(2)

NPS-MA-92-006

NAVAL POSTGRADUATE SCHOOL

Monterey, California

DFTS ON IRREGULAR GRIDS:
THE ANTERPOLATED DFT

by

Van Emden Henson

Technical Report for Period

October 1990-March 1992

Prepared for: Naval Postgraduate School
Monterey, CA 93943

NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943

Rear Admiral R. W. West, Jr.
Superintendent

Harrison Shull
Provost

This report was prepared in conjunction with research conducted for the Naval Postgraduate School and for the Institute for Mathematics and Its Applications. Funding was provided by the Naval Postgraduate School. Reproduction of all or part of this report is authorized.

Prepared by:

VAN EMDEN HENSON
Asst. Professor of Mathematics

Reviewed by:

RICHARD FRANKE
Chairman
Department of Mathematics

Released by:

PAUL J. MARTO
Dean of Research

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S) NPS-MA-92-006	5 MONITORING ORGANIZATION REPORT NUMBER(S) NPS-MA-92-006
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL (If applicable) MA	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code) Monterey, CA 93943	7b ADDRESS (City, State, and ZIP Code) Monterey, CA 93943
8a NAME OF FUNDING/SPONSORING ORGANIZATION NSF, AFOSR, Naval Postgraduate School 8b OFFICE SYMBOL (If applicable) MA	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER OSMN Direct Funding
8c ADDRESS (City, State, and ZIP Code) Washington, D.C. (NFS) Washington, D.C. (NPS) Monterey, CA 93943	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
DFTS on Irregular Grids: The Anterpolated DFT

12 PERSONAL AUTHOR(S)
Van Emden Henson

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Technical Report	FROM 10/90 TO 03/92	31 March 1992	14

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB GROUP	Discrete Fourier Transform FFT Anterpolation

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

Abstract. In many instances the discrete Fourier transform (*DFT*) is desired for a data set that occurs on an irregular grid. Commonly the data are interpolated to a regular grid, and a fast Fourier transform (*FFT*) is then applied. A drawback to this approach is that typically the data have unknown smoothness properties, so that the error in the interpolation is unknown.

An alternative method is presented, based upon multilevel integration techniques introduced by A. Brandt. In this approach, the kernel, $e^{-i\omega t}$, is interpolated to the *irregular grid*, rather than interpolating the data to the regular grid. This may be accomplished by pre-multiplying the data by the adjoint of the interpolation matrix (a process dubbed anterpolation), producing a new regular-grid function, and then applying a standard FFT to the new function. Since the kernel is C^{∞} the operation may be carried out to any preselected accuracy.

A simple optimization problem can be solved to select the problem parameters in an efficient way. If the requirements of accuracy are not strict, or if a small bandwidth is of interest, the method can be used in place of an FFT even when the data are regularly spaced.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL Van Emden Henson	22b TELEPHONE (Include Area Code) / 22c OFFICE SYMBOL

DD Form 1473, JUN 86 — Previous editions are obsolete — S/N 0102-LF-014-6603

DFTS ON IRREGULAR GRIDS: THE ANTERPOLATED DFT

VAN EMDEN HENSON *

Abstract. In many instances the discrete Fourier transform (*DFT*) is desired for a data set that occurs on an irregular grid. Commonly the data are interpolated to a regular grid, and a fast Fourier transform (*FFT*) is then applied. A drawback to this approach is that typically the data have unknown smoothness properties, so that the error in the interpolation is unknown.

An alternative method is presented, based upon multilevel integration techniques introduced by A. Brandt. In this approach, the kernel, $e^{-i\omega t}$, is interpolated to the *irregular grid*, rather than interpolating the data to the regular grid. This may be accomplished by pre-multiplying the data by the adjoint of the interpolation matrix (a process dubbed anterpolation), producing a new regular-grid function, and then applying a standard FFT to the new function. Since the kernel is C^∞ the operation may be carried out to any preselected accuracy.

A simple optimization problem can be solved to select the problem parameters in an efficient way. If the requirements of accuracy are not strict, or if a small bandwidth is of interest, the method can be used in place of an FFT even when the data are regularly spaced.

1. The formal DFT and the ADFT. The *DFT* is defined as an operation that maps a length-N complex-valued sequence $\{x_0, x_1, \ldots, x_{N-1}\}$ to another length-N complex-valued sequence $\{\hat{x}_0, \hat{x}_1, \ldots, \hat{x}_{N-1}\}$ by the rule

$$\hat{x}_k = \sum_{j=0}^{N-1} x_j e^{-i2\pi jk/N}, \quad \text{for} \quad k = 0, 1, \ldots, N-1. \tag{1}$$

As defined in (1), the *DFT* is performed on data that are presumed to be given on a regular grid, with constant spacing between the data points. Furthermore, the transform values $\{\hat{x}_0, \hat{x}_1, \ldots, \hat{x}_{N-1}\}$ are also presumed to lie on a regular grid in the frequency domain. In many applications, however, the data for a problem are not spaced regularly. It is of some interest, then, to determine how a discrete Fourier transform may be computed for such a data set. To perform this computation, we develop and implement in one dimension an algorithm based on multilevel integration techniques outlined by Achi Brandt ([2], [1]). The method presented here can also be developed for higher-dimensional problems. One application of this technique [6] is in the reconstruction of images from projections (inverting the Radon transform).

To begin, it is necessary to decide what is meant by a Discrete Fourier Transform for irregularly spaced data. Therefore, the concept of a *formal DFT* is introduced, which is defined as follows:

Consider any set of N ordered points in the interval $[0, X)$, satisfying

$$0 \le x_0 < x_1 \ldots < x_{N-1} < X$$

and suppose a vector-valued function (grid function) $u(x_j)$ is specified. The *formal DFT* is defined as the $M = M_1 + M_2 + 1$ quantities

$$\tilde{u}(\omega_l) = \sum_{j=0}^{N-1} u(x_j) e^{-i\omega_l x_j}, \quad \omega_l = \frac{2\pi l}{X}, \quad -M_1 \le l \le M_2, \tag{2}$$

where l is an integer, and M_1 and M_2 are positive integers specifying the range of frequencies of interest. The formal *DFT* may be thought of as an approximation to a

* Department of Mathematics, Naval Postgraduate School, Monterey, California 93943

selection ($-M_1 \le l \le M_2$) of the Fourier coefficients of $u(x)$. In this view, $u(x)$ should be regarded as an X-periodic function known only at the grid points x_j.

It is desired that this sum be calculated to a prescribed accuracy, say $\epsilon||u||_1$, where $||u||_1$ is the discrete L_1 norm $||u||_1 = N^{-1}\sum_{j=0}^{N-1}|u(x_j)|$. Note that any grid spacing is allowed for the x_j (in particular, the spacing needn't be constant), that there is no relationship between the integers M_1, M_2, and N, and that there is no requirement that these integers have any special value, (such as being powers of 2). Calculating the sum in (2) directly would have a computational cost of $O(MN)$ operations. Instead of forming this sum directly, though, an approximation to it will be computed, using an FFT to accelerate the computation.

The procedure begins with the definition of an auxiliary grid, $\Omega^h_{N_*}$, covering the range of values $[0, X)$. Let N_* be an integer, whose value will be determined shortly, and let the grid spacing h be defined by $h = X/N_*$. Then the auxiliary grid consists of the points $y_\tau = (\tau - 1)h$, for $\tau = 1, 2, \ldots, N_*$.

Suppose that the value of some function $g(y)$ on the regular grid $\Omega^h_{N_*}$ is to be interpolated to the gridpoints x_j by Lagrangian interpolation. We will identify the interpolation by specifying the degree of the polynomial to be used. Thus, p-degree Lagrangian interpolation is computed using a polynomial of degree p or less. For each x_j, the $p+1$ nearest neighbors on the grid $\Omega^h_{N_*}$ must be located. Let these points be designated $y_{\kappa(j,0)}, y_{\kappa(j,1)}, \ldots, y_{\kappa(j,p)}$. These points should be chosen modulo X, so that a point near the limits of the interval $[0, X)$ may have neighbors near the other end of the interval. (This is justified because, as will be seen shortly, the function to be interpolated for the formal DFT is X-periodic.) For each x_j, the $p+1$ Lagrangian interpolation weights are computed by

$$w_n(x_j) = \prod_{\substack{m=0 \\ m \ne n}}^{p} \frac{(x_j - y_{\kappa(j,m)})}{(y_{\kappa(j,n)} - y_{\kappa(j,m)})} \quad . \tag{3}$$

and the interpolation of the function g to the gridpoints x_j is given

$$g(x_j) \approx \sum_{n=0}^{p} w_n(x_j)\, g(y_{\kappa(j,n)}) .$$

Letting $\vec{g}$ be the vector of function values $g(y_\tau)$ and $\tilde{g}$ be the vector of interpolated values $g(x_j)$, the interpolation may be written in matrix form

$$\tilde{g} = [I^x_y]\vec{g}$$

where I^x_y is the $N \times N_*$ interpolation matrix mapping a function on $\Omega^h_{N_*}$ to the gridpoints $\{x_j\}$. The entries of this matrix are

$$[I^x_y]_{jm} = \begin{cases} w_n(x_j) & \text{if } \kappa(j,n) = m \\ 0 & \text{else} \end{cases} .$$

We are now ready to compute an approximation to equation (2). The strategy will be to interpolate, for each ω_l, values of the kernel $e^{-i\omega_l x}$ from the auxiliary grid $\Omega^h_{N_*}$. That is, equation (2) is approximated by

$$\hat{u}(\omega_l) \approx \sum_{j=0}^{N-1} u(x_j)\phi(\omega_l, x_j) \tag{4}$$

where

$$\epsilon(\omega_l, x_j) = \sum_{n=0}^{p} w_n(x_j) e^{-i\omega_l y_{k(j)+n}} .$$

Let the column vector of exponential values $e^{-i\omega_l y_k}$ be designated $\vec{e}_l$, and the vector of interpolated kernel values $\epsilon(\omega_l, x_j)$ be denoted $\tilde{e}_l$ Then this interpolation can be written

$$\tilde{e}_l = [I_y^x]\vec{e}_l \ .$$

Notice, however, that $\tilde{e}_l$ is to be used as the l^{th} row of the matrix giving the kernel of the summation in (4). To compute $\tilde{e}_l$ as a *row* vector, the adjoint of the matrix equation equation is needed, namely

$$(\tilde{e}_l)^T = (I_y^x \vec{e}_l)^T = (\vec{e}_l)^T [I_y^x]^T \ .$$

Let us define the M-vector $\hat{u} = \hat{u}(\omega_l)$, the N-vector $\vec{u} = u(x_j)$, and the matrices giving the kernels of equations (2) and (4) as W and $\tilde{W}$, respectively. Let the $M \times N_*$ matrix whose l^{th} row consists of $e^{-i\omega_l y_n}$ for the N_* points on $\Omega_{N_*}^h$ be designated W (it is useful to observe that this matrix consists of M consecutive rows of the standard DFT kernel for a uniform grid with N_* points). Then the formal DFT is approximated

$$\begin{aligned} \hat{u} &= W\vec{u} \\ &\approx \tilde{W}\vec{u} \\ &= W[I_y^x]^T\vec{u} \ . \end{aligned}$$

This notation can be simplified slightly by denoting the vector created by multiplying $\vec{u}$ by $[I_y^x]^T$, as $\tilde{u}$. Since the matrix $[I_y^x]^T$ is the adjoint of the Lagrangian interpolation matrix, the process of computing $\tilde{u} = [I_y^x]^T\vec{u}$ has been dubbed *anterpolation*. Then the approximation to the formal DFT is

$$\hat{u} \approx W\tilde{u} \ , \tag{5}$$

which we call the *Anterpolated Discrete Fourier Transform (ADFT)*.

The $ADFT$, as a matrix multiplication, requires $O(MN_*)$ operations. In general, N_* will exceed N, so as a matrix-vector multiplication, the $ADFT$ has no advantage over (2). If, however, N_* is selected appropriately, the approximation can be computed quite rapidly. Let $M_* = \max\{M_1, M_2\}$. Then if N_* is selected such that $N_* \geq 2M_*$, and at the same time N_* is a number for which an FFT module exists, then the fast Fourier transform can be applied to compute the DFT summation

$$FFT\{\tilde{u}\} = \frac{1}{N_*}\sum_{r=0}^{N_*-1} \tilde{u}_r e^{-i2\pi rl/N_*} \quad \text{for } l = -\frac{N_*}{2}+1, -\frac{N_*}{2}+2, \ldots \frac{N_*}{2} \ .$$

Recalling that $h = X/N_*$, it may be seen that the DFT summation therefore yields $(1/N_*)\hat{u}((2\pi l)/X)$. Multiplying by N_* thus yields a set of values that includes, as a subset, all the desired values of $\hat{u}(\omega_l)$.

Computing the $ADFT$, then, consists of two phases:

1. $\tilde{u}$ is computed from $\vec{u}$ by *anterpolation*: $\tilde{u} = [I_y^x]^T\vec{u}$.
2. $\hat{u}$ is computed from $\tilde{u}$ by a Fast Fourier Transform.

2. Operation count for the ADFT. The cost of computing the $ADFT$ consists of the cost of computing the interpolation weights, the cost of computing the vector $\tilde{u} = [I_y^x]^T \vec{u}$, and the cost of the FFT on N_* points.

Computing the $(p+1)N$ interpolation weights, $w_n(x_j)$, by the formula in (3) is the cost of the computing the numerator, since the regular spacing on $\Omega_{N_*}^h$ means that the denominators of $w_n(x_j)$ are independent of j. To compute the numerators, the product $\prod_{m=0}^{p} (x_j - x_{\kappa(j,m)})$ is computed for each x_j, requiring $2p+1$ operations. Then the n^{th} interpolation weight can be obtained by dividing by the product of $(x_m - x_{\kappa(j,m)})$ with the precomputed denominator $\prod_{\substack{m=0 \\ m \neq n}}^{p} (x_{\kappa(n,j)} - x_{\kappa(j,m)})$, requiring 2 operations for each of the $p+1$ weights associated with the point x_j. The calculation of the weights thus requires $O(N(p+1))$ operations. It is important, however, to note that the calculation of the weights is dependent only on the relationships between the gridpoints $\{y_k\}$ and $\{x_j\}$, and is independent of the data set, $u(x_j)$. This means that if a known set of gridpoints $\{x_j\}$ and a standard auxiliary grid $\Omega_{N_*}^h$ are to be used repeatedly, the interpolation weights $w_n(x_j)$ may be precomputed and stored, and needn't be included in the cost of the algorithm. This will be assumed to be the case.

The matrix $[I_y^x]^T$ is $N_* \times N$ and the data vector $\vec{u}$ is $N \times 1$, so the computation of $\tilde{u} = [I_y^x]^T \vec{u}$ would be $O(NN_*)$ if performed as a matrix-vector multiplication. There is, however, a much more efficient method. The index table $\kappa(j,n)$ can be stored along with the interpolation weights. For each x_j and for each n, the value of $\kappa(j,n)$ is the index of the n^{th} interpolation neighbor that is used to interpolate from $\Omega_{N_*}^h$ to the gridpoint x_j. The periodic nature of the kernel being interpolated means that the interpolation is always to a gridpoint x_j in the center of the set of p interpolation points (as is well known, [5], the Lagrangian interpolation is better behaved when this is the case). If p is odd, then x_j always lies between $y_{\kappa(j,\frac{p-1}{2})}$ and $y_{\kappa(j,\frac{p+1}{2})}$, while if p is even, then $y_{\kappa(j,\frac{p}{2})}$ is the closest gridpoint on $\Omega_{N_*}^h$ to x_j.

Computing the vector $\tilde{u}$ is then very easy, and may be done in $2N(p+1)$ operations, according to the algorithm:

1. Initialize $\tilde{u}(y_\tau) = 0$ for all $y_\tau \in \Omega_{N_*}^h$. $(1 \leq \tau \leq N_*)$
2. For $j = 0, 1, \ldots, N-1$
   For $n = 0, 1, \ldots, p$
   Set $\tilde{u}(y_{\kappa(j,n)}) \leftarrow \tilde{u}(y_{\kappa(j,n)}) + w_n(x_j)\, u(x_j)$.

Having computed the values of $\tilde{u}$, consider now the cost of the FFT portion of the $ADFT$. This is simply the cost of an N_*-point FFT. In the next section criteria for choosing N_* will be determined. For now the only requirement is that an FFT can be computed on a vector of length N_*. As such, N_* must have factors for which FFT modules are available. For the purpose of an operation count, however, it is easiest to assume that N_* is a power of 2. Indeed, we shall see that we have great flexibility in our selection of N_*, and since powers of 2 or 4 produce the most efficient FFTs, this is a good assumption. In this case, the cost of the FFT portion of the $ADFT$ is $O(N_* \log_2 N_*)$.

The costs of the $ADFT$ can now be computed. In terms of data storage it requires four arrays. One is the N-point vector containing the input data, $\vec{u}$. In addition, an N_*-point complex vector is required for the input and output of the FFT. Assuming that the weights are precomputed and stored, two auxiliary arrays are necessary, an $N \times (p+1)$ real (or double precision) array holding the interpolation weights $w_n(x_j)$, and the $N \times (p+1)$ integer array of indices, $\kappa(j,n)$.

If the operations of multiplication and addition are counted equally, and if the weights and indices are pre-stored, the operation count is $C_1 N_* \log_2 N_*$ complex operations for the FFT portion of the algorithm, where C_1 depends on the choice of FFT algorithm. The computation of $\tilde{u}$ entails $2N(p+1)$ operations that are real or complex according to whether $\vec{u}$ is real or complex. Counting both phases of the algorithm, the operation count of the $ADFT$ is

$$C_1 N_* \log_2 N_* + 2N(p+1) .$$

This should be compared with the operation count of the formal DFT, which is $O(MN)$. The computation of the $ADFT$ is more efficient provided M is larger than $2(p+1) + (N_* \log_2 N_*)/N$, a condition that will generally occur in practice.

3. Error analysis for the Anterpolated DFT. One of the attractive features of the $ADFT$ is that the interpolation is performed on the *kernel*, which has known smoothness properties, rather than the data set, which generally has unknown smoothness properties. Since interpolation error depends on the smoothness of the interpolated function, the error committed by using the $ADFT$ is relatively easy to analyse.

Consider the error in p-degree Lagrangian polynomial interpolation, when the interpolation is from a set of $p+1$ gridpoints that are equally spaced. Let these gridpoints be designated $\xi_0, \xi_1, \ldots, \xi_p$. A function $f(x)$, whose values are known at these gridpoints, is to be interpolated to the point $x \in [\xi_0, \xi_p]$. Let $x = \xi_0 + th$, where h is the grid-spacing, and $t \in [0, p)$. The approximation to $f(\xi_0 + th)$ is the value of the Lagrangian interpolation polynomial $P_p(\xi_0 + th)$.

$$P_p(x) = \sum_{i=0}^{p} w_i(x) f(\xi_i) \quad \text{where} \quad w_i(x) = \prod_{\substack{n=0 \\ n \neq i}}^{p} \frac{(x - \xi_n)}{(\xi_i - \xi_n)} .$$

Defining $\pi_p(t) = t(t-1)(t-2)\ldots(t-p)$, and $\zeta \in [\xi_0, \xi_p]$, the error in the interpolation is bounded by

$$|f(\xi_0 + th) - P_p(\xi_0 + th)| \leq \frac{|\pi_p(t)| h^{p+1} Q_{p+1}}{(p+1)!} \tag{6}$$

where $Q_{p+1} = \max_{[\xi_0, \xi_p]} |f^{(p+1)}(\zeta)|$. See [8], pages 264-270, for a derivation of this error term.

It is useful to bound this error more precisely. To do this, we examine the behavior of the factorial polynomial $\pi_p(t)$. This polynomial has been well-studied, and many results can be found in various numerical analysis texts, ([5], [8], [9], [11]). These results, however, are developed for the case that x can be anywhere in $[\xi_0, \xi_p]$. In the present case the interpolation is always to the *center* subinterval. Thus for p odd, $t \in [\frac{p-1}{2}, \frac{p+1}{2}]$, while for p even, either $t \in [\frac{p}{2}, \frac{p}{2}+1]$ or $t \in [\frac{p}{2}-1, \frac{p}{2}]$.

To shorten the discussion, assume that p is odd. This is the most common case, where $p = 1$ gives linear interpolation and $p = 3$ give cubic interpolation. Similar results can be obtained for p even. Consider the following lemma, the proof of which may be found in [6].

LEMMA 1. *If p is a positive odd integer, then*

$$\max_{t\in[\frac{p-1}{2},\frac{p+1}{2}]} \frac{|\pi_p(t)|}{(p+1)!} < \frac{1}{2^{p+1}} .$$

This result can be used to find an error bound for the $ADFT$. In this case the functions being interpolated are

$$e^{-i\omega_l x} \text{ for } l = -\frac{N_*}{2}+1, -\frac{N_*}{2}+2, \ldots \frac{N_*}{2} .$$

From this set of functions, the only ones whose values are of interest are those for l between $-M_1$ and M_2. Recalling the definition $M_* = \max\{M_1, M_2\}$, the largest absolute value among the frequencies of interest is $\omega_M = (2\pi M_*)/X$. Therefore

$$\max_{x\in[x_0, x_{N-1}]} \left| \frac{\partial^{p+1}}{\partial x^{p+1}} \left(e^{-i\omega_M x} \right) \right| = (\omega_M)^{p+1} .$$

Inserting this and the bound from Lemma 1 into equation (6) gives

$$|e^{-i\omega_l x_j} - e(\omega_l, x_j)| \leq \left(\frac{h\omega_M}{2}\right)^{p+1} , \tag{7}$$

which is then used to obtain

$$\begin{aligned} |\hat{u} - W\tilde{u}| &= |\textstyle\sum_{j=0}^{N-1} u(x_j) e^{-i\omega_l x_j} - \sum_{j=0}^{N-1} u(x_j)\, e(\omega_l, x_j)| \\ &\leq \textstyle\sum_{j=0}^{N-1} |u(x_j)| |e^{-i\omega_l x_j} - e(\omega_l, x_j)| \\ &\leq (h\omega_M/2)^{p+1} \textstyle\sum_{j=0}^{N-1} |u(x_j)| \\ &= (h\omega_M/2)^{p+1} N \|u\|_1 . \end{aligned}$$

Finally, substituting $h = X/N_*$ and $\omega_M = (2\pi M_*)/X$ establishes the desired error bound. The error in the $ADFT$ approximation to the formal DFT is bounded, for $-M_1 \leq l \leq M_2$, by

$$|\hat{u}(\omega_l) - [\bar{W}\tilde{u}](\omega_l)| \leq \left(\frac{M_*\pi}{N_*}\right)^{p+1} N \|u\|_1, \tag{8}$$

where $\omega_l = 2\pi l/X$, and the $ADFT$ (5) is computed using an FFT of length N_*.

Since the bound holds for all desired values of ω_l, it then follows that

$$\|\hat{u} - [\bar{W}\tilde{u}]\|_\infty \leq \left(\frac{M_*\pi}{N_*}\right)^{p+1} N \|u\|_1 . \tag{9}$$

where $\|\bullet\|_\infty$ is defined as the maximum absolute value in the vector. It is also worth noting that an error bound for any desired frequency can be obtained by replacing ω_M with ω_l in the derivation, leading to

$$|\hat{u}(\omega_l) - [\hat{W}\tilde{u}](\omega_l)| \leq \left(\frac{l\pi}{N_*}\right)^{p+1} N\|u\|_1 \quad . \tag{10}$$

This is especially useful information for those occasions when only the low frequency components are of interest, or when the accuracy required of the approximation is greater for the low frequencies than for the high frequencies.

4. Selection of p and N_*. The error bound just derived is useful in that it provides a way to select the operational parameters N_* and p. Recall that the goal is to calculate an approximation to $\hat{u}$ to some prescribed accuracy, $|\hat{u} - \hat{W}\tilde{u}| \leq \epsilon\|u\|$. In practice we will want to make the error small, so it will be assumed that $\epsilon \ll 1$. Comparison with (8) gives the requirement

$$\left(\frac{M_*\pi}{N_*}\right)^{p+1} \leq \frac{\epsilon}{N},$$

which may be written as

$$N_* \geq M_*\pi \left(\frac{N}{\epsilon}\right)^{\frac{1}{p+1}} \quad . \tag{11}$$

For a given formal DFT, the values of M_*, N, and ϵ are considered to be part of the problem specification. To ensure that the specified accuracy is obtained, it is only necessary to select integers N_* and p so that (11) is satisfied. Naturally, there may be *many combinations of parameter values that achieve this goal.* The parameters should therefore be selected to fullfill some other desirable property as well. Specifically, they should be selected also to minimize the computational effort of the algorithm.

*To see how this may be accomplished, recall that the work involved in comput*ing the *ADFT* with N_* points on the auxiliary grid $\Omega^h_{N_*}$ and p-degree Lagrangian interpolation is

$$O(N_* \log_2 N_*) + O(N(p+1)) \quad .$$

The value of the constant on the $O(N(p+1))$ term depends on whether the weights and indices are pre-stored, or calculated "on the fly". For the analysis that follows, we assume the weights and indices are pre-stored, in which case the constant is 2. The constant on the first term depends on several factors. *FFT*s generally have a complexity of $(N/q)\log(N/q)$ for some number $q > 1$. If the data have certain symmetries, then a specialized *FFT* may be used for faster computation ([3], [7], [10]). The variety of available *FFT* algorithms pursuades us to leave the constant on the first term as an unspecified parameter, C_1.

The total work in computing the *ADFT* can therefore be written as a function of the two parameters N_* and p. For a fixed problem size (N and M_*), and a prescribed error tolerance ϵ, the work in computing the *ADFT* to the required accuracy is

$$W(N_*, p) = C_1 N_* \log_2 N_* + 2N(p+1) \quad . \tag{12}$$

and we seek an optimal parameters minimizing $W(N_*, p)$ over all combinations (N_*, p) satisfying (11), if such a choice exists.

Limiting cases may be determined by examining *nearest neighbor* interpolation ($p = 0$), as well as extremely high degrees of interpolation ($p \to \infty$). Substituting the limiting values of p into (11), and noting that equality will suffice to ensure that the required accuracy is attained, we obtain bounds for the selection of N_*, namely

$$M_*\pi \leq N_* \leq \frac{M_*\pi N}{\epsilon}$$

for all values of $p \geq 0$.

The existence and uniqueness of optimal solutions are fairly easy to establish. $W(N_*, p)$ is continuous with respect to each of its variables, and both of the first partial derivatives are everywhere positive. This observation leads to

LEMMA 2. *Let S be the set $\left\{(N_*, p) : N_* > M_*\pi(N/\epsilon)^{1/(p+1)}\right\}$, and let ∂S be that portion of the boundary of S given by $\left\{(N_*, p) : N_* = M_*\pi(N/\epsilon)^{1/(p+1)}\right\}$. Then if $(x_0, y_0) \in S$, there exists a point $(\xi, \eta) \in \partial S$ such that $W(\xi, \eta) < W(x_0, y_0)$.*

Proof: Since $(x_0, y_0) \in S$, the point $(\xi, y_0) \in \partial S$, where $\xi = M_*\pi\left(\frac{N}{\epsilon}\right)^{\frac{1}{y_0+1}}$. Furthermore, $\xi < x_0$. Then since the partial derivative of the work function with respect to N_* is everywhere positive, $W(\xi, y_0) < W(x_0, y_0)$. ∎

The utility of Lemma 2 is that the optimization problem can be rewritten as a problem in a single variable. Since for every point in S there is some point along ∂S that requires less work, it is only necessary to seek a minimum from the points of ∂S. This can be done by parameterizing N_* and p as functions of a single variable,

$$b = \left(\frac{N}{\epsilon}\right)^{\frac{1}{p+1}} . \tag{13}$$

Then on ∂S we find that

$$N_* = M_*\pi b \quad \text{and} \quad p + 1 = \log_b\left(\frac{N}{\epsilon}\right) . \tag{14}$$

Since $0 \leq p < \infty$, the value of b is restricted to the interval $(1, N/\epsilon]$. Substituting these expressions into (12), the work equation may be rewritten as a function of b alone

$$W(b) = C_1 M_*\pi b \log_2(M_*\pi b) + 2N \log_b\left(\frac{N}{\epsilon}\right) , \tag{15}$$

and the problem is to minimize (15) subject to the constraint $1 < b \leq (N/\epsilon)$. Once b is determined, the necessary values of N_* and p can be obtained from (14). We may now establish

THEOREM 1. *There exists a unique value b that minimizes (15) subject to $1 < b \leq (N/\epsilon)$. Therefore the work function*

$$W(N_*, p) = C_1 N_* \log_2 N_* + 2N(p + 1)$$

has a unique minimum, subject to the constraints

$$N_* \geq M_*\pi\left(\frac{N}{\epsilon}\right)^{\frac{1}{p+1}} \quad \text{and} \quad 0 \leq p < \infty .$$

Proof: $W(b)$ is continuous and differentiable with respect to b on $(1, N/\epsilon]$. Differentiating equation (15) yields

$$W'(b) = K_1 \ln(K_2 b) - \frac{K_3}{b(\ln b)^2} \quad , \tag{16}$$

where

$$K_1 = \frac{C_1 M_* \pi}{\ln 2} \, , \quad K_2 = 2M_* \pi \, , \text{ and } \quad K_3 = 2N \ln\left(\frac{N}{\epsilon}\right) .$$

For $W'(b) = 0$, then, b must satisfy $b(\ln b)^2 \ln(K_2 b) = K_3/K_1$. Now W' is also continuous and differentiable on $(1, N/\epsilon]$, and differentiating yields

$$W''(b) = \frac{K_1}{b} + K_3 \left(\frac{\ln b + 2}{b^2 (\ln b)^3}\right) \quad .$$

Since $b > 1$ we see that $W''(b) > 0$ for all $b \in (1, N/\epsilon]$, so any critical point in the interval must correspond to a local minimum.

It is apparent that $W'(b) \to -\infty$ as $b \to 1$. Examination of the endpoint $b = N/\epsilon$ reveals that since $K_1 > \pi$, $K_2 > \epsilon$, and $\epsilon \ll 1$, we have that $W'(N/\epsilon) > 0$. Further, $W''(b) > 0$ implies that $W'(b)$ has exactly one sign change in the interval $(1, N/\epsilon]$. The point b_0 at which this occurs is therefore a global minimum for $W(b)$, and the value $W(N_*, p)$, where

$$N_* = M_* \pi b_0 \quad \text{and} \quad p = \log_2\left(\frac{N}{\epsilon}\right) - 1$$

is the unique global minimum for W on ∂S. ∎

The values of N_* and p obtained in this manner are real numbers. There is only a limited number of integers for which efficient FFTs exist, and Lagrangian interpolation requires p to be an integer. Further, this entire discussion has been predicated on the assumption that p is an odd integer, although a similar analysis can be made for p even. Once the theoretical values of N_* and p are determined, they must be modified to allow computation. There is some flexibility in this, but certainly selecting N_* to be the first integer larger than $M_* \pi b$ for which an FFT exists, and choosing p to be the smallest odd integer greater than

$$\log_2\left(\frac{N}{\epsilon}\right) - 1$$

will suffice.

In order to find the optimal values of p and N_* it is necessary to find the value of b satisfying

$$b(\ln b)^2 \ln(K_2 b) = \frac{K_3}{K_1} \, . \tag{17}$$

While an analytic solution of this equation cannot be found, Newton's iteration may be used. Table 1 displays optimal parameters N_* and p for several combinations of N, M_*, and ϵ.

N	M_*	ϵ	N_*	p		N	M_*	ϵ	N_*	p
32	8	.1	48.7	7.7		128	32	.1	193	9.9
32	32	.1	142.6	15.5		128	64	.1	325.7	13.8
32	64	.1	257.6	22.2		128	128	.1	570.7	19.4
32	8	.01	52.9	9.8		128	32	.01	206.7	12.1
32	32	.01	150.1	19.1		128	64	.01	344.1	16.6
32	64	.01	267.8	27.1		128	128	.01	595.6	23.1

Table 1. *Optimal parameters N_* and p computed for various problems.*

5. An $ADFT$ Example. To illustrate the $ADFT$, consider the problem of computing the formal DFT of the function $u(x) = [(\pi - x)/\pi]^2$, sampled on an irregular grid. The irregular grid consists of $N = 128$ points x_j randomly spaced in the interval $(0, 2\pi)$. Since the extent of the interval is 2π, the frequencies ω_l are just the integers l, and the formal DFT is

$$\hat{u}(l) = \sum_{j=0}^{N-1} u(x_j) e^{-ilx_j}, \qquad -64 \le l \le 64 . \tag{18}$$

The sampled data are shown in the top of Figure 1. The real part of (18) is plotted on the bottom of Figure 1. The $ADFT$ was used to approximate the formal DFT, with values of $N_* = 128, 256, 512$, and 1024. Figure 2 displays, for each choice of N_*, the absolute value of the error $|\hat{u}(l) - [W\tilde{u}]_j(l)|$, plotted as a function of l. Linear interpolation ($p = 1$) was used in each case. Note that increasing the value of N_* produces a noticeable decrease in the error, and that the error increases with increasing wavenumber, as might be inferred from (10). Figure 3 displays the effect of using different values of p for fixed N_*. It may be seen that the error decreases rapidly as p is increased. Equation (9) predicts that the error should decrease at least as fast as $\left(\frac{M_*}{N_*}\right)^{p+1}$ decreases as p or N_* are increased. Table 2 gives both the infinity and L_2 norms of the error $|\hat{u}(l) - [W\tilde{u}]_j(l)|$ for several values of p and each of $N_* = 256$ and $N_* = 512$. For $N_* = 256$, the error bound decreases by 0.6169 each time p is increased by 2. The experimental error is diminished by a factor of approximately 0.3 as p is increased from 1 to 3, and by a factor of approximately 0.4 with each succeeding increase, better than the theory predicts. Similarly, for $N_* = 512$, the theoretical bound decreases by 0.15421 as p is increased by 2, while the experimental decrease is approximately 0.11 for each increase, a slightly better result. Numerical experiments on numerous other irregularly sampled functions, with various degrees of smoothness, produced similar results. In these experiments the $ADFT$ behaved in a similar fashion as it did for the function discussed above. There is dramatic improvement with increasing values of N_*, and p. As might be expected, the error diminished faster with smooth functions than discontinuous functions.

N_*	p	$\|\|Error\|\|_\infty$	$\|\|Error\|\|_2$		N_*	p	$\|\|Error\|\|_\infty$	$\|\|Error\|\|_2$
256	1	1.20663	0.434861		512	1	0.290501	0.109943
256	3	0.357889	0.116080		512	3	0.029351	0.008315
256	5	0.144478	0.039136		512	5	0.003360	0.000817
256	7	0.061305	0.014983		512	7	0.000419	9.66385e-05

Table 2. *Errors of the $ADFT$ for various values of N_* and p.*

6. Some Open Questions about the *ADFT*. Like the continuous Fourier transform, the *DFT* has several important properties, such as linearity, the convolution and correlation properties, the shifting property, the modulation property, and Parseval's relation. To what extent these properties hold for the *ADFT* is an open question. The linearity holds can be established immediately, by noting that both the formal *DFT* and the *ADFT* can be written as matrix operations, so they are linear operators. Certain symmetry properties are easy to establish. For example, applying the *ADFT* to a real-valued vector will yield a *conjugate symmetric* result, that is $\hat{u}(\omega) = \overline{\hat{u}(-\omega)}$, because the vector $[\mathcal{I}_y^x]^T \vec{u}$ is real-valued, and because the *ADFT* is computed by applying the *FFT* operator to this vector. The *DFT*, and therefore the *FFT*, maps a real vector to a conjugate symmetric vector [4]. Applying the *DFT* to data vectors with other symmetries (even, odd, quarter-wave, etc.) yields output vectors with other types of symmetries [10]. It is natural to ask which of these symmetry properties are inherited by the formal *DFT* or the *ADFT*. It seems reasonable to postulate that if the irregular gridpoints are symmetrically disposed and the function $u(x_j)$ is symmetric then the symmetry property of the *DFT* might be inherited by the formal *DFT* and the *ADFT*.

An important question is: How is the formal *DFT* related to the continuous Fourier transform? That is, to what extent, and with what error, does the formal *DFT* approximate the *FT*? Answering this question may prove to be a lengthy process. Many related questions will also arise. For example, how does the sampling theorem apply to an irregular grid? What frequencies can be represented accurately, and what constitutes aliasing? Is there some analog to the Poisson summation theorem? Many problems feature irregularly spaced data, so it may be assumed that these questions are of some interest.

7. Acknowledgements. The author wishes to thank William L. Briggs and Stephen F. McCormick, of the University of Colorado at Denver, and Achi Brandt, of the Wiezmann Institute of Science, for their many constructive criticisms and suggestions. This research was supported in part by grants NSF DMS-8704169 and AFOSR-86-0126.

REFERENCES

[1] Achi Brandt. Multilevel computations of integral transforms and particle interactions with oscillatory kernels. In *Proceedings of the IMACS 1st International Conference on Computational Physics*, Boulder, Colorado, 1990.

[2] Achi Brandt and A. A. Lubrecht. Multilevel matrix multiplication and fast solution of integral equations. *Journal of Computational Physics*, 1990.

[3] William L. Briggs. Further symmetries of in-place FFTs. *SIAM Scientific and Statistical Computing*, 8:644–655, 1987.

[4] James W. Cooley, P. A. W. Lewis, and P. D. Welch. The fast Fourier transform algorithm: considerations in the calculation of sine, cosine, and Laplace transforms. *Journal of Sound Vibrations*, 12:27–34, 1970.

[5] Germund Dahlquist and Ake Bjorck. *Numerical Methods*. Prentice-Hall, Englewood Cliffs, N.J., 1974.

[6] Van Emden Henson. *Fourier Methods of Image Reconstruction*. PhD thesis, University of Colorado at Denver, 1990.

[7] Van Emden Henson. Parallel compact symmetric FFTs. In *Vector and Parallel Computing: issues in applied research and development*, John Wiley and Sons, Inc., 1989.

[8] Eugene Isaacson and Herbert Bishop Keller. *Analysis of Numerical Methods*. John Wiley and Sons, New York, N.Y., 1966.

[9] Theodore J. Rivlin. *An Introduction to the Approximation of Functions*. Dover Publications, Inc., New York, N.Y., 1969.

[10] Paul N. Swarztrauber. Symmetric FFTs. *Mathematics of Computation*, 47:323–346, 1986.

[11] David M. Young and Robert Todd Gregory. *A Survey of Numerical Mathematics*. Volume 1, Dover Publications, Inc., New York, N.Y., 1972.

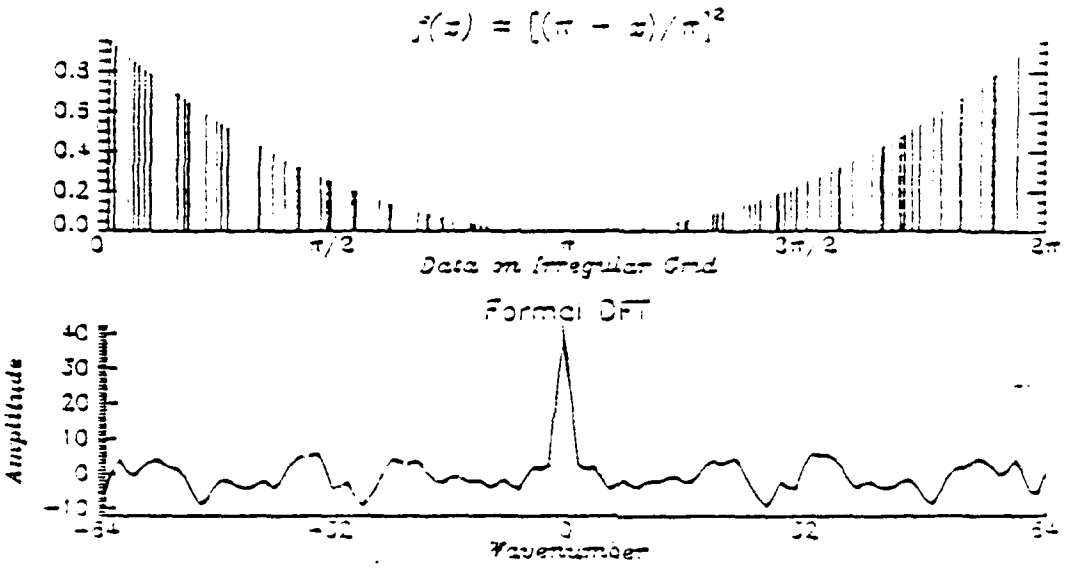

FIG. 1. The function $f(x) = [(\pi - x)/\pi]^2$ sampled on an irregular grid, and its formal DFT. Only the real part of the formal DFT is plotted.

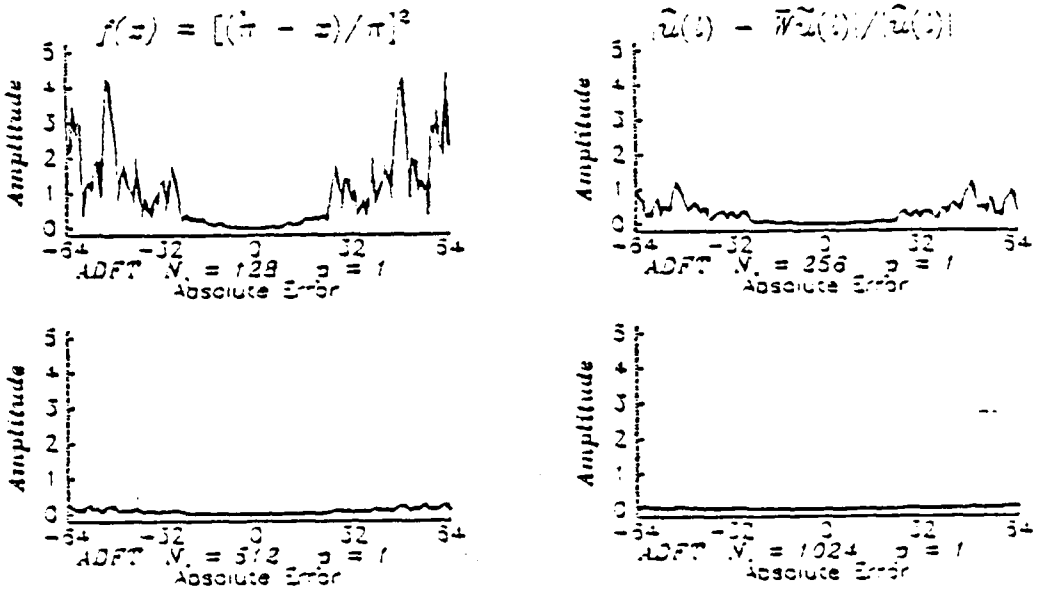

FIG. 2. The absolute value of the error of the $ADFT$, plotted as a function of wavenumber. Each graph corresponds to a different choice of N_s, while linear interpolation ($p = 1$) is used for all.

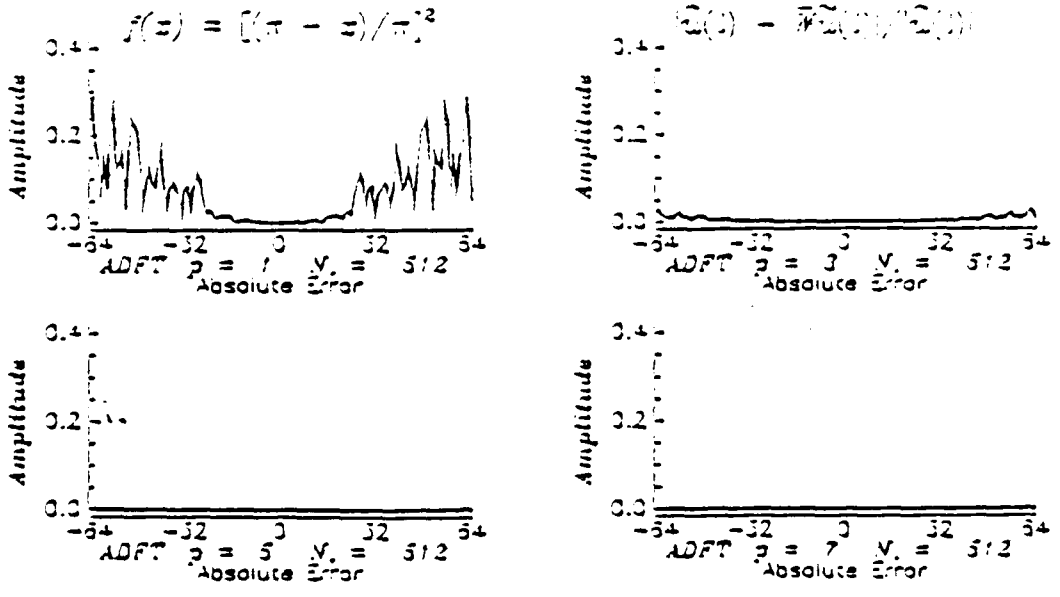

FIG. 3. The absolute value of the error of the $ADFT$, plotted as a function of wavenumber. $N_s = 512$ for each graph, while several values of p are used.

DISTRIBUTION LIST

DIRECTOR (2)
DEFENSE TECH. INFORMATION
CENTER, CAMERON STATION
ALEXANDRIA, VA 22314

DIRECTOR OF RESEARCH ADMIN.
CODE 81
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943

LIBRARY (2)
CODE 52
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943

NATIONAL SCIENCE FOUNDATION
WASHINGTON, D.C. 20550

DEPARTMENT OF MATHEMATICS
CODE MA
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943

ASST. PROFESSOR VAN EMDEN HENSON (10)
CODE MA/HV
DEPARTMENT OF MATHEMATICS
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943