Discrete Fourier transforms and related topics

Paul N. Swarztrauber*

June 26, 2003

Much of the theory and analysis for computations on the sphere can be best understood in the context of comparable computations in Cartesian geometry where Fourier theory and analysis are applicable. In addition, Fourier analysis is part of harmonic analysis.

Therefore we begin our study of spherical computations with a review Fourier analysis, which facilitates the understanding of comparable topics in spherical geometry.

TOPICS

*National Center for Atmospheric Research, which is supported by the National Science Foundation, P.O. Box 3000, Boulder, CO 80307

SPECTRAL ACCURACY

Given $f(\theta)$ on the interval $[0, 2\pi]$ and define

$$a_n = \frac{1}{\pi}\int_0^{2\pi} f(\theta)\cos n\theta d\theta \quad ; \quad b_n = \frac{1}{\pi}\int_0^{2\pi} f(\theta)\sin n\theta d\theta \tag{1}$$

then for piecewise smooth $f(x)$

$$f(x) = \frac{1}{2}a_0 + \lim_{L\to\infty}\sum_{n=1}^{L}(a_n\cos n\theta + b_n \sin n\theta) \tag{2}$$

From a computational viewpoint the rate of convergence is fundamental. Integrating by parts:

$$a_n = -\frac{1}{n\pi}\int_0^{2\pi} f'(\theta)\sin n\theta d\theta\,, \tag{3}$$

$$b_n = \frac{1}{n\pi}[f(0) - f(2\pi)] + \frac{1}{n\pi}\int_0^{2\pi} f'(\theta)\cos n\theta d\theta\,. \tag{4}$$

If $f^{(k)}(\theta)$ is continuous and periodic then repeated integration by parts yields kth order algebraic convergence.

$$|a_n| \leq \frac{C(k)}{n^{k+1}} \quad ; \quad |b_n| \leq \frac{C(k)}{n^{k+1}} \tag{5}$$

If all derivatives are continuous and periodic then a_n and b_n must decrease faster than any inverse power of n (say Ce^{-n}). This attribute is called spectral accuracy, which in practice is rarely, if ever, achieved.

NONPERIODIC FUNCTIONS

Assume we have a very smooth function that is not periodic. For example $f(x) = x$ on the interval $[0, 2\pi]$

The trigonometric series representation is periodic and therefore converges to a periodic "sawtooth" or discontinuous function.

The rate of convergence $[\mathcal{O}(n^{-1})]$ is determined by the "sawtooth" periodic function and not by the smooth $f(x) = x$.

A thousand terms could be required to obtain 3 digits of accuracy and then only in double precision.

Under such circumstances spectral accuracy can still be obtained by switching to Chebyshev polynomials, for which a fast transform also exists.

Good Fourier theory reference

Georgi P. Tolstov, Fourier Series, Dover, New York,
ISBN 0-486-63317-9

Good Chebyshev reference

Gottlieb and Orszag, Numerical Analysis of Spectral Methods, SIAM, Philadelphia, 1977.

DISCRETE TRIGONOMETRIC SERIES REPRESENTATIONS

Assume we are given a tabulation f_i on a set of N **equally spaced** points x_j. The purpose of discrete Fourier analysis is to find a continuous approximation $\hat{f}(x)$ to $f(x)$, which is unknown except at the points x_j where $f(x_j) = f_j$. Then, one can approximate derivatives, integrals or any operation by formal application to the continuous trigonometric representation.

That is, we wish to determine coefficients a_n and b_n such that

$$\hat{f}(x) = \sum_{n=0}^{L} [a_n \cos 2\pi n \frac{x-a}{b-a} + b_n \sin 2\pi n \frac{x-a}{b-a}] \qquad (6)$$

satisfies $\hat{f}(x_j) = f_j$.

What is L?

Define $\delta x = (b-a)/N$, then at the points $x_j = j\delta x + a$, $j = 0, 1, \ldots, N-1$.

$$f_j = \sum_{n=0}^{L} [a_n \cos nj \frac{2\pi}{N} + b_n \sin nj \frac{2\pi}{N}] \qquad (7)$$

L is equal to the number of independent trigonometric functions that can be defined on the N points x_j.

THE DISCRETE TRIGONOMETRIC BASIS AND ALIASING

Trigonometric aliases:

$$\cos(n+\frac{N}{2})j\frac{2\pi}{N} = \cos[N-(\frac{N}{2}-n)]j\frac{2\pi}{N} = \cos(\frac{N}{2}-n)j\frac{2\pi}{N} \quad (8)$$

$$\sin(n+\frac{N}{2})j\frac{2\pi}{N} = \sin[N-(\frac{N}{2}-n)]j\frac{2\pi}{N} = -\sin(\frac{N}{2}-n)j\frac{2\pi}{N} \quad (9)$$

Any trig function corresponding to $m > N/2$ is identical (on the points) to one corresponding to $m \leq N/2$. That is, any function with $m > N/2$ has an alternate representation (or alias) in terms of one with $m \leq N/2$.

Therefore if we set $L = N/2$ the resulting discrete trigonometric basis includes all distinct trigonometric functions on the set of points $j\frac{2\pi}{N}$ for $j = 0, \ldots, N-1$.

The number of points x_j is equal to the number of coefficients a_n, b_n, which implies that the interpolation problem has a solution.

This is VERY different from harmonic analysis on the sphere where the number of basis functions is half the number of points on the sphere and the interpolation problem does not necessarily have a solution.

TRIGONOMETRIC INTERPOLATION

Now that the discrete set of basis functions is defined it remains only to compute the coefficients a_n, b_n.

Given f_j then a_n and b_n are computed from the following discrete approximations to their integral representations

$$a_n = \frac{1}{N} \sum_{j=0}^{N-1} f_j \cos nj \frac{2\pi}{N} \quad ; \quad b_n = \frac{1}{N} \sum_{j=0}^{N-1} f_j \sin nj \frac{2\pi}{N} \tag{10}$$

Rectangle rule (ordinarily inaccurate) is best possible for smooth periodic functions and provides spectral accuracy.

The resulting trigonometric representation is identical to f_j on x_j:

$$\hat{f}(x) = \frac{1}{2} a_0 + \sum_{n=1}^{N/2} (a_n \cos 2\pi n \frac{x-a}{b-a} + b_n \sin 2\pi n \frac{x-a}{b-a}) \tag{11}$$

IMPORTANT

The computed coefficients (**??**) have period N whereas the the exact coefficients trail off to zero.

Therefore the computed coefficients will likely have 100% relative error near $n = N/2$

INTERPOLATION ERROR

To examine interpolation error we artificially chose an "exact" function $f(\theta)$ with $2N$ terms in its series representation, which is double the number of functions in the discrete basis.

This simplifies exposition and makes evident the error in a_n and b_n for any $f(\theta)$. At θ_j the exact function is then

$$f(\theta_j) = \sum_{n=0}^{N} [a_n \cos nj\frac{2\pi}{N} + b_n \sin nj\frac{2\pi}{N}] \ . \tag{12}$$

Recall that on a discrete set of N points, wave numbers $(N/2+n)$ are indistinguishable from numbers $(N/2 - n)$, consequently

$$f(\theta_j) = \sum_{n=0}^{N/2-1} [(a_n+a_{N-n}) \cos nj\frac{2\pi}{N}+(b_n-b_{N-n}) \sin nj\frac{2\pi}{N}]. \tag{13}$$

Therefore instead of computing a_n and b_n a discrete Fourier analysis yields

$$\hat{a}_n = a_n + a_{N-n} \quad \text{and} \quad \hat{b}_n = b_n - b_{N-n} \tag{14}$$

Thus the error is "reflective" about $n = N/2$. For general $f(\theta)$ the error also "reflects" about $n = 0$, then again about $n = N/2$ and so forth. The sum total of these reflections is the interpolation error.

INTERPOLATION ERROR
observations

1. For general $f(\theta)$, $L = \infty$ and the error is the sum of an infinite number of coefficients that alias onto the interval $n \leq N/2$.

2. The error is a "reflection" of the a_n about the maximum wave number $N/2$. Hence the error in $a_{N/2-1}$ is at least $a_{N/2+1}$, which is likely comparable in magnitude and can induce 100% error.

3. The major source of numerical error in spectral models occurs with the differentiation of the trigonometric representation with coefficients $na_{N/2+1}$, which define the error in the derivative.

4. A useful trick here is to decrement the magnitude of the coefficients by the roundoff or other acceptable error.

INTERPOLATION ERROR

(continued)

5. If $f(\theta)$ and its derivatives are smooth then spectral accuracy dictates that a_n and b_n decrease exponentially. However the computed coefficients never decrease below truncation or roundoff error which is then multiplied by n when computing the approximate derivative of $f(\theta)$.

6. If $f(\theta)$ is not periodic, i.e. $f(0) \neq f(2\pi)$ then its Fourier representation is discontinuous at the endpoints and the convergence of a_n is very slow $\mathcal{O}(n^{-1})$ which is unacceptable.

7. The bottom line is that one does not even obtain the first $N/2$ coefficients a_n and b_n but rather the sum of all coefficients whose trig functions alias to wave number n.

ALIAS CONTROL

PDEs that model geophysical processes usually have nonlinear product (quadratic) terms such as $f(\theta)g(\theta)$ or $f(\theta)g'(\theta)$.

On N points trigonometric representations are limited to $N/2$ wave numbers. However the quadratic terms have N wave numbers so that half the coefficients will alias to the lower wave numbers.

If the decision is made to truncate all trig representations to $N/4$ wave numbers then the coefficients in the product terms can be computed exactly without aliasing.

The product term is then truncated from $N/2$ to $N/4$ wave numbers in agreement with the decision to limit all trig representations to $N/4$ wave numbers. In this manner alias error can be eliminated. Of course truncation error remains.

The truncation error provides an estimate of the interpolation error.

Spectral accuracy is maintained only if the resolution includes high order coefficents that are decreasing spectrally.

THE TWO-THIRDS RULE

The "reflective" aspect of aliasing implies that the coefficients a_n, b_n are computed without aliasing for $n \leq N/2 - m$ if $a_n = b_n = 0$ for $n \geq N/2 + m$. This observation leads to the two-thirds rule that increases the spectral resolution relative to the proceedure on the previous slide.

The reflective aspect of trigonometric aliasing permits the increase in the number of wave numbers from $N/4$ to $N/3$ without introducing any alias error in the coefficients.

The resolution in spectral space is then two-thirds the resolution in physical space and alias error is eliminated.

NOTE

On the sphere the two-thirds rule does not work for equally spaced latitudinal points because harmonics beyond the discrete basis alias onto all of the basis functions. That is, aliasing is not "reflective".

SUBROUTINE EZFFT

The good news is that given f_j the coefficients a_n and b_n can be computed using FFTPACK. In particular the EZFFT programs are simple to use but not quite as fast as the RFFT programs.

```
SUBROUTINE EZFFTF (N,F,AZERO,A,B,WSAVE)
```

The array WSAVE must be initialized by

```
SUBROUTINE EZFFTI (N,WSAVE)
```

The sequence f_j can be reconstructed from a_n and b_n by calling the backward transform or Fourier synthesis:

```
SUBROUTINE EZFFTB (N,F,AZERO,A,B,WSAVE) .
```

AZERO$= a_0$, A$(n) = a_n$, B$(n) = b_n$ F$(j) = f_j$. The coefficients satisfy

$$f_j = a_0 + \sum_{n=1}^{N/2}(a_n \cos nj\frac{2\pi}{N} + b_n \sin nj\frac{2\pi}{N}) \tag{15}$$

However the derivatives, integrals, etc. must be computed from the continuous form:

$$f(x) = a_0 + \sum_{n=0}^{N/2}(a_n \cos 2\pi n\frac{x-a}{b-a} + b_n \sin 2\pi n\frac{x-a}{b-a}) \tag{16}$$

This answers the most often asked question; namely, What do I do with the A's and B's and how are they scaled?

HINTS FOR USING EZFFT

Consider now the following hints for using subroutines `EZFFTI`, `EZFFTF`, and `EZFFTB` from `FFTPACK`. Basically it is a compilation of questions that have been asked over many years. It also provides hints for the other programs in `FFTPACK`.

1. When sampling periodic data, any sequence of N points may correspond to a phase shift of the set $f_0, \ldots, f_{N-1}$ and consequently the coefficients $\hat{a}_n$ and $\hat{b}_n$ computed by `EZFFTF` may differ from a_n and b_n, where f_0 is always assumed to be at the origin $x = a$.

2. Unlike the other transforms in `FFTPACK`, the `EZFFT` programs scale the coefficients, but at some additional cost compared to the `RFFT` programs.

3. It is conventional for FFT software to avoid entering redundant data and therefore $f_N = f_0$ is not included in the `F` array. Rather only $f_0, \ldots, f_{N-1}$

4. Subroutine `EZFFTI` initializes the `WSAVE` array for subsequent repeated use by `EZFFTF` and `EZFFTB`. It is called only once unless different values of N are used in which case different `WSAVE` arrays must be used.

HINTS FOR USING EZFFT
(continued)

5. Note the summation limits and the range of indices. The `F` array contains $(f_0, \ldots, f_{N-1})$. `AZERO` contains a_0. For even N, `A` contains $(a_1, \ldots, a_{N/2})$ and `B` contains $(b_1, \ldots, b_{N/2-1})$. For odd N, `A` contains $(a_1, \ldots, a_{(N-1)/2})$ and `B` contains $(b_1, \ldots, b_{(N-1)/2})$.

6. A rough estimate of the error in the trigonometric interpolation is given by the magnitude of the coefficients a_n and b_n for n near $N/2$. If they remain large, even as N is increased, it is likely that $f(\theta)$ or some low order derivative is discontinuous OR f_N has been included in the array `F`.

7. Where possible N should be selected as a product of small primes, preferably 2, 3, and 5. Otherwise the FFT can be quite inefficient. In the unlikely event N is restricted to a large prime then Bluestein's FFT should be used.

8. Do not pad a sequence with zeros because it then becomes discontinuous function whose spectral coefficients converge quite slowly and could therefore obscure the coefficients of $f(\theta)$ which might converge much faster.

W. L. Briggs, P. N. Swarztrauber, R. A. Sweet, V. E. Henson, and J. Otto, Bluestein's FFT for arbitrary N on the hypercube, *Parallel Computing*, **17**(1991), pp. 607-617

STAGGERED GRIDS
(PHASE SHIFTING)

Say we wish to interpolate to the center of the grid system or equivalently assume the model is posed on a staggered grid system.

$$f_{j+1/2} = \sum_{n=0}^{N/2} (a_n \cos n(j+1/2)\frac{2\pi}{N} + b_n \sin n(j+1/2)\frac{2\pi}{N}) \quad (17)$$

or

$$f_{j+1/2} = \sum_{n=0}^{N/2} [\hat{a}_n \cos nj\frac{2\pi}{N} + \hat{b}_n \sin nj\frac{2\pi}{N}) \quad (18)$$

where

$$\hat{a}_n = a_n \cos n\frac{\pi}{N} + b_n \sin n\frac{\pi}{N} \quad ; \quad \hat{b}_n = -a_n \sin n\frac{\pi}{N} + b_n \cos n\frac{\pi}{N} \, . \quad (19)$$

Therefore to obtain the coefficients for the shifted function one computes the unshifted coefficients a_n, b_n and then computes the shifted coefficients $\hat{a}_n$ and $\hat{b}_n$. $f_{j+1/2}$ can then be tabulated by calling `EZFFTB`.

THE DISCRETE COMPLEX FOURIER TRANSFORM (DFT)

The complex DFT is most frequently written as

$$c_n = \frac{1}{N} \sum_{j=0}^{N-1} f_j e^{-jn\frac{2\pi i}{N}} \qquad n = 0, \ldots, N-1 \, . \tag{20}$$

With inverse

$$f_j = \sum_{n=0}^{N-1} c_n e^{jn\frac{2\pi i}{N}} \quad , \qquad j = 0, \ldots, N-1. \tag{21}$$

However this is an aliased form that is not suitable for differentiation etc. The nonaliased form is

$$f_j = \sum_{n=0}^{N/2} c_n e^{jn\frac{2\pi i}{N}} + \sum_{n=1}^{N/2-1} c_{N-n} e^{-jn\frac{2\pi i}{N}} \, . \tag{22}$$

With $e^{jn\frac{2\pi i}{N}}$ replaced by $e^{2\pi i n\frac{x-a}{b-a}}$, this form is suitable for differentiation, interpolation, etc. because it contains the discrete basis with the smallest wave numbers n. The aliased forms provide the most convenient notation for algorithmic development.

THE RELATIONSHIP BETWEEN THE REAL AND COMPLEX TRANSFORMS

If f_j is real then replacing n with $n - N$ in the conjugate of the DFT yields

$$\bar{c}_{N-n} = \frac{1}{N} \sum_{j=0}^{N-1} f_j e^{j(N-n)n\frac{2\pi i}{N}} = c_n \; . \tag{23}$$

Substituting into the nonaliased form we obtain

$$f_j = c_0 + \sum_{n=0}^{N/2-1} [\mathrm{Re}(c_n) \cos jn\frac{2\pi}{N} - \mathrm{Im}(c_n) \sin jn\frac{2\pi}{N}] + (-1)^j c_{N/2} \tag{24}$$

Therefore the real trigonometric representation is related to the complex transform by

$$a_n = \mathrm{Re}(c_n) \quad \text{and} \quad b_n = -\mathrm{Im}(c_n) \; . \tag{25}$$

In this manner the complex FFT can be used to transform real sequences; however, most packages contain FFTs specifically for real sequences that are about twice as fast. e.g. `EZFFT` and `RFFT` in `FFTPACK`.

THE FAST FOURIER TRANSFORM

Separating even and odd indices, the DFT can be written

$$c_n = \sum_{j=0}^{N/2-1} f_{2j} e^{-k2j\frac{2\pi i}{N}} + \sum_{j=0}^{N/2-1} f_{2j+1} e^{-k(2j+1)\frac{2\pi i}{N}} \tag{26}$$

If we define

$$d_n = \sum_{j=0}^{N/2-1} f_{2j} e^{-jn\frac{2pi}{N/2}} \quad ; \quad g_n = \sum_{j=0}^{N/2-1} f_{2j+1} e^{-jn\frac{2pi}{N/2}} \tag{27}$$

then

$$c_n = d_n + e^{-n\frac{2\pi i}{N}} g_n \quad ; \quad c_{n+N/2} = d_n - e^{-n\frac{2\pi i}{N}} g_n \tag{28}$$

1. First compute d_n and g_n and then c_n, which requires the computation of two transforms of length $N/2$ or $2(N/2)^2$ operations, which is half the operations required by (**??**).
2. However d_n and g_n have the same form as c_n but with N replaced with $N/2$. Hence they can also benefit from the splitting algorithm to produce four DFTs of length $N/4$.
3. The recursive application of the splitting algorithm to successively shorter sequences yields the FFT, which requires $5N \log_2 N$ real floating point operations.

MULTIPROCESSOR FFTS

If N has factors $N = N_0 N_1$ then by using the standard index maps

$$n = i + jN_0 \quad ; \quad k = l + mN_1 \tag{29}$$

we can define the two-dimensional arrays

$$f_{i,j} = f_n \qquad i = 0, ..., N_0 - 1 \;\; ; \;\; j = 0, ..., N_1 - 1 \tag{30}$$

$$c_{l,m} = c_k \qquad l = 0, ..., N_1 - 1 \;\; ; \;\; m = 0, ..., N_0 - 1 \tag{31}$$

which when substituted into the DFT

$$c_k = \sum_{n=0}^{N-1} f_n e^{-nk\frac{2\pi i}{N}} \tag{32}$$

yields

$$c_{l,m} = \sum_{i=0}^{N_0-1} \omega_{N_0}^{im} \omega_N^{il} \sum_{j=0}^{N_1-1} f_{i,j} \omega_{N_1}^{jl} \tag{33}$$

where $\omega_N = e^{-i\frac{2\pi}{N}}$. Therefore $c_{l,m}$, and hence c_k, can be computed as two multiple transforms. First, N_0 transforms of length N_1 (multiplied by ω_N^{il}) can be computed simultaneously from

$$c_{i,l}^{(1)} = \omega_N^{il} \sum_{j=0}^{N_1-1} f_{i,j} \omega_{N_1}^{j} l. \tag{34}$$

MULTIPROCESSOR FFTS
(continued)

Next, N_1 transforms of length N_0 can be computed simultaneously from

$$c_{m,l}^{(2)} = \sum_{i=0}^{N_0-1} c_{i,l}^{(1)} \omega_{N_0}^{im} . \tag{35}$$

In practice this algorithm distributes well; however, an ordered transform requires three transpositions that can dominate compute time.

Years were spent parallelizing computations like above but in practice, communication dominates compute time.

P, N, Swarztrauber, Multiprocessor FFTs, *Parallel Computing*, **5**(1987), pp. 197-210.

P. N. Swarztrauber, Transposing arrays on multicomputers using de Bruijn sequences, *J. Parallel Distrib. Comput.*, **53**(1998) pp. 63-77.

P. N. Swarztrauber and S. W. Hammond, A comparison of optimal FFTs on torus and hypercube multicomputers, *Parallel Comput.*, **27**(2001) pp. 847-859.

SYMMETRIC FFTS

If the sequence f_j has some symmetry (real, sine, cosine, etc) then compute time can be reduced using special FFTs. Each symmetry cuts compute time in half, e.g. a real odd sequence can be transformed in a quarter of the time required by the complex FFT.

There are two kinds of symmetric transforms:

Algorithmic restructuring

The FFT itself is modified, e.g. the real FFT is developed by computing only half of the coefficients at each stage of the complex FFT because the other half are given by $c_{N-n} = \bar{c}_n$. Sounds simple but a real pain to implement. Yet even more difficult are algorithms for odd, even, and odd, even quarter wave transforms.

P. N. Swarztrauber, Symmetric FFTs, *Math. Comp.*, **B47**(1986), pp. 323-346.

Pre and postprocessing

The symmetric sequence is preprocessed before calling the real FFT and postprocessed to extract the desired coefficients.

SYMMETRIC FFTS
(continued)

As an example consider the pre and postprocessing FFT of an odd sequence

$$f_i \;=\; \sum_{n=1}^{N/2-1} c_n \sin ni\frac{2\pi}{N}\ . \tag{36}$$

First we preprocess the sequence f_i by computing a new sequence e_i

$$e_i = \frac{1}{2}(f_i - f_{N/2-i}) + \sin i\frac{2\pi}{N}(f_i + f_{N/2-i})\ . \tag{37}$$

Substituting trig representation for f_i we obtain

$$e_i = c_1 + \sum_{n=1}^{N/4-1}\left[(c_{2n+1} - c_{2n-1})\cos ni\frac{4\pi}{N} + c_{2n}\sin ni\frac{4\pi}{N}\right] - (-1)^i b_{N/2-1}\ . \tag{38}$$

Therefore e_i is a real periodic sequence with coefficients a_n and b_n that can be determined by a real transform.

Postprocessing then consists of computing

$$c_{2n} = b_n \quad ; \quad c_{2n+1} = c_{2n-1} + a_n \tag{39}$$

With this approach the transform of an odd sequence requires about the same time as a real sequence with N/2 elements or half the time required to extend to an odd periodic function and use the real FFT.

THE FAST FRACTIONAL FOURIER TRANSFORM

D. H. Bailey and P. N. Swarztrauber, Fast fractional Fourier transforms and applications, *SIAM Rev.*,**33**(1991), pp. 389-404.

Differs from the traditional DFT by the introduction of an arbitrary real parameter α in the exponent. The sampling interval can be different from the period of the data.

$$c_n = \frac{1}{N} \sum_{j=0}^{N-1} f_j e^{-2\pi i \; jn\alpha} \qquad n = 0, \ldots, N-1 \,. \tag{40}$$

- The FFFT was developed for the SETI project because it can lock onto a signal of varying frequency....
- It can also be used to identify lines (buildings) in a digitized video image.
- Computing segments of the DFT of a sparse sequence.
- Computing the DFT of a sequence whose length is a prime number (or any other size that is ill-suited for ordinary FFTs).
- Analyzing sequences with non-integer periodic components.
- Performing high-resolution trigonometric interpolation.

Facilitates computation of the Laplace transform.

D. H. Bailey and P. N. Swarztrauber, A fast method for the numerical evaluation of continuous Fourier and Laplace transforms, *SIAM J. Sci. Compt.*, **15**(1994), pp. 1105-1110.

FFTPACK

The good news is that the sine, cosine, and the quarterwave transforms have been implemented in FFTPACK.

```
1.   RFFTI    initialize  RFFTF and RFFTB
2.   RFFTF    forward transform of a real periodic sequence
3.   RFFTB    backward transform of a real coefficient array

4.   EZFFTI   initialize EZFFTF and EZFFTB
5.   EZFFTF   a simplified real periodic forward transform
6.   EZFFTB   a simplified real periodic backward transform

7.   SINTI    initialize SINT
8.   SINT     sine transform of a real odd sequence

9.   COSTI    initialize COST
10.  COST     cosine transform of a real even sequence

11.  SINQI    initialize SINQF and SINQB
12.  SINQF    forward sine transform with odd wave numbers
13.  SINQB    unnormalized inverse of SINQF

14.  COSQI    initialize COSQF and COSQB
15.  COSQF    forward cosine transform with odd wave numbers
16.  COSQB    unnormalized inverse of COSQF
```

FFTPACK
(continued)

```
17.  CFFTI     initialize CFFTF and CFFTB
18.  CFFTF     forward transform of a complex periodic sequence
19.  CFFTB     unnormalized inverse of CFFTF
```

FFTPACK is available via anonymous ftp by executing the command

ftp ftp.ucar.edu

Then enter “anonymous” for your name, and your email address for the password. Then follow this session:

```
ftp> cd dsl/lib/fftpack
ftp> mget *
  .
  .  answer y to each question
  .
ftp> quit
```

or from netlib.org where it has been accessed 400,000 times.

Good FFT Reference Text

William L. Briggs and Van Emden Henson, The DFT: An Owner's Manual for the Discrete Fourier Transform, Society for Industrial and Applied Mathematics, Philadelphia, 1995.

PROJECTS

Implement the restructured symmetric FFTs in FFTPACK

Implement Bluestein's FFT.

Implement FFTPACK 5.0