Mathematical Geophysics Summer School, Stanford, August 1998

On Applications of Unequally Spaced Fast Fourier Transforms

G. Beylkin [1]

I Introduction

The Fast Fourier Transform (FFT) algorithm of Cooley and Tukey [7] requires sampling on an equally spaced grid which is a significant limitation in many applications. For example, the discrete Radon transform [3] involves computation of sums

$$\hat{u}_j = \sum_{l=1}^{N} u_l \ \mathrm{e}^{-2\pi \mathrm{i} k \phi(l) j/N} \tag{1.1}$$

where k is a parameter, $j = 1, \ldots, M$, and $\phi(l)$ is a function defined by the selection of a family of curves. Sums of this type are ubiquitous and appear in a variety of applications. We clearly need a fast algorithm for their evaluation. Indeed, the direct evaluation of trigonometric sums

$$\hat{g}_n = \sum_{l=1}^{N_p} g_l \ \mathrm{e}^{-2\pi \mathrm{i} N x_l \xi_n}, \tag{1.2}$$

$n = 1, \ldots, N_f$, where $g_l \in \mathbb{C}$, $|\xi_n|, |x_l| \leq \frac{1}{2}$ is costly and requires $O(N_f \cdot N_p)$ operations (typically $N_f \approx N_p \approx N$ and $N_f \cdot N_p = O(N^2)$). The computational cost is prohibitive in multiple dimensions, where the complexity estimate for computing an analog of (1.2) is $O(N^4)$ in 2D and $O(N^6)$ in 3D.

Computation of the sum in (1.2) can be viewed as an application of the matrix

$$F^0_{ln} = \mathrm{e}^{\pm 2\pi \mathrm{i} N x_l \xi_n}, \tag{1.3}$$

$l = 1, \ldots, N_p$, $n = 1, \ldots, N_f$ to a vector. A special case of (1.3) is the matrix

$$F_{ln} = \mathrm{e}^{\pm 2\pi \mathrm{i} l \xi_n}, \tag{1.4}$$

$l = -N/2, \ldots, N/2 - 1$, $n = 1, \ldots, N_f$ and its adjoint. Algorithms for the fast application of matrices in (1.3), (1.4) and their adjoints to vectors (as well as their multidimensional generalizations) constitute Unequally Spaced Fast Fourier Transform (USFFT) algorithms.

A number of such algorithms has been known in EE and geophysical literature as algorithms for interpolation in the frequency domain (see e.g., [15]). These algorithms were constructed by approximating the ideal filter and were not intended to produce high accuracy although they were adequate in a number of applications.

In [11] Press and Rybicki suggested using Lagrange interpolation to replace the function values at an arbitrary point by several function values on an equally spaced grid surrounding

[1] Department of Applied Mathematics, University of Colorado at Boulder, Boulder, CO 80309-0526 and Fast Mathematical Algorithms and Hardware, 1020 Sherman Avenue, Hamden, CT 06514

that point. Essentially the same idea was proposed by Brandt in [6]. In [16] the Taylor expansion was used to correct for deviations from an equally spaced grid. Although such approaches are significantly better than the direct evaluation of (1.2), they do not lead to very efficient algorithms especially in multidimensional generalizations. A more careful analysis and a much faster algorithm to evaluate (1.2) have appeared in the paper by Dutt and Rokhlin [9], where a specialized interpolation scheme using Gaussian bells has been developed and implemented.

In [4] this problem was addressed by considering projections of functions with singularities (including generalized functions, e.g. $g(x) = \sum_{l=0}^{N_p-1} g_l \, \delta(x - x_l)$) on a subspace of a Multiresolution Analysis effectively replacing these functions by their bandlimited version. This point of view yields very good estimates of the error of the algorithm not available via other approaches. By selecting parameters, it is easy to achieve any prescribed accuracy. Algorithms in [4] consist of three steps. The first step replaces an interpolation scheme (Lagrange interpolation in [11], Taylor expansion in [16], a specialized interpolation scheme involving Gaussian bells in [9] and the steps in [14] involving the use of the Gauss-Legendre quadratures and Lagrange interpolation). The second step is the same as in all algorithms of this type and involves the usual FFT. The third step is a modification (or correction) step which involves multiplying values at each frequency by a pre-computed factor.

If we measure the speed of USFFT in the units of the usual FFT of the same size, then the speed of the algorithm in 1D is (roughly) between 3 and 6 FFTs depending on accuracy, the actual size and on whether the initialization has been counted separately. In 2D USFFT requires asymptotically $\approx$ 22 2D FFTs for the double precision and $\approx$ 12 2D FFTs for the single precision computations.

II Approximation of the ideal filter

From mathematical point of view USFFT solves the problem of replacing a singular function by a smooth function so that their Fourier transforms are almost the same within some frequency band. In particular, for any $\epsilon > 0$ we can choose the space of splines of appropriate order and scale so that the projection on that subspace contains sufficient information to account for half of the frequencies (if we select the oversampling factor 2) with accuracy ϵ. Such projections of generalized functions are useful in their own way for solving partial differential equations with singular coefficients or source terms. Approximations of generalized functions (the so-called discrete approximation to singular functions) appear for example in the context of the Immersed Interface method (see e.g. [8], [12]).

Perhaps the most simple way to understand USFFT algorithm is to use the point of view developed by electrical engineers. Let $\beta^{(m)}$ be the m-th order central B-spline (for convenience m is odd). We need: the Fourier transform of $\beta^{(m)}$,

$$\hat{\beta}^{(m)}(\xi) = \left(\frac{\sin \pi\xi}{\pi\xi} \right)^{m+1}, \tag{2.1}$$

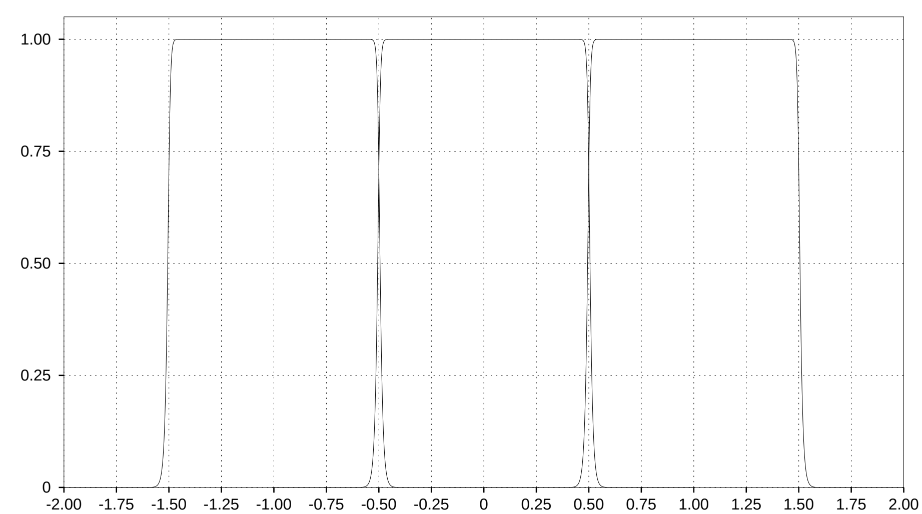

Figure 1: The Fourier transform of Battle-Lemarié scaling function of order $m = 23$. Shown are functions $\hat{\varphi}^{(m)}(\xi)$, $\hat{\varphi}^{(m)}(\xi+1)$ and $\hat{\varphi}^{(m)}(\xi-1)$.

the periodic function $a^{(m)}$,

$$a^{(m)}(\xi) = \sum_{l=-\infty}^{l=\infty} |\hat{\beta}^{(m)}(\xi + l)|^2 = \sum_{l=-m}^{l=m} \beta^{(2m+1)}(l)\ \mathrm{e}^{2\pi \mathrm{i} l \xi}, \tag{2.2}$$

and the Fourier transform of the Battle-Lemarié scaling function ([10], [2]),

$$\hat{\varphi}^{(m)}(\xi) = \frac{\hat{\beta}^{(m)}(\xi)}{\sqrt{a^{(m)}(\xi)}}, \tag{2.3}$$

where $a^{(m)}(\xi) \neq 0$. The Battle-Lemarié scaling function is a very good approximation to the ideal filter as m becomes larger (see Figure 1). We have

$$\hat{\varphi}^{(m)}(\xi) = 1 + O(\xi^{2m+2}), \tag{2.4}$$

whereas (for comparison), we have from (2.1) $\hat{\beta}^{(m)}(\xi) = 1 - \frac{(1+m)\pi^2}{6}\xi^2 + O(\xi^4)$.

In order to compute the bandlimited version of the function and preserve frequencies of the original function within the band, we have to multiply the Fourier Transform of this function by the ideal filter. However, we have to apply such ideal filter in the original domain (where it is a convolution). The problem is that the better is the approximation to the ideal filter, the larger is the significant support of the kernel of the convolution and the less efficient is the algorithm. The key observation in [4] is that the approximation of the form (2.3) can be applied partially in original domain and partially in the Fourier domain. Namely, convolution can be performed with the B-spline in the original domain (which accounts for the numerator in (2.3)) and the denominator in (2.3) can be applied in the Fourier domain (modification step). Such approach leads to a significant improvement in the overall performance. Algorithms in [9] (implicitly) have a similar feature.

We refer to [4] for estimates and details of the algorithms.

III Stolt Migration

An example of a low precision USFFT is Stolt migration [15]. A typical implementation uses the so-called Sinc interpolation. A slight improvement is possible even in this case [5].

In its simplest form, Stolt migration requires evaluation of $U(z,x,0)$,

$$U(z,x,0) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\frac{\omega}{c}}^{\frac{\omega}{c}} \mathrm{e}^{\mathrm{i}z\sqrt{\frac{4\omega^2}{c^2}-k_x^2}+\mathrm{i}xk_x}\,\hat{U}(0,k_x,\omega)\,\mathrm{d}k_x\,\mathrm{d}\omega, \tag{3.1}$$

where $\hat{U}(0,k_x,\omega)$ is obtained by taking the Fourier transform of measured data $U(0,x,t)$,

$$\hat{U}(0,k_x,\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} U(0,x,t)\,\mathrm{e}^{-\mathrm{i}\omega t}\,\mathrm{e}^{-\mathrm{i}k_x x}\,\mathrm{d}x\,\mathrm{d}t. \tag{3.2}$$

Stolt migration [15] is based on the change of variables from the angular frequency ω to the wavenumber k_z in (3.1) according to the formula

$$k_z = \sqrt{\frac{4\omega^2}{c^2} - k_x^2}. \tag{3.3}$$

In order to compute (3.1) using FFT, the wavenumber k_z has to be discretized using equal spacing. This implies that we need $\hat{U}(0,k_x,\omega)$ at non-equally spaced locations obtained from (3.3). These values are obtained by interpolation.

For similar reasons interpolation is needed for Synthetic Aperture Radar (SAR) Imaging. Figures 2 compare design of interpolation filters (Sinc, Parks-McClellan) with spline (USFFT) design. An improvement of about $20-30\%$ can be observed. For details see [5].

Using USFFT package implementation of Stolt migration is very simple: the function $\hat{U}(0,k_x,\omega)$ in (3.2) can be evaluated at any set of points with the desired accuracy by just calling a subroutine from the library.

IV Several Applications

Currently there are several applications where USFFT was used in an essential manner. In [13] the Fourier integrals in domains of complex shape are evaluated using appropriate (unequally spaced) quadratures and USFFT is used as a tool for their computation. Without this tool the cost of computing these integrals is prohibitive. In [1] the problem of computing far field from the near field measurements at unequally spaced grid is solved. Again, without USFFT the cost of such computations is prohibitive.

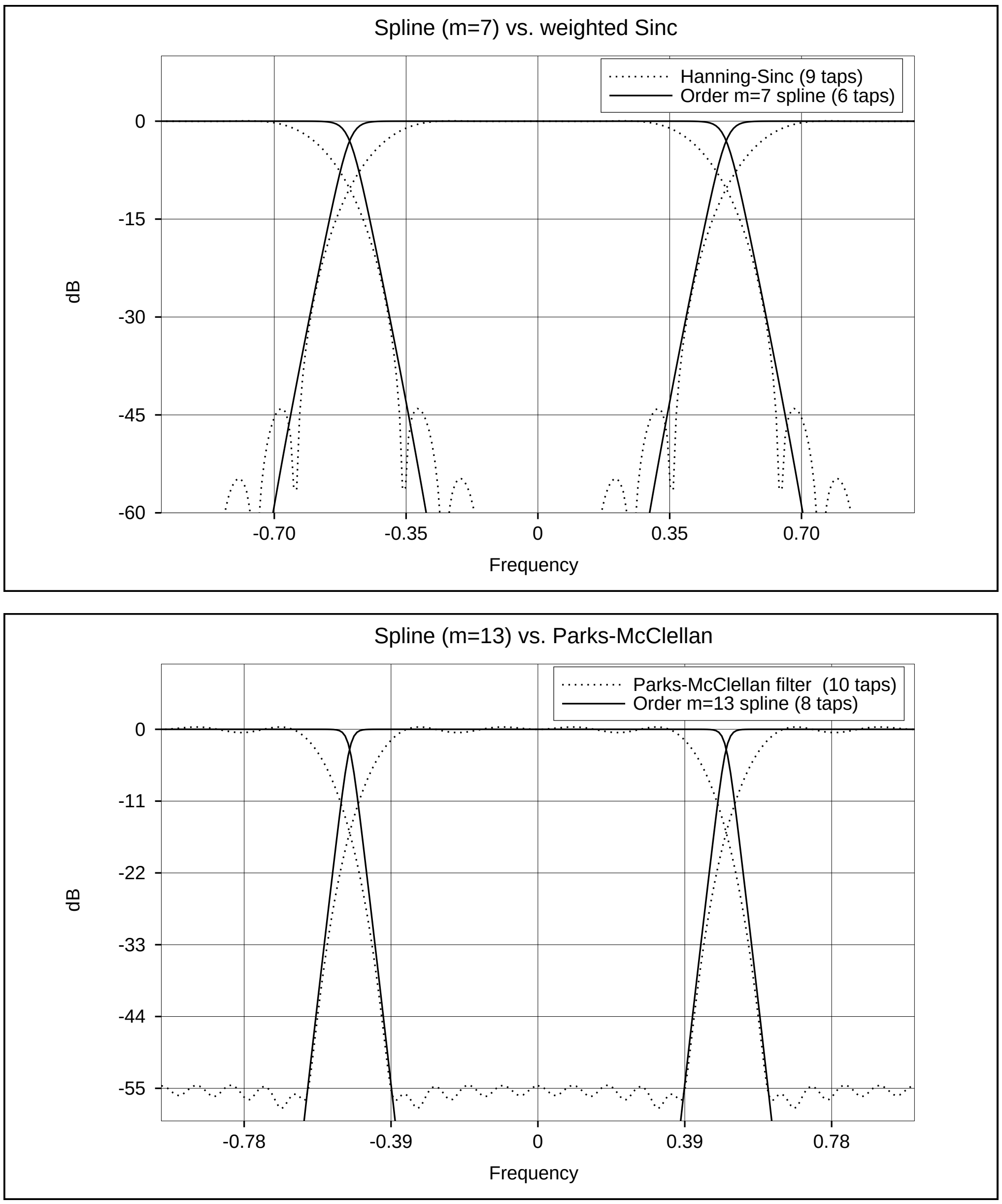

Figure 2: Comparison of spline, Sinc and Parks-McClellan designs of the ideal filter

Let us consider the problem of trigonometric interpolation,

$$f(x_l, y_l) = \sum_{n=-N/2}^{N/2-1} \sum_{m=-M/2}^{M/2-1} g_{nm} \, \mathrm{e}^{-2\pi \mathrm{i} x_l n} \, \mathrm{e}^{-2\pi \mathrm{i} y_l m} \tag{4.1}$$

where $|x_l|, |y_l| < 1/2$, $l = 1, \ldots, N_p$ and $N_p > N \cdot N'$. In order to find g_{nm} given $f(x_l, y_l)$, let us form the (scaled) normal equations for this system,

$$\tilde{f}_{n',m'} = \sum_{n=-N/2}^{N/2-1} \sum_{m=-M/2}^{M/2-1} g_{nm} T_{n-n',m-m'}, \tag{4.2}$$

where

$$\tilde{f}_{n',m'} = \sum_{l=1}^{N_p} f(x_l, y_l) w_l \, \mathrm{e}^{2\pi \mathrm{i} x_l n'} \, \mathrm{e}^{2\pi \mathrm{i} y_l m'}, \tag{4.3}$$

and

$$T_{n-n',m-m'} = \sum_{l=1}^{N_p} w_l \, \mathrm{e}^{-2\pi \mathrm{i} x_l (n-n')} \, \mathrm{e}^{-2\pi \mathrm{i} y_l (m-m')}, \tag{4.4}$$

where $T_{n-n',m-m'}$ is a Toeplitz matrix in both indices and w_l is a set of positive weights. The set of weights w_l can be introduced in a variety of ways depending on application. Both sums in (4.3) and (4.4) can be computed using USFFT and the problem reduces to that of solving a Toeplitz system of linear equations (4.2).

Similarly, the problem of inversion of the Discrete Radon Transform (DRT) in [3] reduces to solving a Toeplitz linear system of equations for each frequency parameter k in (1.1). This algorithm may be called FRT for the Fast Radon Transform.

As an illustration we provide in Figure 4 an example of the harmonic interpolation of magnetic data (collected using a helicopter). This image was generated by solving (4.2).

References

[1] B.K. Alpert, M.H. Francis, and R.C. Wittmann. Planar near-field antenna measurements using nonideal measurement locations. Preprint, 1997.

[2] G. Battle. A block spin construction of ondelettes. Part i: Lemarié functions. *Comm. Math. Phys.*, 110:601–615, 1987.

[3] G. Beylkin. Discrete Radon transform. *IEEE Trans. Acoust., Speech, Signal Processing*, 35(2):162–172, 1987.

[4] G. Beylkin. On the Fast Fourier Transform of Functions Singularities. *ACHA*, 2:363–381, 1995. PAM Report 195, 1994.

[5] G. Beylkin, J.D. Gorman, S. Li-Fliss, and M.A. Ricoy. SAR imaging and multiresolution analysis. In Dominick A. Giglio, editor, *Proc. SPIE: Algorithms for Synthetic Aperture Radar Imagery II*, volume 2487, pages 2144–152. SPIE, 1995.

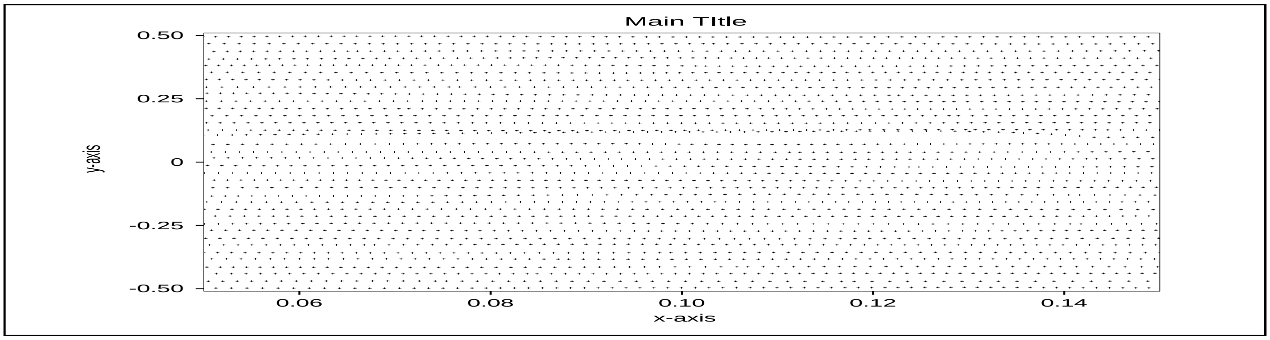

Figure 3: Original Grid

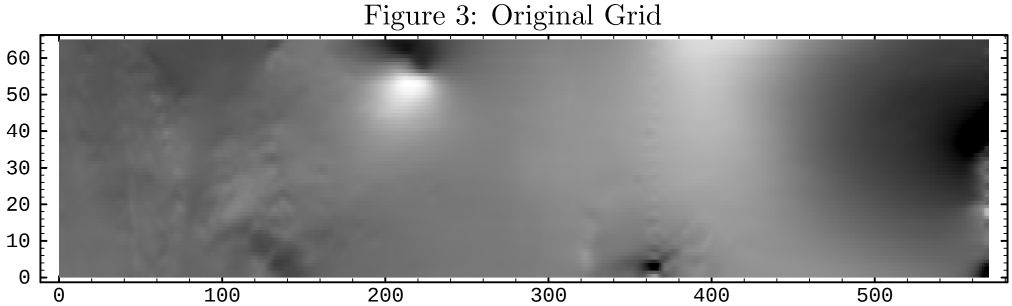

Figure 4: Harmonic Interpolation of Magnetic Data using USFFT

[6] A. Brandt. Multilevel computations of integral transforms and particle interactions with oscillatory kernels. *Computer Physics Communications*, 65:24–38, 1991.

[7] J.W. Cooley and J.W. Tukey. An algorithm for the machine computation of complex Fourier series. *Math. Comp.*, 19:297–301, 1965.

[8] C.S.Peskin. Numerical analysis of blood flow in the heart. *J. Comput. Phys.*, 25:220–252, 1977.

[9] A. Dutt and V. Rokhlin. Fast Fourier Transform for Nonequispaced Data. *SIAM J. Sci. Stat. Comp.*, 1993.

[10] P.G. Lemarié. Ondelettes à localisation exponentielles. *J. Math. Pures et Appl.*, 67:227–236, 1988.

[11] W. H. Press and G. B. Rybicki. Fast Algorithm for spectral analysis of unevenly sampled data. *The Astrophysical Journal*, 338:227–280, 1989.

[12] R.P.Beyer and R.J.LeVeque. Analysis of a one-dimensional model for the immersed boundary method. *SIAM J. Num. Anal.*, 29:332–364, 1992.

[13] S.J. Ruuth. Efficient algorithms for diffusion-generated motion by mean curvature. *Ph.D. Thesis*, University of British Columbia, 1996.

[14] E. Sorets. Fast Fourier Transform of Piecewise Constant Functions. 1993. Yale University Research Report, YALEU/DCS/RR-986.

[15] R. H. Stolt. Migration by fourier transform. *Geophysics*, 43:23–48, 1978.

[16] T. D. Sullivan. A technique for convolving unequally spaced samples using fast fourier transforms. 1990. Sandia Report.