DFTS ON IRREGULAR GRIDS:
THE ANTERPOLATED DFT

by
Van Emden Henson

Technical Report for Period
October 1990-March 1992

Approved for public release; distribution unlimited

Prepared for: Naval Postgraduate School
Monterey, CA 93943

92-25841
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
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^0, 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.
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 \( C \) 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

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

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 [1] 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

\[ 0 \leq x_0 < x_1 \ldots < x_{N-1} < X \]

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

\[ \hat{u}(\omega_l) = \sum_{j=0}^{N-1} u(x_j) e^{-i\omega_l x_j}, \quad \omega_l = \frac{2\pi l}{X}, \quad -M_1 \leq l \leq M_2, \]

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 \((-M_1 \leq l \leq M_2)\) 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 \(\|u\|_1 = X^{-1} \sum_{j=0}^{N-1} |u(x_j)|\). 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, \(\Omega_{X*}\), 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 \(y_r = (r - 1)h\), for \(r = 1, 2, \ldots, N_*\).

Suppose that the value of some function \(g(y_r)\) on the regular grid \(\Omega_{X*}\) 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 \(\Omega_{X*}\) must be located. Let these points be designated \(y_{k_{(j,0)}}, y_{k_{(j,1)}}, \ldots, y_{k_{(j,p)}}\). 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

\[
(3) \quad w_{i_{(j,0)}}(x_j) = \prod_{r \neq i} \frac{(x_j - y_{k_{(j,r)}})}{(y_{k_{(j,r)}} - y_{k_{(j,r)}})},
\]

and the interpolation of the function \(g\) to the gridpoints \(x_j\) is given

\[
g(x_j) \approx \sum_{r=0}^{p} w_{i_{(j,0)}}(x_j) g(y_{k_{(j,r)}}).
\]

Letting \(\hat{g}\) be the vector of function values \(g(y_r)\) and \(\hat{g}\) be the vector of interpolated values \(g(x_j)\), the interpolation may be written in matrix form

\[
\hat{g} = [I^*_{y}] \hat{g}^y
\]

where \(I^*_{y}\) is the \(N \times N_*\) interpolation matrix mapping a function on \(\Omega_{X*}\) to the gridpoints \(\{x_j\}\). The entries of this matrix are

\[
[I^*_{y}]_{mn} = \begin{cases} 
  w_{i_{(j,0)}}(x_j) & \text{if } k_{(j,r)} = m \\
  0 & \text{else} 
\end{cases}
\]

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

\[
\hat{u}(\omega_l) \approx \sum_{j=0}^{N-1} u(x_j) e^{-i\omega_l x_j}
\]
where
\[
\epsilon(\omega_1, x_j) = \sum_{i=0}^{p} w_i(x_j) e^{-i \omega_1 \lambda_i \tau_{ji}}.
\]

Let the column vector of exponential values \(e^{-i \omega_1 \lambda_i \tau_{ji}}\) be designated \(\epsilon_i\), and the vector of interpolated kernel values \(\epsilon(\omega_1, x_j)\) be denoted \(\epsilon_i\). Then this interpolation can be written
\[
\epsilon_i = [T \hat{T}] \epsilon_i.
\]

Notice, however, that \(\epsilon_i\) is to be used as the \(i\)th row of the matrix giving the kernel of the summation in (4). To compute \(\epsilon_i\) as a row vector, the adjoint of the matrix equation equation is needed, namely
\[
(\epsilon_i)^T = (T \hat{T})^T = (T^T) \hat{T}^T.
\]

Let us define the \(M\)-vector \(\hat{u} = \hat{u}(\omega_1)\), the \(N\)-vector \(\bar{u} = \bar{u}(x_j)\), and the matrices giving the kernels of equations (2) and (4) as \(W\) and \(\hat{W}\), respectively. Let the \(M \times N\) matrix whose \(i\)th row consists of \(e^{-i \omega_1 \lambda_i \tau_{ji}}\) for the \(N\) points on \(\Omega^i\) 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
\[
\hat{u} = W \bar{u}
\]
\[
\approx \hat{W} \bar{u} = W [T \hat{T}]^T \bar{u}.
\]

This notation can be simplified slightly by denoting the vector created by multiplying \(\bar{u}\) by \([T \hat{T}]^T\), as \(\hat{u}\). Since the matrix \([T \hat{T}]^T\) is the adjoint of the Lagrangian interpolation matrix, the process of computing \(\hat{u} = [T \hat{T}]^T \bar{u}\) has been dubbed interpolation. Then the approximation to the formal DFT is
\[
\hat{u} \approx W \hat{u},
\]
which we call the Interpolated 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
\[
\text{FFT} \{\hat{u}\} = \frac{1}{N} \sum_{t=0}^{N-1} \hat{u}_t e^{-i \omega_1 t / N} \quad \text{for } l = -N/2 + 1, -N/2 + 2, \ldots N/2.
\]

Recalling that \(h \approx X / N\), 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 \(\hat{u}(\omega_1)\).

Computing the ADFT, then, consists of two phases:
1. \(\hat{u}\) is computed from \(\bar{u}\) by interpolation: \(\hat{u} = [T \hat{T}]^T \bar{u}\).
2. \(\hat{u}\) is computed from \(\hat{u}\) 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 \( \tilde{u} = [I_{r_0}^T] \tilde{u} \), 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 \( \Omega_{r_0}^1 \) means that the denominators of \( w_n(x_j) \) are independent of \( j \). To compute the numerators, the product \( \prod_{m=0}^{p} (x_j - x_{k(j,m)}) \) 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 \( (x_m - x_{k(j,m)}) \) with the precomputed denominator \( \prod_{m \neq n} (x_{k(n,m)} - x_{k(j,m)}) \), 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 \( \Omega_{r_0}^1 \) 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 \( [I_{r_0}^T]^T \) is \( N_* \times N \) and the data vector \( \tilde{u} \) is \( N \times 1 \), so the computation of \( \tilde{u} = [I_{r_0}^T]^T \tilde{u} \) would be \( O(NN_*) \) if performed as a matrix-vector multiplication. There is, however, a much more efficient method. The index table \( k(j,n) \) can be stored along with the interpolation weights. For each \( x_j \) and for each \( n \), the value of \( k(j,n) \) is the index of the \( n^{th} \) interpolation neighbor that is used to interpolate from \( \Omega_{r_0}^1 \) 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 \( y_{k(j,\frac{p+1}{2})} \) and \( y_{k(j,\frac{p-1}{2})} \), while if \( p \) is even, then \( y_{k(j,\frac{p}{2})} \) is the closest gridpoint on \( \Omega_{r_0}^1 \) 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 \( y_{\tau} \in \Omega_{r_0}^1 \), \( 1 \leq \tau \leq N_* \)

2. For \( j = 0,1,\ldots,N - 1 \)
   For \( n = 0,1,\ldots,p \)
   Set \( \tilde{u}(y_{k(j,n)}) = \tilde{u}(y_{k(j,n)}) + w_n(x_j)U_j \).

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, \( \tilde{\mathbf{u}} \). 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 \( w_i(x_j) \), and the \( N \times (p + 1) \) integer array of indices, \( s(j,n) \).

If the operations of multiplication and addition are counted equally, and if the weights and indices are pre-stored, the operation count is

\[
C_1 N \log_2 N + 2N(p + 1)
\]

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 Interpolated DFT. One of the attractive features of the ADFT is that the interpolation is performed on the \( k \)-real, 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 \( x = \xi_0 + tb \), where \( h \) is the grid spacing, and \( t \in [0, p) \). The approximation to \( f(\xi_0 + tb) \) is the value of the Lagrangian interpolation polynomial \( P_t(\xi_0 + tb) \).

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

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

\[
|f(\xi_0 + tb) - P_t(\xi_0 + tb)| \leq \frac{|\tau_p(t)| h^{p+1} Q_{p+1}}{(p+1)!}
\]

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 \( \tau_p(t) \). This polynomial has been well-studied, and many results can be found in various numerical analysis texts, ([3], [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 $p = 3$ gives 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

$$\max_{t \in \{\pm \frac{1}{p+1}, \pm \frac{3}{p+1}\}} |\pi_p(t)| < \frac{1}{2^{p+1}}.$$ 

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

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

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 $\omega_M = (2\pi M_*)/N$. Therefore

$$\max_{x \in \{x_1, x_2, \ldots, x_p\}} \left| \frac{\partial^{p+1}}{(i\pi)^{p+1}} \left( e^{-i\omega_M^2} \right) \right| = (\omega_M)^{p+1}.$$ 

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

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

which is then used to obtain

$$|\hat{u} - \hat{W}\hat{u}| = |\sum_{j=0}^{N-1} u(x_j) e^{-i\omega_l x}| \leq \sum_{j=0}^{N-1} |u(x_j)|,$$

$$\leq (h \omega_M/2)^{p+1} \sum_{j=0}^{N-1} |u(x_j)|,$$

$$= (h \omega_M/2)^{p+1} N ||u||_1.$$ 

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

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

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 $\omega_l$, it then follows that

$$||\hat{u} - [\hat{W}\hat{u}]||_\infty \leq \left( \frac{M_* \pi}{N_*} \right)^{p+1} N ||u||_1.$$
where \( \| \cdot \|_\infty \) 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 \( \omega_M \) with \( \omega_l \) in the derivation, leading to

\[
|\hat{u}(\omega_l) - [W\hat{u}](\omega_l)| \leq \left( \frac{l\pi}{N} \right)^{l+1} N\|u\|_1 .
\]

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, \( |\hat{u} - W\hat{u}| \leq \epsilon\|u\|_1 \).

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

\[
\left( \frac{M_*}{N_*} \right)^{l+1} \leq \frac{\epsilon}{N} ,
\]

which may be written as

\[
N_* \geq M_* \left( \frac{N}{\epsilon} \right)^{l+1} .
\]

For a given formal DFT, the values of \( M_* \), \( N_* \), and \( \epsilon \) 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 fulfill 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 computing the ADFT with \( N_* \) points on the auxiliary grid \( \Omega_{N_*} \) and \( p \)-degree Lagrangian interpolation is

\[ O(N_* \log_2 N_* ) + O(N(p + 1)) . \]

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. FFTs 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 persuades 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 \((N \text{ and } M_* )\), and a prescribed error tolerance \( \epsilon \), the work in computing the ADFT to the required accuracy is

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

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

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 $\{(N_*, p) : N_\ast > M_\pi (N/\epsilon)^{1/(p+1)}\}$, and let $\partial S$ be that portion of the boundary of $S$ given by $\{(N_*, p) : N_\ast = M_\pi (N/\epsilon)^{1/(p+1)}\}$. 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 $\xi = M_\pi \left(\frac{N}{\epsilon}\right)^{1/(p+1)}$. Furthermore, $\xi < x_0$. Then since the partial derivative of the work function with respect to $N_\ast$ 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 $\partial S$ that requires less work, it is only necessary to seek a minimum from the points of $\partial S$. This can be done by parameterizing $N_\ast$ and $p$ as functions of a single variable.

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

Then on $\partial S$ we find that

$$N_\ast = M_\pi b \quad \text{and} \quad p + 1 = \log_\epsilon \left(\frac{N}{\epsilon}\right) .$$

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

$$W(b) = C_1 M_\pi b \log_\epsilon (M_\pi b) + 2N \log_\epsilon \left(\frac{N}{\epsilon}\right) .$$

and the problem is to minimize (15) subject to the constraint $1 < b \leq (N/\epsilon)$. Once $b$ is determined, the necessary values of $N_\ast$ 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

$$W(N_*, p) = C_1 N_\ast \log_\epsilon N_\ast + 2N(p + 1)$$

has a unique minimum, subject to the constraint

$$N_\ast \geq M_\pi \left(\frac{N}{\epsilon}\right)^{1/(p+1)} \quad \text{and} \quad 0 \leq p < \infty .$$
Proof: \( W(b) \) is continuous and differentiable with respect to \( b \) on \((1,N/\epsilon)\). Differentiating equation (15) yields

\[
W'(b) = K_1 \ln(K_2 b) - \frac{K_3}{b \ln(b)^2}.
\]

where

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

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

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

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 \( W(N/\epsilon, b_0) \), where

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

is the unique global minimum for \( W \) on \((0,\epsilon)\).

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

\[
\log \left( \frac{N}{\epsilon} \right) - 1
\]

will suffice.

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

\[
b \ln(b)^2 \ln(K_2 b) = \frac{K_3}{K_1}.
\]

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 \( N, M_{\pi}, \) and \( \epsilon \).
Table 1. Optimal parameters $N_*$ and $p$ computed for various problems.

<table>
<thead>
<tr>
<th>$N_*$</th>
<th>$M_*$</th>
<th>$\epsilon$</th>
<th>$N_*$</th>
<th>$p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>8</td>
<td>.1</td>
<td>48.7</td>
<td>7.7</td>
</tr>
<tr>
<td>32</td>
<td>32</td>
<td>.1</td>
<td>142.6</td>
<td>15.5</td>
</tr>
<tr>
<td>32</td>
<td>64</td>
<td>.1</td>
<td>257.6</td>
<td>22.2</td>
</tr>
<tr>
<td>32</td>
<td>8</td>
<td>.01</td>
<td>52.9</td>
<td>9.8</td>
</tr>
<tr>
<td>32</td>
<td>32</td>
<td>.01</td>
<td>150.1</td>
<td>19.1</td>
</tr>
<tr>
<td>32</td>
<td>64</td>
<td>.01</td>
<td>267.8</td>
<td>27.1</td>
</tr>
</tbody>
</table>

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 $2\pi$, the frequencies $\omega_i$ are just the integers $l$, and the formal DFT is

$$
\hat{u}(l) = \sum_{j=0}^{N-1} u(x_j) e^{-i \omega_i l}, \quad -64 \leq l \leq 64.
$$

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 $|\hat{u}(l) - [W \hat{u}](l)|$, 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 $(\frac{2\pi}{N_*})^p$ decreases as $p$ or $N_*$ are increased. Table 2 gives both the infinity and $L_2$ norms of the error $|\hat{u}(l) - [W \hat{u}](l)|$ 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 0.1 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.

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

| $N_*$ | $p$ | $||Error||_\infty$ | $||Error||_2$ |
|-------|-----|---------------------|----------------|
| 256   | 1   | 1.20663             | 0.334861       |
| 256   | 3   | 0.357889            | 0.116050       |
| 256   | 5   | 0.144478            | 0.039136       |
| 256   | 7   | 0.061305            | 0.01493        |
| 512   | 1   | 0.290501            | 0.109943       |
| 512   | 3   | 0.029351            | 0.008315       |
| 512   | 5   | 0.003360            | 0.00017        |
| 512   | 7   | 0.000419            | 9.66385e-05    |
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 \( \hat{u}(\omega) = \overline{u(-\omega)} \), because the vector \((T_x^*)^T \hat{u}\) 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 Weizmann 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


The function $f(z) = [(\pi - z)/\pi]^2$ sampled on an irregular grid and its formal DFT. Only the real part of the formal DFT is plotted.
The absolute value of the error of the ADPT, plotted as a function of wavenumber. Each graph corresponds to a different choice of \( N \), while linear interpolation \( p = 1 \) is used for all.

\[
f(z) = \left( \frac{\pi - z}{\pi} \right)^2
\]

\[
\frac{\tilde{a}(z)}{\tilde{b}(z)} = \tilde{a}(z_i) / \tilde{b}(z_i)
\]

\[
\frac{\tilde{a}(z)}{\tilde{b}(z)} = \tilde{a}(z) / \tilde{b}(z)
\]

The absolute value of the error of the ADFT, plotted as a function of wavenumber. \( N = 512 \) 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