Accurate Numerical Fourier Transform in d-Dimensions.

Normand Beaudoin* and Steven S. Beauchemin**

The University of Western Ontario, London, Canada N6A 5B7
normand@csd.uwo.ca, beau@csd.uwo.ca

Abstract. The classical method of numerically computing Fourier transforms of digitized functions in one or in d-dimensions is the so-called *Discrete Fourier Transform* (DFT) efficiently implemented as *Fast Fourier Transform* (FFT) algorithms. In many cases, the DFT is not an adequate approximation of the continuous Fourier transform. Because the DFT is periodical, spectrum aliasing may occur. The method presented in this contribution provides accurate approximations of the continuous Fourier transform with similar time complexity. The assumption of signal periodicity is no longer posed and allows to compute numerical Fourier transforms in a broader domain of frequency than the usual half-period of the DFT. The aliasing introduced by periodicity can be reduced to a negligible level even with a relatively low number of sampled data points. In addition, this method yields accurate numerical derivatives of any order and polynomial splines of any odd order with their optimum boundary conditions. The numerical error on results is easily estimated. The method is developed in one and in d-dimensions and numerical examples are presented.

1 Introduction

The ubiquitous Fourier transform and its numerical counterpart, the Discrete Fourier Transform (DFT), in one or many dimensions, are used in many fields, such as mathematics (linear systems, random processes, probability, boundary-value problems), physics (quantum mechanics, optics, acoustics, astronomy), chemistry (spectroscopy, crystallography), and engineering (telecommunications, signal processing, image processing, computer vision, multidimensional signal processing) [1, 2, 3, 4, 5].

Although it is usual to consider the DFT as a mathematical tool with its own properties, it certainly makes sense to conceptualize it as the discrete version of the analytical Fourier transform and as an approximation of the latter [1]. In

* Normand Beaudoin is a post-doctoral fellow at the Department of Computer Science, Middlesex College, University of Western Ontario, London, Canada, N6A 5B7. Phone: (519) 661-2111.

** Steven S. Beauchemin is with the Department of Computer Science, Middlesex College, University of Western Ontario, London, Canada, N6A 5B7. Phone: (519) 661-2111.

this regard, the DFT, usually computed via a fast Fourier transform (FFT) algorithm, must be used with caution since it is not a correct approximation in all cases [6, 7, 8, 9]. First, the DFT is periodical and it is only on one half of a period that it constitutes an approximation of the Fourier transform. Second, the sampling rate of the function to be submitted to the DFT is a critical issue. Without sampling the time[1] function at a sufficiently high rate, a phenomenon known as aliasing may become intolerable and spoil the accuracy of the DFT as an approximation of the Fourier transform. It could be thought that if the Nyquist criterion is fulfilled, everything should come out fine. The Nyquist criterion states that the sampling rate must be at least twice the highest frequency of the initial function [1, 2, 10]. However, in applied science, a function may be defined between 0 and T only. Hence, the highest frequency of such a time-limited function is infinite. Consequently, the DFT produces aliasing[2]. One could argue that, even though the highest frequency is infinite, it is possible to sufficiently increase the number of sampled data points such that the error of the DFT becomes as small as one desires. However, the required number of data points could be huge. As an example, for the function $h(t) = e^{-50t}$, $t \in [0,1]$, the error on $DFT\{h\}$, around $f = 64$, decreases roughly as $N^{-1/3}$. Hence, one must increase N by a factor of 1000 to decrease the error by a factor of 10.

In some cases where the result of the DFT is used qualitatively a high accuracy is not absolutely mandatory. But in some applications, such as in deconvolution where a division is performed in the frequency domain, a slight error in the denominator function, particularly when it is close to zero, can seriously distort the result [11].

However, one may increase the accuracy of the numerical Fourier transform when the number of sampled data points is limited. This can be implemented through the assumption that the function from which the sampled data points are extracted and its derivatives are continuous. The sampling process, performed through the so-called Dirac comb [1], in a sense, isolates each data point and considers them as independent from each other. The function and its derivatives are no longer continuous. By re-establishing the continuity between the sampled data points, a method that yields a highly accurate numerical Fourier transform can be devised.

2 Theory in d-dimension

Let $\boldsymbol{t} = (t_1, t_2 \dots t_d) \in \mathbb{R}^d$ and $\boldsymbol{f} = (f_1, f_2 \dots f_d) \in \mathbb{R}^d$, $d \in \mathbb{N}^*$. $\mathbb{R}$ is the set of real numbers, $\mathbb{N}$ the set of nonnegative integers and $\mathbb{N}^* = \mathbb{N} \backslash \{0\}$. Let us define

[1] Without loss of generality, the reciprocal variables *time* (t) and *frequency* (f) are used throughout this article.

[2] The usual method to avoid aliasing is to filter out the high frequency components thus modifying the original signal.

a d-dimensional Heaviside's function:

$$\boldsymbol{\chi} : \mathbb{R}^d \to \mathbb{R}, \quad \boldsymbol{\chi}(\boldsymbol{t}) = \prod_{i=1}^{d} \chi(t_i) \tag{1}$$

in which $\boldsymbol{\chi}$ and χ are Heaviside's functions in d-dimensions and in one dimension respectively. Let us define two d-dimensional rectangular functions such as:

$$R(\boldsymbol{t}) = \boldsymbol{\chi}\left(\boldsymbol{t} - \boldsymbol{0}^-\right) \boldsymbol{\chi}\left(-\boldsymbol{t} + \boldsymbol{T}^+\right) \qquad \text{and} \qquad S(\boldsymbol{t}) = \boldsymbol{\chi}(\boldsymbol{t})\, \boldsymbol{\chi}(-\boldsymbol{t} + \boldsymbol{T}) \tag{2}$$

with $\boldsymbol{0}^- = (0^-, 0^-, \ldots 0^-)$ and $\boldsymbol{T}^+ = \left(T_1^+, T_2^+, \ldots, T_d^+\right)$, $T_\alpha \in \mathbb{R}$, $T_\alpha > 0$, $\forall \alpha$, and in which:

$$0^- = \lim_{\varepsilon \to 0} (0 - \varepsilon) \ , \qquad T_\alpha^+ = \lim_{\varepsilon \to 0} (T_\alpha + \varepsilon) \ , \qquad \varepsilon \in \mathbb{R} \ , \varepsilon > 0 \ . \tag{3}$$

Let $g : \mathbb{R}^d \to (\mathbb{R} \text{ or } \mathbb{C})$, ($\mathbb{C}$ is the field of complex numbers) be a continuous function that admits directional derivatives of any order in any directions for all $\boldsymbol{t}$ such that $S(\boldsymbol{t}) \neq 0$. We now define the following function:

$$h(\boldsymbol{t}) = R(\boldsymbol{t})\, g(\boldsymbol{t}) \ . \tag{4}$$

This way of defining $h(\boldsymbol{t})$ ascertain that the function is continuous and derivable between 0 and T_α and at 0 and T_α for all values of α.

We adopt the following definition for the Fourier transform:

$$\mathcal{F}\{h(\boldsymbol{t})\} = \int_{\mathbb{R}^d} h(\boldsymbol{t})\, e^{-i2\pi \boldsymbol{f} \cdot \boldsymbol{t}} d\boldsymbol{t} \ . \tag{5}$$

By expanding the inner product and reorganizing the terms, (5) becomes:

$$\mathcal{F}\{h(\boldsymbol{t})\}$$
$$= \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} h(t_1, \ldots, t_d)\, e^{-i2\pi f_1 \cdot t_1} dt_1 \right] e^{-i2\pi f_2 \cdot t_2} dt_2 \cdots e^{-i2\pi f_d \cdot t_d} dt_d \ . \tag{6}$$

It is a known fact, evident from (6), that a d-dimensional Fourier transform of a function can be performed by d successive one dimensional Fourier transforms. Consequently, in the next section we develop the theory in one dimension. In that case, generic non-indexed variables as t, f, T... that stand for any indexed variable of a particular dimension of the d-dimensional space are used.

3 Theory in one dimension

In virtue of the properties of the differentiation of Heaviside's and Dirac-delta functions (δ) [2, 12], the n^{th} derivative of h with respect to t is:

$$h^{(n)}(t) = \chi\left(t - 0^-\right) \chi\left(-t + T^+\right) g^{(n)}(t) + D_n(t) \tag{7}$$

in which $D_n(t)$ is defined as:

$$D_n(t) = \begin{cases} 0 \quad \text{if} \quad n = 0 \\ \sum_{m=0}^{n-1} \left\{ g^{(m)}(0^-)\,\delta^{n-m-1}(t-0^-) - g^{(m)}(T^+)\,\delta^{n-m-1}(t-T^+) \right\} \\ \text{if} \quad n \in \mathbb{N}^* \ . \end{cases} \tag{8}$$

Equation (7) and (8) express the fact that the n^{th} derivative of h with respect to t is the ordinary n^{th} derivative of the function h strictly inside the rectangular box where it is continuous and differentiable, in addition to the n^{th} derivative of h in the regions where it is discontinuous.

According to our definition of the Fourier transform, we have:

$$\mathcal{F}\left\{h^{(n)}(t)\right\} = \int_{-\infty}^{\infty} h^{(n)}(t)\, e^{-i2\pi f t} dt \ . \tag{9}$$

We can expand the integral in (9) into parts to form:

$$\mathcal{F}\left\{h^{(n)}(t)\right\} = \int_{-\infty}^{0} h^{(n)}(t)\, e^{-i2\pi f t} dt + \int_{0}^{T} h^{(n)}(t)\, e^{-i2\pi f t} dt + \int_{T}^{\infty} h^{(n)}(t)\, e^{-i2\pi f t} dt \ . \tag{10}$$

The sum of the first and last integrals of the right hand side of (10) clearly is $\mathcal{F}\{D_n(t)\}$. Hence, (10) becomes:

$$\mathcal{F}\left\{h^{(n)}(t)\right\} = \int_{0}^{T} h^{(n)}(t)\, e^{-i2\pi f t} dt + \mathcal{F}\{D_n(t)\} \ . \tag{11}$$

By separating the interval $[0,T]$ into N equal $\Delta t = T/N$ subintervals, (11) can be rewritten as:

$$\mathcal{F}\left\{h^{(n)}(t)\right\} = \sum_{j=0}^{N-1} \left\{ \int_{j\Delta t}^{(j+1)\Delta t} h^{(n)}(t)\, e^{-i2\pi f t} dt \right\} + \mathcal{F}\{D_n(t)\} \ , \quad j \in \mathbb{N} \ . \tag{12}$$

Since $h^{(n)}$ is continuous and differentiable between 0 and T, it can be approximated for $t \in [j\Delta t, (j+1)\Delta t]$, for each $j \in [0, N-1]$, by a Taylor expansion:

$$h^{(n)}(t) = \sum_{p=0}^{\infty} \frac{h_j^{(p+n)} (t - j\Delta t)^p}{p!} \ , \quad p \in \mathbb{N} \tag{13}$$

where $h_j^{(m)}$is the m^{th} derivative of h at $t = j\Delta t$. Merging (12) and (13) yields:

$$\mathcal{F}\left\{h^{(n)}(t)\right\}$$
$$= \sum_{j=0}^{N-1} \left\{ \int_{j\Delta t}^{(j+1)\Delta t} \left(\sum_{p=0}^{\infty} \frac{h_j^{(p+n)}(t-j\Delta t)^p}{p!} \right) e^{-i2\pi f t} dt \right\} + \mathcal{F}\{D_n(t)\} \ , \quad j \in \mathbb{N} \ . \tag{14}$$

With the substitution $\tau = t - j\Delta t$ and an adequate permutation of the integral and sums on j and p, (14) becomes:

$$\mathcal{F}\left\{h^{(n)}(t)\right\}$$
$$= \sum_{p=0}^{\infty} \left\{ \left(\int_0^{\Delta t} \frac{\tau^p e^{-i2\pi f \tau}}{p!} d\tau \right) \left(\sum_{j=0}^{N-1} h_j^{(p+n)} e^{-i2\pi f j \Delta t} \right) \right\} + \mathcal{F}\{D_n(t)\} \ . \tag{15}$$

To numerically compute the Fourier transform of h, we must evaluate it for some discrete values of f. Let $f = k\Delta f = k/T$, $k \in \mathbb{N}$ be these discrete variables. In addition, let us define H_k as the discrete version of $\mathcal{F}\left\{h^{(n)}(t)\right\}$. The integral in (15) depends only on the variable f (or k) and on the parameters p and Δt and can be evaluated analytically, whether f is continuous or discrete, once and for all, for each value of p as:

$$I_p = \frac{1}{p!} \int_0^{\Delta t} \tau^p e^{-i2\pi f \tau} d\tau \ . \tag{16}$$

Since the integral in the definition of I_p is always finite and, in the context of the Gamma function [13], $p! = \pm\infty$ when p is a negative integer, then $I_p = 0$ for $p < 0$.

The summation on j in (15), when $f = k\Delta f = k/T$, is the discrete Fourier transform of the sequence $h_j^{(p+n)}$, $j \in [0, N-1] \subset \mathbb{N}$ [1]. We denote it as $F_{p+n,k}$. Since $\Delta t = T/N$ and $f = k/T$, we have:

$$F_{p+n,k} = \sum_{j=0}^{N-1} h_j^{(p+n)} e^{-i2\pi \frac{kj}{N}} \ . \tag{17}$$

One should note that although we wrote I_p and $F_{p+n,k}$ instead of $I_p\,(f \text{ or } k)$ and $F_{p+n,k}\,(f \text{ or } k)$, these functions always depend on f or k.

Substituting (16) and (17) in (15), we obtain the following result:

$$\mathcal{F}\left\{h^{(n)}(t)\right\} = \sum_{p=0}^{\infty} I_p F_{p+n,k} + \mathcal{F}\{D_n(t)\} \ . \tag{18}$$

When $n = 0$, (18) becomes:

$$H_k = \sum_{p=0}^{\infty} I_p F_{p,k} \ . \tag{19}$$

Now, integrating by parts the right hand side of (9) yields:

$$\mathcal{F}\left\{h^{(n+1)}\right\} = i2\pi f \mathcal{F}\left\{h^{(n)}\right\} \ . \tag{20}$$

Defining $b_n = i2\pi f \mathcal{F}\left\{D_n\right\} - \mathcal{F}\left\{D_{n+1}\right\}$, combining (18) and (20) and reorganizing the terms yields:

$$-i2\pi f I_0 F_{n,k} + \sum_{p=1}^{\infty} \left(I_{(p-1)} - i2\pi f I_p\right) F_{p+n,k} = b_n \ . \tag{21}$$

With the definition $J_\alpha = I_{\alpha-1} - i2\pi f I_\alpha$, (21) becomes:

$$J_0 F_{n,k} + \sum_{p=1}^{\infty} J_p F_{p+n,k} = b_n \ . \tag{22}$$

Given the definition of g and h, we have $g^{(n)}\left(0^-\right) = g^{(n)}\left(0\right) = h^{(n)}\left(0\right)$ and $g^{(n)}\left(T^+\right) = g^{(n)}\left(T\right) = h^{(n)}\left(T\right)$. Using these facts in addition to the properties of Fourier transforms and those of Dirac delta functions [12], one easily observes that expanding b_n results in the simple following form:

$$b_n = h^{(n)}\left(T\right) e^{-i2\pi f T} - h^{(n)}\left(0\right) \ . \tag{23}$$

In the discrete case, where $f = k/T$, (23) takes the following simple and significant form:

$$b_n = h^{(n)}\left(T\right) - h^{(n)}\left(0\right) = h_N^{(n)} - h_0^{(n)} \ . \tag{24}$$

Up to this point, all equations are rigorously exact since p tends towards infinity. However, in practical situations we introduce an approximation by limiting the range on p. Let us define $\theta \in \mathbb{N}$, the truncating parameter, which, for reasons discussed later, is always chosen as an odd integer. We refer to it as the order of the system.

Let us expand (22) for each value of $n \in [0, \theta - 1] \subset \mathbb{N}$. This generates a system of θ different equations and, for each of these, we let p range from 1 to $\theta - n - 1$. This gives the following system, which is written in matrix form:

$$\begin{bmatrix} J_0 & J_1 & \cdots & J_{\theta-1} \\ 0 & J_0 & \cdots & J_{\theta-2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & J_0 \end{bmatrix} \begin{bmatrix} F_{0,k} \\ F_{1,k} \\ \vdots \\ F_{\theta-1,k} \end{bmatrix} \simeq \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{\theta-1} \end{bmatrix} \tag{25}$$

or, more compactly as:

$$\mathsf{M}_a \mathsf{F}_a \simeq \mathsf{B} \ . \tag{26}$$

Note that the matrix M_a is completely known since each of its terms depends only on f. The general expression for the elements of M_a is:

$$\left(\mathsf{M}_a\right)_{\mu\nu} = I_{\mu-\nu-1} - i2\pi f I_{\nu-\mu} = J_{\nu-\mu} \ . \tag{27}$$

Matrix B is unknown. If it were known, we could evaluate F_a from (26):

$$\mathsf{F}_a \simeq \mathsf{M}_a^{-1}\mathsf{B} \ . \tag{28}$$

However, for $f = \kappa N \Delta f$, $\kappa \in \mathbb{N}$, the solutions would strictly diverge. Indeed, for these particular values of f, $\det(\mathsf{M}_a) = 0$. However, for values of f around $N\Delta f/2$, the approximation (28) is quite accurate. We take advantage of this fact to compute B.

The first element of F_a, which is $(\mathsf{F}_a)_1 = F_{0,k}$, is the DFT of the sequence h_j. It is completely determined for each value of k. It is not the same situation for the other elements of F_a which are still unknown. Furthermore, the elements of matrix M_a^{-1} are given for each value of f. We can then extract the following from (28):

$$F_{0,k} \simeq \left(row_1 \mathsf{M}_a^{-1}\right)\mathsf{B} = \sum_{\mu=1}^{\theta} \left(\mathsf{M}_a^{-1}\right)_{1,\mu} (\mathsf{B})_\mu \ . \tag{29}$$

Let us now define Ω , an interval of θ values of k, centered at $N/2$:

$$\Omega = \left[\frac{N}{2} - \left(\frac{\theta-1}{2}\right) \quad , \quad \frac{N}{2} + \left(\frac{\theta-1}{2}\right)\right] = [k_1, k_2, \ldots, k_\theta] \subset \mathbb{N} \ . \tag{30}$$

Let us expand (29) for each value of $k \in \Omega$. (It is understood that in practical cases, for each instance of f in each term, one has to replace it by $k\Delta f$.) Doing so yields the following system of linear equations:

$$\begin{bmatrix} F_{0,k_1} \\ F_{0,k_2} \\ \vdots \\ F_{0,k_\theta} \end{bmatrix} \simeq \begin{bmatrix} \left(\left(\mathsf{M}_a^{-1}\right)_{1,1}|_{k_1}\right) & \left(\left(\mathsf{M}_a^{-1}\right)_{1,2}|_{k_1}\right) & \cdots & \left(\left(\mathsf{M}_a^{-1}\right)_{1,\theta}|_{k_1}\right) \\ \left(\left(\mathsf{M}_a^{-1}\right)_{1,1}|_{k_2}\right) & \left(\left(\mathsf{M}_a^{-1}\right)_{1,2}|_{k_2}\right) & \cdots & \left(\left(\mathsf{M}_a^{-1}\right)_{1,\theta}|_{k_2}\right) \\ \vdots & \vdots & \ddots & \vdots \\ \left(\left(\mathsf{M}_a^{-1}\right)_{1,1}|_{k_\theta}\right) & \left(\left(\mathsf{M}_a^{-1}\right)_{1,2}|_{k_\theta}\right) & \cdots & \left(\left(\mathsf{M}_a^{-1}\right)_{1,\theta}|_{k_\theta}\right) \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{\theta-1} \end{bmatrix} . \tag{31}$$

Note that $\left(\mathsf{M}_a^{-1}\right)_{1,\alpha}|_{k_\beta}$ is compact notation for $\left(\mathsf{M}_a^{-1}\right)_{1,\alpha}$ evaluated at $k = k_\beta$ and $f = k_\beta \Delta f$. Let us express (31) in a more compact form as $\mathsf{F}_c \simeq \mathsf{W}\mathsf{B}$, from which we directly deduce the following matrix equation:

$$\mathsf{B} \simeq \mathsf{W}^{-1}\mathsf{F}_c \ . \tag{32}$$

Equation (32) completely determines B from $F_{0,k}$ (the discrete Fourier transform of the digitized function h). According to (24), the knowledge of B specifies b_m ($m \in [0, \theta-1]$). They are the boundary conditions of the system.

Although B is completely determined, (28) cannot be used, for reasons mentioned earlier, to evaluate F_a. Considering the first element of F_a as known (it is the DFT of h), we again expand (22), but in a slightly different manner than we did to obtain (25). We once more expand it for each value of $n \in [0, \theta-1] \subset \mathbb{N}$. This generates a system of θ equations, and for each of these we let p range, this time, from 1 to $\theta - n$. Thereafter, the terms are reorganized to obtain the following system, which is written again in matrix form:

$$\begin{bmatrix} J_1 & J_2 & \cdots & J_\theta \\ J_0 & J_1 & \cdots & J_{\theta-1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & J_1 \end{bmatrix} \begin{bmatrix} F_{1,k} \\ F_{2,k} \\ \vdots \\ F_{\theta,k} \end{bmatrix} \simeq \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{\theta-1} \end{bmatrix} + \begin{bmatrix} -J_0 F_{0,k} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{33}$$

or, more compactly as:

$$\mathsf{M}_b \mathsf{F}_b \simeq \mathsf{B} + \mathsf{C} \ . \tag{34}$$

The general expression for elements of M_b is thus:

$$(\mathsf{M}_b)_{\mu\nu} = I_{\mu-\nu} - i2\pi f I_{\nu-\mu+1} = J_{\nu-\mu+1} \ . \tag{35}$$

Let us now write (34) as:

$$\mathsf{F}_b \simeq \mathsf{M}_b^{-1} (\mathsf{B} + \mathsf{C}) \ . \tag{36}$$

The advantage of (28) over (36) is that it allows, through (32), to compute B. However, it shows singularities at $f = \kappa N \Delta f$, $\kappa \in \mathbb{N}$ which prevent us to compute F_a. Conversely, the advantage of (36) is that it exhibits a higher order than (28) and, provided that θ is odd, F_b is computable for values of $f = k\Delta f$. One should note that, with (36), undetermined values appear for $f = \kappa N \Delta f$, but they can always be solved by Hospital's rule for any odd order. For even orders, there are singularities at $f = (\kappa + 1/2) N \Delta f$ that cannot be removed. This is an imperative reason to choose odd values for θ.

With the knowledge of F_b from (36), the terms of (19), for $p \in [0, \theta]$, are completely determined. Thus, the truncated version of (19) can be written as:

$$H_k \simeq \sum_{p=0}^{\theta} I_p F_{p,k} \ . \tag{37}$$

Let us define a one-row matrix as $\mathsf{I}_\theta = [I_1 \ I_2 \ \cdots \ I_\theta]$, and write (37) as follows:

$$H_k \simeq I_0 F_{0,k} + \mathsf{I}_\theta \mathsf{F}_b \ . \tag{38}$$

With (38), we approximate the Fourier transform (or the inverse Fourier transform) of a digitized function in one dimension. The digitized Fourier transform calculated with (38) is not band-limited (as with the DFT which is periodical). Equation (38) remains valid and accurate as an approximation of the analytical Fourier transform for positive or negative values of k such that $|k| > N/2$ or even for $|k| > N - 1$. (This last property and the following, briefly mentioned in the rest of this section, have been discussed and illustrated with examples in [14].)

A close examination of (31), (33) and (38) reveals that the computation of only one FFT is required. The other terms form a correcting operation to be applied once on each of the N values of the FFT. The time complexity of the correcting operation is $O(N)$ and the time complexity of the FFT is $O(N \log N)$. Hence, the time complexity of the entire algorithm is $O(N \log N)$ when θ is kept constant. The time complexity relatively to θ, the order of the system, is $O(\theta^2)$, but, as long as $\theta \le \theta_{opt}$, the errors on computed results decrease exponentially with the increase of the order θ. Hence, as long as one can afford to increase θ, the trade-off is strongly beneficial.

Equation (38) contains the symbolic form of F_b which can be used as is to form a single symbolic formula without having its terms evaluated numerically. On the other hand, if, for instance, (36) is used to numerically compute each term of F_b for values of k from 0 to $N-1$, it produces θ different sequences of numbers which are actually accurate approximations of the DFT of the derivatives $h_j^{(p)}$, for values of $p \in [1, \theta]$. Thus, applying the inverse DFT operation to each of these sequences generates the corresponding sequences $h_j^{(p)}$ that are very accurate numerical derivatives of the initial function $h_j^{(0)}$of all orders from 1 to θ. This implies that one can numerically compute very accurately the derivatives of any order of a digitized function or signal.

Derivatives calculated in that way are continuous in-between and at each data point. Thus, we obtain spline polynomials of any odd order, with their corresponding properties, merely with DFT (FFT). Furthermore, since (32) is used to compute B, these spline interpolation polynomials, whatever their order, always exhibit optimal boundary conditions, that is to say, accurate end derivatives for each order [15]. Such accurate high-order spline interpolation polynomials allow integrals between any limit to be accurately computed.

Let R_θ be any result (Fourier transform, derivative or integral) obtained with an arbitrary order θ. As long as $\theta + 2 \le \theta_{opt}$, the error on R_θ (noted E_θ) can be fairly estimated since $R_{\theta+2}$ relatively to R_θ can be considered almost as the exact result. To do so, one can use the following relation: $E_\theta = \mathcal{O}(R_{\theta+2} - R_\theta)$, where $\mathcal{O}$ is any operator one can define to meet specific needs.

4 Back in *d*-dimensions

In the previous section we have obtained a highly accurate method to compute the Fourier transform in one dimension. According to (6), this method can be applied sequentially to compute an accurate d-dimensional Fourier transform. In this multidimensional case, for each $\alpha \in \{1, 2, \cdots, d\}$ we have $t_\alpha \in [0, T_\alpha]$. This interval is separated into N_α equal $\Delta t_\alpha = T_\alpha / N_\alpha$ parts, and $f_\alpha = k_\alpha \Delta f_\alpha = k_\alpha / T_\alpha$. As with the ordinary DFT (FFT), the order in which the dimensions are treated is irrelevant. The number of times (38) has to be applied to compute a d-dimensional Fourier transform is:

$$PQ \quad \text{where} \quad P = \prod_{\alpha=1}^{d} N_\alpha \quad \text{and} \quad Q = \sum_{\beta=1}^{d} \frac{1}{N_\beta}\,. \tag{39}$$

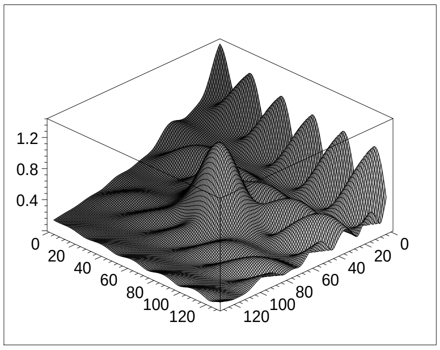
The time complexity is then $O(P \log P)$. Let us put $N_\alpha = a_\alpha N$, $\forall\alpha$, a_α being constants. It is easy to show that the time complexity is $O\left(N^d \log N\right)$ which is the same as for the DFT in d-dimensions.

5 Example in 2 dimensions

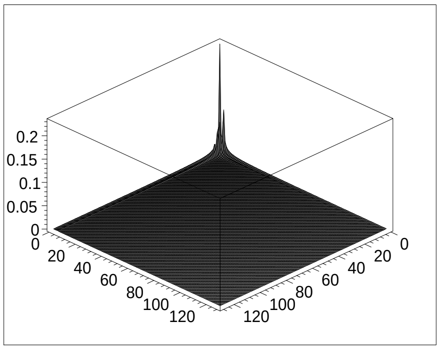
In this section, an example in two dimensions is used to illustrate the algorithm. The choice of such an example is not obvious. That is to say, the function should not be a trivial one; it must be difficult enough for the computation of the Fourier transform to be numerically demanding. On the other hand, for purpose of comparison and accuracy testing, the Fourier transform of the function must be analytically known. The chosen initial function for our example is then the following complex function:

$$\begin{aligned} &h(t_1, t_2) \\ &= \left(\cos(9t_1)\cos(11t_1 + 17t_2)\, e^{-2.5t_1}, e^{-2(t_1+t_2)} + e^{\left[-100(t_1-0.5)^2 - 50(t_2-0.5)^2\right]}\right) \\ &T_1 = T_2 = 1\,. \end{aligned} \tag{40}$$

The real part is a combination of damped oscillations that are slanted in virtue of the damping. The imaginary part is a non-symmetrical Gaussian peak purposely slanted by an exponential to avoid error cancellation by symmetry. For both variables, the function is discontinuous at 0 and at T_α, $\forall\alpha$. Figures 1.a and 1.b show, respectively, the modulus of (40) and of its analytical Fourier transform for $N_1 = N_2 = 128$. The formula of the analytical Fourier transform of this two dimensional function is not shown here since it requires several pages of text.



(a) $|h(t_1, t_2)|$

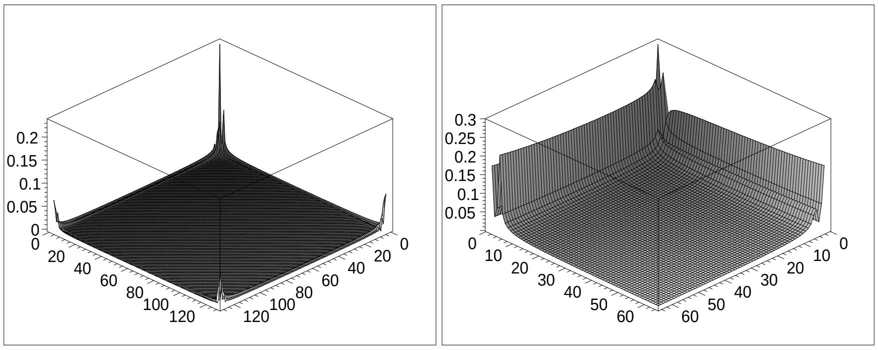


(b) $|\mathcal{F}\{h(t_1, t_2)\}|$

Fig. 1. Modulus of h and of $\mathcal{F}\{h\}$.

Figure 2.a shows the DFT of h. The expected periodical behavior is evident. Figure 2.b shows the modulus of the error of the DFT relatively to the analytical

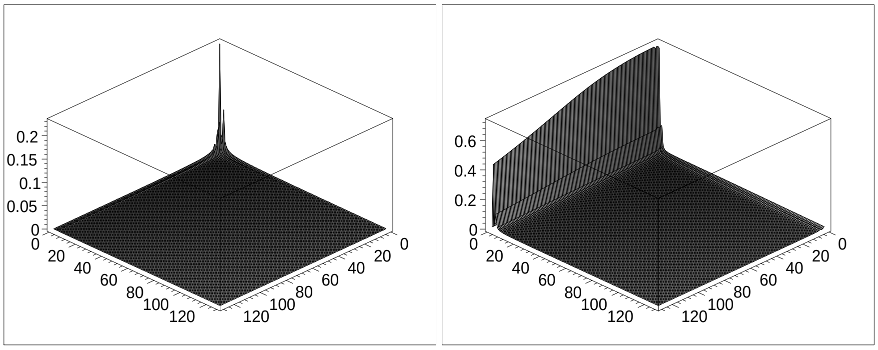
Fourier transform. To be fair, this error must indeed be computed on the first $(N_1/2) \times (N_2/2)$ data points only, since the DFT is periodical.



(a) $|DFT\{h\}|$ (b) $|\text{Error on } DFT\{h\}| \times 10^{-2}$

Fig. 2. Modulus of $DFT\{h\}$ and of error on $DFT\{h\}$ $\left(\times 10^{-2}\right)$. The error is computed on the first 64×64 data points only. Maximum error $= 0.3 \times 10^{-2}$.

Figure 3.a shows the numerical Fourier transform of h computed with (38) for $\theta = 13$. It is clearly seen that this approximation behaves as the analytical Fourier transform and is not periodical. Figure 3.b shows the modulus of the error of this approximation relatively to the analytical Fourier transform, computed, this time, on the full range of the $N_1 \times N_2$ data points. For comparison, one should note the vertical scale factor in Fig. 2.b and 3.b.



(a) $|H_k|$ (b) $|\text{Error on } H_k| \times 10^{-17}$

Fig. 3. Modulus of the numerical Fourier transform given by (38), with $\theta = 13$, and modulus of error on H_k $\left(\times 10^{-17}\right)$. The error is computed on the full range of the 128×128 data points. Maximum error $= 0.7 \times 10^{-17}$.

This same equation (38), used for Fig. 3, is used again, on the same function (40) with different values of $N = N_1 = N_2$ and θ. The averages of the moduli of the error on the numerical Fourier transform given by (38) relatively to the exact analytical Fourier transform are shown in Table 1 in addition to the results obtained with the DFT.

We observe that for small values of N (actually for $N = 8$), increasing θ does not improve accuracy. It means that the optimum value of θ called θ_{opt} is already reached. If N is slightly increased, θ_{opt} grows rapidly and the smallest possible error decreases dramatically. For $N = 8$, $\theta_{opt} < 3$. For $N = 16$, $\theta_{opt} < 5$. For $N = 32$, $\theta_{opt} = 5$. For $N \geq 64$, $\theta_{opt} \geq 13$. We also note that, except for small values of N, the error from (38) is always much smaller than the error from the DFT and, for any θ, it decreases more rapidly with the increase of N than does the error from the DFT with the same increase in N.

Table 1. Average of the modulus of the error on the numerical Fourier transform of the function given by (40) and computed with (38) for different values of N and θ. The last line is the average of the modulus of the error of the DFT of the same function.

	N				
θ	8	16	32	64	128
1	1×10^{-2}	1×10^{-3}	2×10^{-4}	2×10^{-5}	3×10^{-6}
3	3×10^{-1}	1×10^{-3}	9×10^{-6}	3×10^{-7}	1×10^{-8}
5		1×10^{-2}	8×10^{-7}	6×10^{-9}	5×10^{-11}
7			4×10^{-6}	1×10^{-10}	3×10^{-13}
9				3×10^{-12}	2×10^{-15}
11				8×10^{-14}	9×10^{-18}
13				2×10^{-15}	8×10^{-20}
DFT	3×10^{-2}	9×10^{-3}	3×10^{-3}	8×10^{-4}	2×10^{-4}

6 Conclusion

An analytical function contains an infinite quantity of information [16]. It is possible, in principle, at least when the Fourier integral is analytically obtainable, to compute its Fourier transform exactly. In all other cases we have to resort to numerical techniques and depart from analytical forms. The classical sampling of a function uses the Dirac comb which rejects an infinite quantity of information to make the digitized function manageable on a finite computer. The usual tool to numerically compute the Fourier transform of a digitized function in d-dimensions is the DFT, efficiently implemented as the FFT algorithm. However, in many cases the DFT is not an adequate numerical approximation of the Fourier transform.

On the one hand, there is the perfect analytical Fourier transform that is, most of the time, not practical and, on the other hand, we have the very efficient FFT which computes an approximation of the Fourier transform without any attempt to reduce the unavoidable ravages of the Dirac comb. The method presented in this contribution is in-between these two extremes; its position being adjusted by the value of θ, the order of the system. The method provides accurate approximations of the continuous Fourier transform, is no longer periodical and computes the numerical Fourier transform in a broader frequency domain than the half-period of the DFT. The aliasing can be reduced to a negligible level even with a relatively low number of sampled data points. The method gives accurate numerical partial derivatives of any order and the polynonial splines of any odd order with their optimal boundary conditions. The error can be easily computed by comparing the results of two successive odd orders. The time complexity, relatively to the number of sampled data points, is the same as for the FFT. The time complexity, relatively to θ (independent of the time complexity relatively to N) is $O\left(\theta^2\right)$, while the accuracy increases exponentially with θ. Hence, the numerical accuracy increases much more rapidly than the computational cost of the proposed method.

References

[1] Brigham, E. O. (1974): *The Fast Fourier Transform.* Prentice-Hall
[2] Morrison, N. (1994): *Introduction to Fourier Analysis.* Wiley-Interscience
[3] Marchildon, L. (2000): *Mécanique Quantique.* De Boeck Université
[4] Kittel, C. (1983): *Physique de l'État Solide.* Dunod Université
[5] Shapiro, L. G., Stockman G. C. (2001): *Computer Vision.* Prentice Hall
[6] Schutte J. (1981): *New Fast Fourier Transform Algorithm for Linear System Analysis Applied in Molecular Beam Relaxation Spectroscopy.* Rev. Sci. Instrum., 52(3), 400-404
[7] Makinen S. (1982): *New Algorithm for the Calculation of the Fourier Transform of Discrete Signals.* Rev. Sci. Instrum., 53(5), 627-630
[8] Sorella S., Ghosh S. K. (1984): *Improved Method for the Discrete Fast Fourier Transform.* Rev. Sci. Instrum., 55(8), 1348-1352
[9] Froeyen M., Hellemans L. (1985): *Improved Algorithm for the Discrete Fourier Transform.* Rev. Sci. Instrum., 56(12), 2325-2327
[10] Gaskill, J. D. (1974): *Linear Systems, Fourier Transform, and Optics.* Prentice-Hall
[11] Beaudoin, N. (1999): *PhD. Thesis.* Université du Québec à Trois-Rivières. Canada
[12] Butkov, E. (1968): *Mathematical Physics.* Addison-Wesley Reading
[13] Kreyszig, E. (1983): *Advanced Engineering Mathematics.* Wiley
[14] Beaudoin N. (1998): *A High-accuracy Mathematical and Numerical Method for Fourier Transform, Integral, Derivative, and Polynomial Splines of any Order.* Canadian Journal of Physics, 76(9), 659-677
[15] Gerald C. F., Wheatley P. O. (1989): *Applied Numerical Analysis.* Addison-Wesley Reading
[16] Ventsel H. (1973): *Théorie des Probabilités.* Editions Mir