation', *IEEE Transactions on Signal Processing*, 2003, **51**, pp. 560–574

15 Capozzoli, A., Curcio, C., Liseno, A., Riccardi, A. 'Selecting parameters of type-3 NUFFTs to control accuracy in MoM methods'. In: Antennas and Propagation (EuCAP), 2014 8th European Conference on. (IEEE, 2014. pp. 1157–1161

16 Liu, Q.H., Nguyen, N.: 'An accurate algorithm for nonuniform fast fourier transforms (NUFFT's)', *IEEE Microwave and Guided Wave Letters*, 1998, **8**, (1), pp. 18–20

17 Selva, J.: 'An efficient Newton-type method for the computation of ML estimators in a Uniform Linear Array', *IEEE Transactions on Signal Processing*, 2005, **53**, (6), pp. 2036–2045

18 Lee, J.Y., Greengard, L.: 'The type 3 nonuniform FFT and its applications', *Journal of Computational Physics*, 2005, **206**, (1), pp. 1–5