THE ASTROPHYSICAL JOURNAL, 343:874–887, 1989 August 15

STUDIES IN ASTRONOMICAL TIME SERIES ANALYSIS. III. FOURIER TRANSFORMS, AUTOCORRELATION FUNCTIONS, AND CROSS-CORRELATION FUNCTIONS OF UNEVENLY SPACED DATA

JEFFREY D. SCARGLE
Theoretical Studies Branch, Space Science Division, NASA-Ames Research Center
Received 1988 August 8; accepted 1989 January 24

ABSTRACT

This paper develops techniques to evaluate the discrete Fourier transform (DFT), the autocorrelation function (ACF), and the cross-correlation function (CCF) of time series which are not evenly sampled. The series may consist of quantized point data (e.g., yes/no processes such as photon arrival). The DFT, which can be inverted to recover the original data and the sampling, is used to compute correlation functions by means of a procedure which is effectively, but not explicitly, an interpolation. The CCF can be computed for two time series not even sampled at the same set of times. Techniques for removing the distortion of the correlation functions caused by the sampling, determining the value of a constant component to the data, and treating unequally weighted data are also discussed. FORTRAN code for the Fourier transform algorithm and numerical examples of the techniques are given.

Subject headings: analytical methods — BL Lacertae objects — numerical methods

I. THE PARADOX OF CORRELATION FUNCTIONS WITH UNEVENLY SAMPLED DATA

Correlation functions are useful time series analysis tools. They yield physical information such as the time scale of a process or the time delay between two related processes. But astronomical time series data are often unequally spaced in time, due to a variety of practical considerations. (The times may be irregular, or they be evenly spaced but with missing observations—"gaps.") Such unevenness produces a fundamental difficulty in the estimation of correlation functions, the resolution of which is the main point of this paper.

For data $X_n = X(t_n)$ sampled at evenly spaced times $t_n = (n-1)\Delta t$, $n = 1, 2, \ldots, N$ the traditional estimator of the autocorrelation function is

$$\rho_X(k) = (1/N) \sum_{n=1}^{N-k} X_n X_{n+k} \,. \tag{I.1}$$

This expression makes sense only if the sample times t_n are evenly spaced, since it can be thought of as a kind of vector dot product of X with X shifted in time by k. The times of the shifted data must match up with those of the unshifted data. Therefore the sampling interval must be constant and the lag k must be an integer multiple of this interval.

How should one estimate the ACF of unevenly sampled data? Possible approaches are to interpolate the data to even spacing and use equation (I.1), or to sum product-pairs $X(t_i)X(t_j)$ in bins of the lag $t_i - t_j$ (Mayo, Shay, and Riter 1974; Edelson and Krolik 1988). Gastner and Roberts (1975, 1977) circumvent the fact that the interval (t_n, t_{n+k}) is not a definite length of time, noting that statistically it does correspond to a fixed time interval—namely, k divided by the mean sampling rate. While these procedures may be satisfactory in some applications, they all produce some distortion and loss of information.

The goal of this work is a correlation function estimator which uses all of the information contained in unevenly spaced data. The proposed approach steps briefly into the frequency domain (computing the power spectrum) and returns to the time domain (computing the autocorrelation function with the Autocorrelation Theorem). While it does not explicitly interpolate, it can be thought of as effecting an implicit interpolation in the time domain. The basic tool of the computations is the discrete Fourier transform (§ II), which yields the power spectrum used in the computation of the autocorrelation function (§ III) and the cross-spectrum used to compute the cross-correlation function (§ IV). Examples using artificial data appear in all three of these sections. Section V exhibits correlation functions for some actual data on BL Lacertae—the prototype of a class of violently variable radio sources. The FORTRAN code for computing the discrete Fourier transform is given in Appendix A. Appendix B discusses the frequencies used in the inverse transformation. The remaining appendices treat an underlying constant component to the data, and unequally weighted data.

II. DISCRETE FOURIER TRANSFORM

This section presents an algorithm for the discrete Fourier transform (DFT) of unevenly sampled data. Later this transform will be used to estimate correlation functions, but it is of interest in its own right and in connection with power spectra.

Scargle (1982, hereafter Paper II) modified the classical definition of the DFT in order that the resulting power spectrum (or periodogram) of unevenly sampled data have the simple statistical behavior which obtains in the case of even sampling (Paper II, Appendix A), while maintaining time translation invariance (Paper II, Appendix B). In addition, spectral analysis using this estimator is equivalent to least-squares fitting of sine waves to the data (Paper II, Appendix C). Paper II dealt with power spectra, so the phase of the Fourier transform was unimportant; the present work differs slightly in correctly treating the complex phase of the transform.

Press and Teukolsky (1988) give an informative discussion of the beneficial properties of this periodogram, as well as a FORTRAN algorithm that uses a recurrence technique to gain a factor of 3 in speed. Further improvement is obtained with Press and Rybicki's (1989) clever $N \log N$ algorithm.

Appendix A of Paper II contains a minor error: the derivation given for equation (A9) tacitly assumes that C and S, defined in equations (A3) and (A6), respectively, are uncorrelated, which in general is not true. However, condition (C5) assures that they are uncorrelated, and therefore all of the results of Paper II are unaffected. I am grateful to Donald Percival for calling this point to my attention.

a) *Estimator*

Define the Fourier transform of $\{X(t_n), n = 1, 2, \ldots, N\}$ to be

$$\mathrm{FT}_X(\omega) = F_0 \sum (AX_n \cos \omega t'_n + iBX_n \sin \omega t'_n) , \quad \text{(II.1)}$$

where

$$F_0(\omega) = (N/2)^{1/2} \exp(-i\omega t_1) , \quad \text{(II.2)}$$

$$A(\omega) = (\sum \cos^2 \omega t'_n)^{-1/2} ; \quad B(\omega) = (\sum \sin^2 \omega t'_n)^{-1/2} , \quad \text{(II.3)}$$

$$t'_n = t_n - \tau(\omega) , \quad \text{(II.4)}$$

and

$$\tau(\omega) = (1/2\omega) \tan^{-1} (\sum \sin 2\omega t_n / \sum \cos 2\omega t_n) . \quad \text{(II.5)}$$

The only difference from the definition given in Appendix A of Paper II is the phase factor $\exp(-i\omega t_1)$ in equation (II.2). Without this factor $\mathrm{FT}_X(\omega)$ would be invariant under the transformation $t_n \to t_n + T_0$, since $t_n - \tau$ is invariant as discussed in Appendix B of Paper II. But the Fourier transform should experience a (complex) phase shift of $\exp(-i\omega T_0)$ under such a time translation. The choice of t_1 is somewhat arbitrary, but corresponds to a specification of the location of the fiducial origin of time. Other choices (such as the average of the times) are possible, but correspond to nonstandard locations of the origin. Inspection of equations (II.1)–(II.5) shows that the order of the t_n's is irrelevant in the definition of $\mathrm{FT}_X(\omega)$.

b) *Computing the DFT*

The following problems arise in the computation of the DFT with the above formulae. (i) Expression (II.5) for τ and the imaginary part of equation (II.1) are undefined for $\omega = 0$. (ii) The imaginary part of (II.1) is undefined at the Nyquist frequency for the case of even sampling. (iii) Equations (II.3) and (II.5) have ambiguities, due to (a) the sign ambiguity of the square root; (b) the multi-valued nature of the arctangent. (iv) For larger ω the arguments of the functions sin, cos, and exp are larger than standard computer routines can handle.

These difficulties are treated as follows:

Problem (i).—All terms at zero frequency are evaluated using the well defined limits as $\omega \to 0$, namely $\tau(0) = (\sum t_n)/N$, and $\mathrm{FT}(0) = (\sum X_n)/N^{1/2}$.

Problem (ii).—This problem can be ignored if the spacing of the time points will always be uneven. Otherwise it is probably best to branch to an FFT routine if the spacing passes a test for evenness. Nevertheless, it is of some interest to construct an algorithm valid no matter what the spacing. An 0/0 ambiguity in the imaginary part of (II.1) can be removed using L'Hopital's rule—differentiating the numerator (from eq. [II.1], omitting B) and the denominator ($1/B$ in eq. [II.3]) with respect to ω. The resulting denominator goes to infinity (giving zero for the imaginary part of FT), unless the crossproduct expression (called "CROSS" in the code) vanishes. See Appendix A for the details of the implementation of this procedure, which was tested on evenly spaced data (see Fig. 1*e*).

Problem (iii).—These ambiguities can be resolved by starting from the values at $\omega = 0$, computed as indicated in (i), and imposing continuity on the DFT as a function of ω. However, it turns out that ambiguity (b) produces just a sign error in the terms in (II.1). The result is a sort of cancellation of (*a*) and (*b*). Numerical studies have verified that, in at least the cases investigated, simply ignoring the ambiguities gives the same results as does imposing continuity.

Problem (iv).—The arguments are evaluated modulo an integer multiple of 2π.

No other numerical problems were uncovered in the present studies, but there may be kinds of data which cause difficulties. The reader is advised to use caution, especially in cases of unusual sampling. The above comments regarding (iii) are made with some caution. There may be some kinds of data for which this result is not true. I recommend checking the validity of the transform by comparing the original data with the inverse transform (see § II*c*). If this gives unsatisfactory results, one should evaluate the transform at a more finely spaced grid of frequencies; any abrupt sign reversals of the terms of equation (II.1) would indicate that the ambiguities were not correctly resolved.

Appendix A presents FORTRAN code that computes the DFT for arbitrary sampling.

c) *Inverting the DFT*

Inversion back to the time domain can be accomplished using standard techniques, the FFT in particular, by evaluating the DFT at the usual evenly spaced frequencies. Appendix B discusses some details of this inversion. The three points which are different from what might be expected are (1) the fundamental frequency is calculated using a period slightly larger than the range which the samples span, (2) the fundamental frequency is further reduced by a factor of 2, in order to eliminate wraparound, and (3) the maximum or Nyquist frequency is to be regarded as a smoothing parameter which can be adjusted depending on the use that will be made of the transform.

d) *Example Computations*

Figure 1 compares a specific randomly sampled time series with the inversion of its Fourier transform. The sampling was Poisson: the points were randomly placed on the time axis over the interval (0, 1) with a uniform probability distribution. (This corresponds to a Poisson distribution for the intervals between adjacent points.) The value of X at each of these random samples was also random, uniformly distributed from 0 to 1. The sample mean was removed from X prior to analysis. The raw data for a particular realization of this process are shown as triangular symbols in the various panels of this figure, each of which illustrates a particular feature of the procedure.

Figure 1*a* shows a set of 16 such randomly sampled random data points, analyzed using the techniques described above. The only free parameter is $\omega_{\max}$, the maximum frequency out to which the transform is evaluated in computing the inverse. This plot shows the results for the standard choice for this smoothing parameter, i.e., π divided by the mean sampling interval. As can be seen in the figure, this choice roughly corresponds to no smoothing of the data. The resulting inverse DFT (which, of course, is real) is plotted as a solid line and is reasonably close to the data points.

The inversion is not exact because some of the information

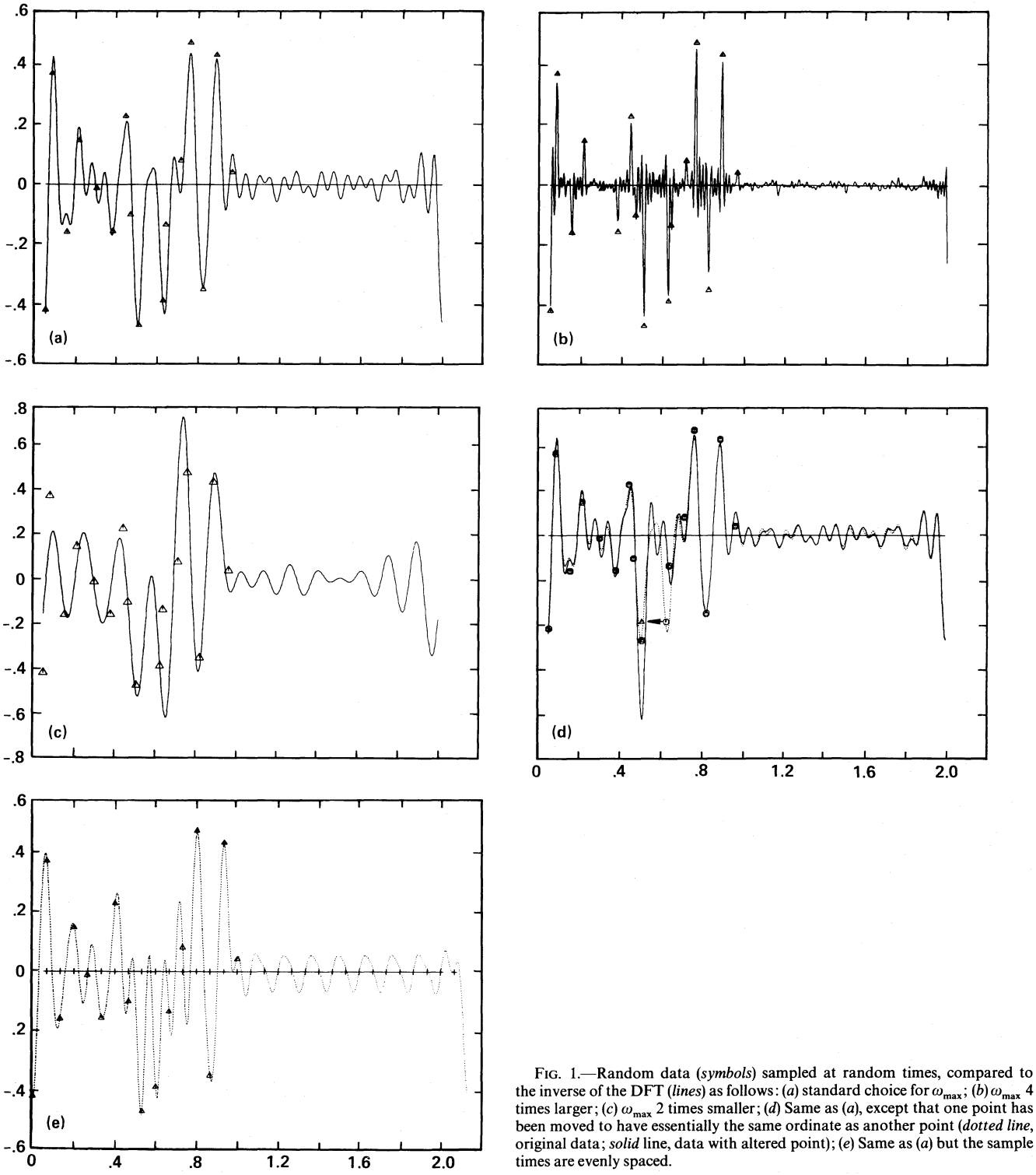

FIG. 1.—Random data (*symbols*) sampled at random times, compared to the inverse of the DFT (*lines*) as follows: (*a*) standard choice for ω_{max}; (*b*) ω_{max} 4 times larger; (*c*) ω_{max} 2 times smaller; (*d*) Same as (*a*), except that one point has been moved to have essentially the same ordinate as another point (*dotted line*, original data; *solid* line, data with altered point); (*e*) Same as (*a*) but the sample times are evenly spaced.

in the original data at high frequencies (corresponding to the sample times which are closer together than average) is not used. The period of the inversion is twice as long as the sampling range due to the wrap-around removal procedure, equation (B9). The "unsampled data" in the interval (1, 2) is, to a fair approximation, represented by the mean value (zero, in this case).

Figure 1*b* shows what happens if $\omega_{\max}$ is chosen to be relatively high: the inverse tends to zero between the samples, approaching a series of δ functions. Note that the locations of the δ-functions reproduce the sample times, a feature that may be of use in some applications.

Figure 1*c* shows the smoothing effect if $\omega_{\max}$ is smaller than the standard value. As expected for such purely random data, this smoothing degrades the quality of the fit to the data, but in many applications it can be useful.

Figure 1*d* shows an interesting result: our DFT treats data points which are sufficiently close together in time as a single point with amplitude equal to the sum of the individual amplitudes. One of the data points analyzed in Figure 1*a* was moved to essentially the same time as another point (as indicated by the arrow in the figure). The two inversions are similar in regions away from the moved point, but near it the amplitude of the inversion is approximately the sum of the two cotemporal data values.

Figure 1*e* shows that the algorithm works for the singular case of evenly spaced data. The algorithm in Appendix A contains special coding for this case. If one is sure that the data spacing will never be even, then this coding can be dispensed with.

III. AUTOCORRELATION FUNCTION

The autocorrelation function (ACF) is a measure of how closely a quantity observed at a given time is related to the same quantity at another time. It measures the degree of resemblance of the signal with itself as time passes, to use signal processing terminology. The ACF can also be viewed as a measure of the predictability of a process or signal, based on past data. Prediction of a process at time t into the future is good as long as the value of the correlation function at "lag" t is large. As soon as the correlation function becomes small the similarity of the signal to its previous history disappears, and the signal becomes unpredictable.

a) Definition

The autocovariance of a zero-mean random process X is defined as the expectation of the product of the values of X observed at times separated by the lag t:

$$\rho_X(t) = \langle X(t')X(t' + t)\rangle \,, \tag{III.1}$$

where $\langle \ldots \rangle$ is the ensemble average (over realizations of X). The autocorrelation function (ACF) is defined as the autocovariance normalized to unity at $t = 0$. The ACF of a stationary ergodic process can be computed as an average over t', and depends on only the difference between the two times, i.e., the lag t.

The traditional estimator of $\rho_X(t)$, given a finite set of evenly spaced observations, $\{X(t_n), n = 1, 2, \ldots, N\}$, is (e.g., Jenkins and Watts 1968; Box and Jenkins 1970):

$$\rho_X(k) = (1/N) \sum_{n=1}^{N-k} [X(t_n) - \bar{X}][X(t_{n+k}) - \bar{X}] \,. \tag{III.2}$$

It is often emphasized that the sample mean $\bar{X} = (1/N) \sum X_n$ must be subtracted as indicated here, but Appendix C shows that this is not always desirable.

The use of the factor $1/N$ in this equation, instead of the correct $1/(N - k)$ (to make equation [III.2] an average of $N - k$ terms) is a curiosity. Apparently the incorrect factor is used because it yields an estimator with a smaller variance (e.g., Jenkins and Watts 1968; Box and Jenkins 1970). In practice, it is argued, we are interested primarily in lags $k \ll N$ anyway, and the reduced variance of equation (III.2) is worth the small error (bias) which results. [The estimator is on the average too small by the factor $N/(N - k)$, which becomes significant only at large lags $k \sim N$.] This reduction in variance simply by making ρ smaller seems arbitrary, and in addition the presence of a bias can be devastating in many applications. On the other hand, the use of $1/N$ assures that the autocorrelation is "nonnnegative definite" (Brockwell and Davis 1987, § 1.5). The discussion below in § III*d* sheds more light on the role of this factor.

b) Estimator

The Autocorrelation Theorem (Papoulis 1962; Bracewell 1965) states that the power spectrum of a stationary random process is the Fourier transform of its autocorrelation function. Therefore *the ACF can be estimated by computing the inverse Fourier transform of the square of the complex absolute value of the Fourier transform*, since the latter is an estimate of the power spectrum. The cornerstone of this paper is the implementation of this relationship for unevenly spaced data.

Specifically, the discrete Fourier transform $\mathrm{FT}_X(\omega)$ is calculated as in the previous section, the power spectrum obtained from

$$P_X(\omega) = |\,\mathrm{FT}_X(\omega)|^2 \tag{III.3}$$

as in Paper II, and the ACF estimated with

$$\rho_X(t) = \mathscr{F}^{-1}[P_X(\omega)] \,, \tag{III.4}$$

where $\mathscr{F}^{-1}$ is the inverse Fourier transform (see Appendix B).

c) Removing the Sampling Distortion

The ACF estimate in equation (III.4) is altered by the sampling, in a way that enhances one problem and diminishes another. Leakage of power to nearby frequencies ("sidelobes") is enhanced by uneven sampling, while aliasing (the leakage of high-frequency power to low frequencies) is diminished.

There is a simple way to partially remove the effects of sampling from the ACF. Scott (1976) proved that the ACF of the observed process, $\rho_X(t)$, and the true (theoretical) ACF $\rho_X^{\mathrm{true}}(t)$, are related by

$$\rho_X(t) = \rho_X^{\mathrm{true}}(t)\rho_S(t) \,. \tag{III.5}$$

where $\rho_S(t)$ is the ACF of the sampling, considered as a random process. Equation (III.5) is the time domain analog of the frequency space convolution relation: observed quantity = true quantity * window function.

Scott does not provide a practical method for estimating ρ_S. I propose to estimate the transform of the sampling by setting $X_i = 1$ in equation (II.1), yielding

$$\mathrm{FT}_S(\omega) = F_0 \sum [A \cos \omega t'_n + iB \sin \omega t'_n] \tag{III.6}$$

and then by analogy with equation (III.4)

$$\rho_S(t) = \mathscr{F}^{-1}[|\,\mathrm{FT}_S(\omega)|^2] \,. \tag{III.7}$$

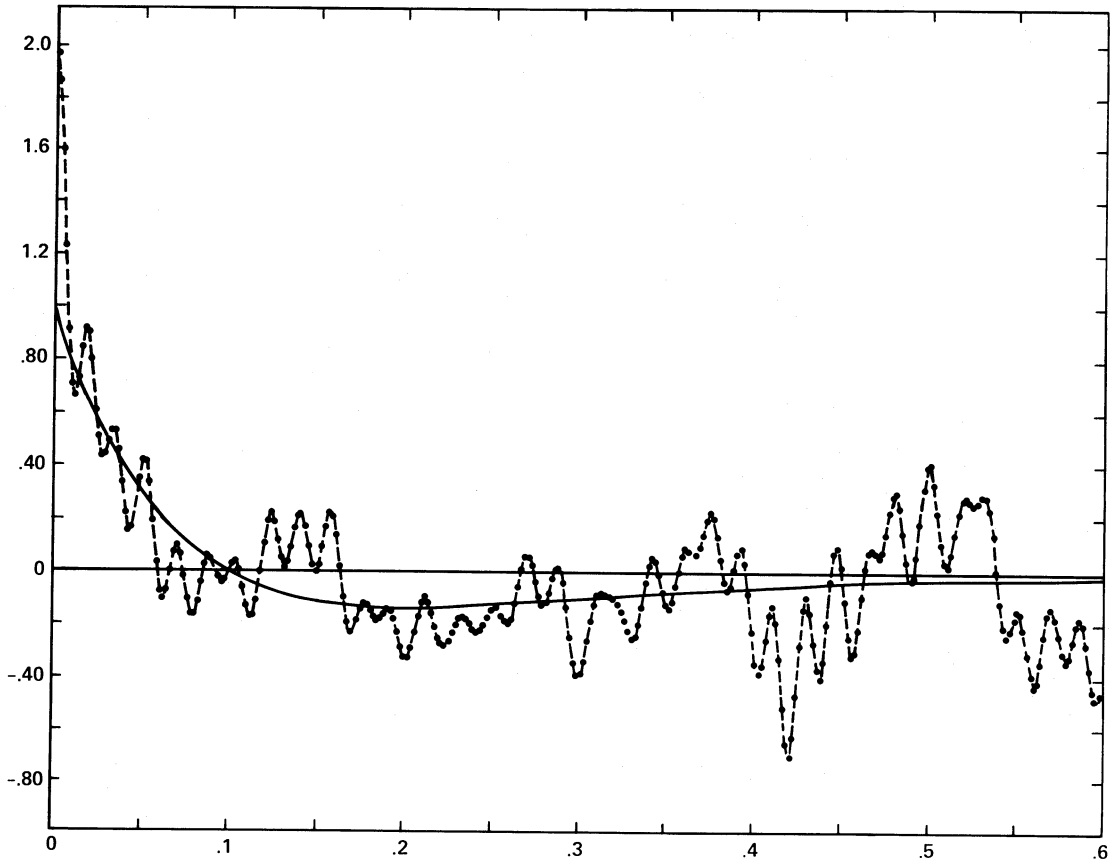

FIG. 2.—Autocorrelation for an artificial shot noise process (*symbols and dashed line*) compared with the theoretical ACF (*solid line*)

As noted in Appendix D of Paper II, functions defined in this way are not authentic "window functions"; i.e., we have not proved that equation (III.5) holds with this definition of $\rho_S(t)$. However the numerical experiments reported below show that the following estimate for the ACF removes at least some of the sampling distortions:

$$\rho_X^{\text{true}}(t) = \rho_X(t)/\rho_S(t) \, . \qquad \text{(III.8)}$$

There are obviously problems when $\rho_S(t)$ vanishes or is small, but often this does not happen for the lags of interest. Typically ρ_S is significantly greater than zero until t becomes relatively large.

d) Example: A Shot Noise Process

This section demonstrates the technique using artificial data from a moving average, or generalized shot noise, process (Appendix C and Scargle 1981, hereafter Paper I) in the form

$$X(t) = C(t) * R(t) + N(t) \, , \qquad \text{(III.9)}$$

with a pulse shape given by

$$C(t) = \begin{cases} +\exp\,(-t/a) (t \geq 0) \\ -\exp\,(+t/a) (t < 0) \end{cases} . \qquad \text{(III.10)}$$

The additive noise $N(t)$ represents observational errors. This pulse shape has a discontinuity at $t = 0$, but since it has zero area the problems discussed in Appendix C are avoided. The random variable $R(t)$ was taken to be a Poisson process, consisting of randomly occurring pulses all with the same amplitude.

Figure 2 compares the estimated ACF with the exact autocorrelation [easily shown to be $(1 - t/a) \exp(-t/a)$]. The agreement is good, although there is considerable scatter in the points for the estimate. By sampling the same process evenly it can be shown that this scatter is not due to the unevenness of the sampling, but to the finiteness of the realization and of the number of pulses sampled.

The sampling correction in Figure 2 is small, as expected because the sampling is moderately uniform. In fact, the effect of the correction is mainly to amplify the uncertainty in the estimated autocorrelation, because it amounts to dividing by a small number at relatively large lags. Indeed, *whether or not one applies the sampling correction factor is—on the average—equivalent to whether one uses* $1/(N - k)$ *or* $1/N$ *as the normalization factor in the definition of the ACF*, as discussed in § III*a*.

IV. CROSS-CORRELATION FUNCTION (CCF)

The cross-correlation function measures how closely two different observables are related to each other at the same or differing times. It also measures how well one process can be predicted based on observations of the other. Note that since we don't know which is the cause and which is the effect, this "predictability" can refer to either earlier or later times.

a) Definition

The cross-correlation function of two variables $X(t)$ and $Y(t)$ is

$$\rho_{XY}(t) = \langle X(t')Y(t' + t) \rangle \qquad \text{(IV.1)}$$

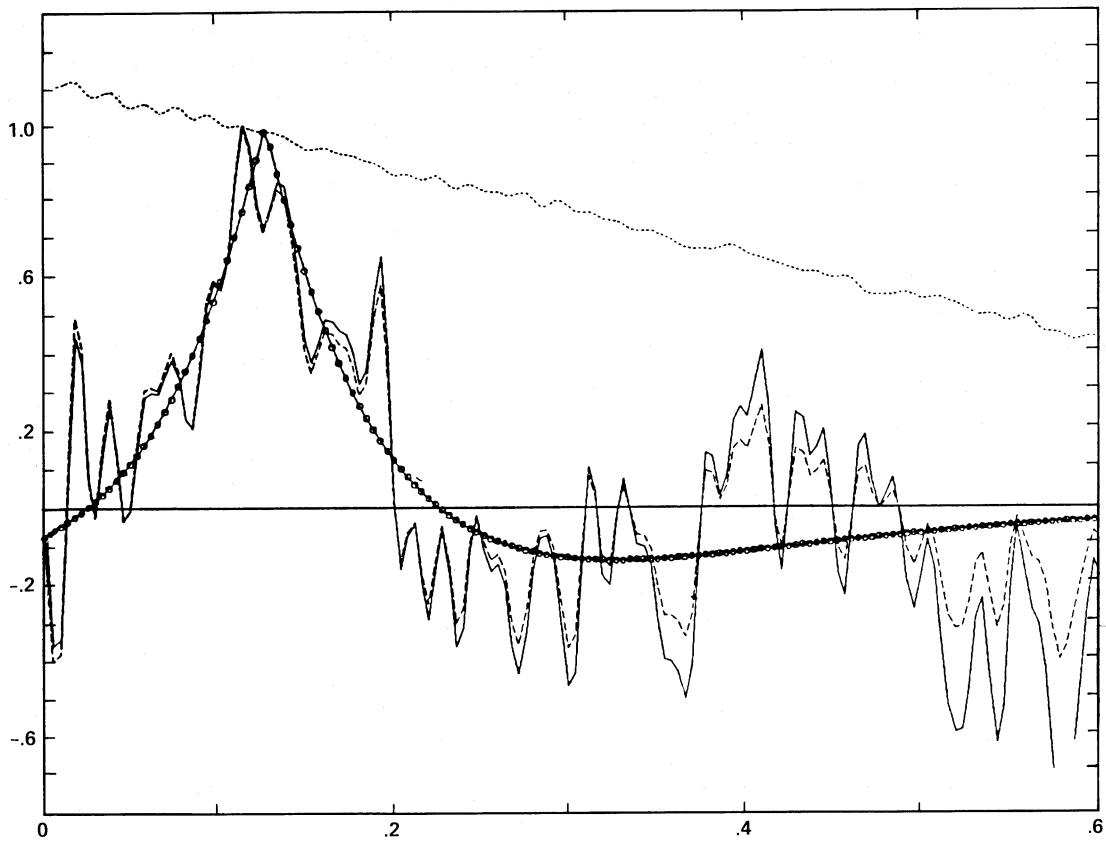

FIG. 3.—Cross-correlation for an artificial shot noise process, both with (*solid line*) and without (*dashed line*) sampling correction, compared to the theoretical cross-correlation function (*symbols and solid line*). The dotted line is the sampling correction.

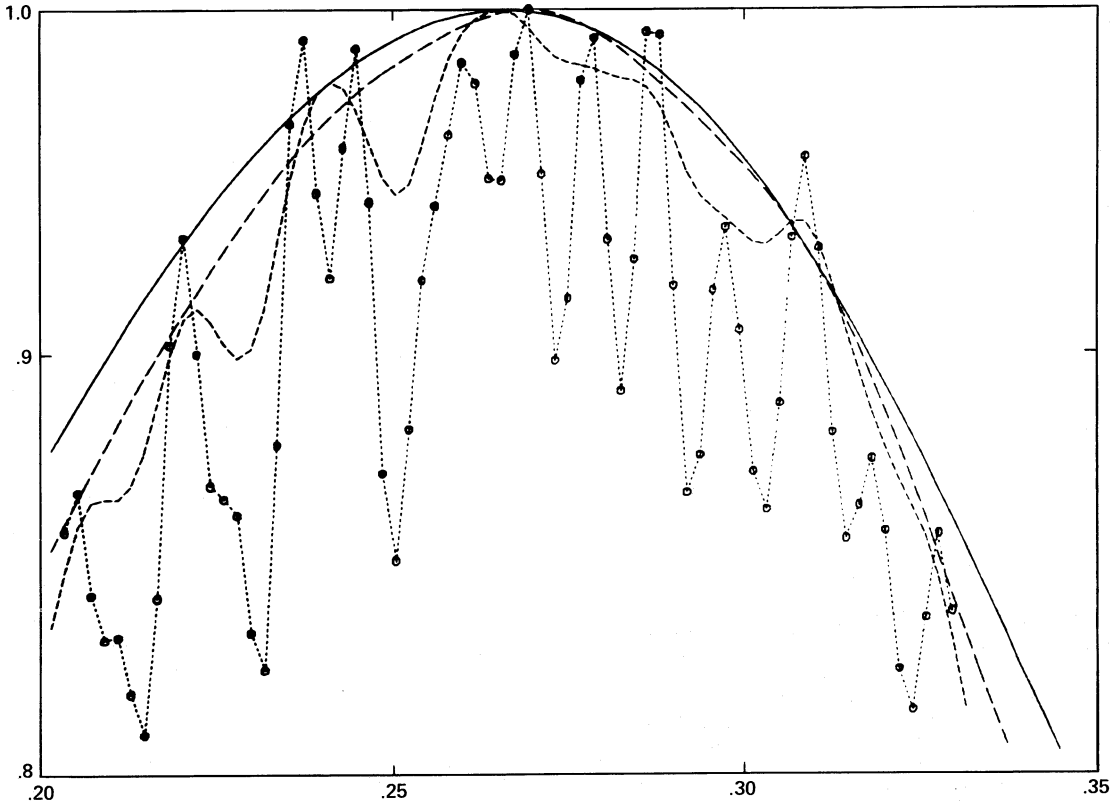

FIG. 4.—Cross-correlation for artificial point data, from a simulation of the Multichannel Astrometric Photometer: unsmoothed (*symbols and dotted line*) and with various degrees of smoothing.

(Jenkins and Watts 1968); X and Y are assumed to have zero mean. As with the ACF, the structure of the classical estimator

$$\rho_{XY}(k) = (1/N) \sum X(t_n)Y(t_{n+k}) \qquad \text{(IV.2)}$$

demands that the samples of X and Y be evenly spaced and identical, and that the lag be an integer multiple of the sampling interval (see § I). We again temporarily go to the frequency domain and use the relation that the cross-correlation function is the inverse Fourier transform of the cross-spectrum (Jenkins and Watts 1968, § 8.3.2).

b) Estimator

Specifically, the cross-spectrum is defined as

$$P_{XY}(\omega) = \mathrm{FT}_X(\omega)\mathrm{FT}^*_Y(\omega) , \qquad \text{(IV.3)}$$

and the cross-correlation function is then

$$\rho_{XY}(t) = \mathscr{F}^{-1}[P_{XY}(\omega)] . \qquad \text{(IV.4)}$$

(Appendix B discusses the frequencies to use in this inversion.) The CCF is readily evaluated for any samplings of X and Y. They need not be evenly spaced. They need not be the same. The sampling of X and Y need not even overlap! The CCF is determined only for lags which shift the two sample intervals so that they overlap significantly. For other lags the formula give garbage.

It is well known that observational errors produce a spike in the ACF at zero lag. Similarly *correlated errors in X and Y produce a zero-lag spike in the CCF*. It may be difficult to distinguish this artifact from a "real" effect; namely, simultaneous rapid variations in X and Y.

Much of the discussion for the ACF on sampling distortion, constant components, and weighted data (§ III and the Appendices) applies here and will not be repeated.

c) Example: Two Correlated Shot Noise Processes

The next example consists of two moving average processes,

$$X(t) = C(t) * R_X(t) + N_X(t) , \qquad \text{(IV.5)}$$

$$Y(t) = C(t) * R_Y(t) + N_Y(t) , \qquad \text{(IV.6)}$$

related because we let the pulse amplitudes of one be the same as for the other, but delayed in time by a constant amount:

$$R_Y(t) = R_X(t - t_0) . \qquad \text{(IV.7)}$$

Both X and Y have the same two-sided exponential pulse shape (equation [III.10]) used in the previous example. The noise processes N_X and N_Y are assumed to be independent of each other.

Figure 3 compares the cross-correlation function computed for a realization of this process with the exact theoretical function. As in Figure 2, there is considerable scatter caused by the finiteness of the sample—and not by the unevenness of the sampling.

d) Example: Cross-Correlations of Point Data

Sometimes the process being sampled is a yes/no signal, such as the arrival of photons in a detector. The data then consist of

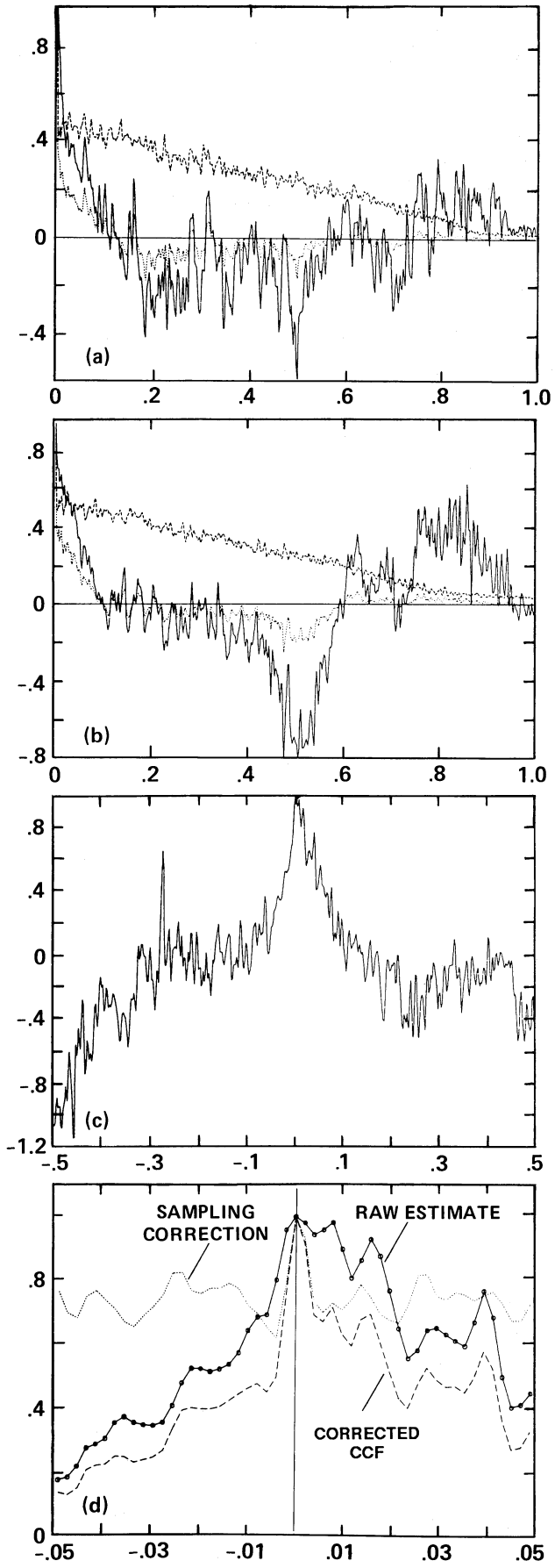

FIG. 5.—Computations based on the Algonquin Radio Observatory data on BL Lacertae (Medd *et al.* 1972; Andrew *et al.* 1978): (*a*) The autocorrelation function of the 2.8 cm data both with (*solid line*) and without (*dotted line*) sampling correction (*dashed line*, sampling correction); (*b*) Same as (*a*), for the 4.5 cm data; (*c*) The cross-correlation function, 2.8 cm data vs. 4.5 cm data, including correction for sampling; (*d*) expanded view of the zero-lag region of the cross-correlation function with (*dashed line*) and without sampling correction (*solid line*).

a sequence of times, $\{t_n;\ n = 1, 2, \ldots, N\}$. This situation is handled by considering the processes to be a sequence of δ-functions located at the times t_i; i.e., $X(t) = \sum \delta(t - t_i)$ and similarly for Y. Thus to evaluate the Fourier transform replace X_n and Y_n by unity, yielding an expression identical to equation (III.6).

For example, consider the determination of the relative positions of star images in a telescope. Assume we have a detector which records one rectangular coordinate of each photon, so the data are the values $\{X_n;\ n = 1, 2, \ldots, N\}$. (*Warning*: We have just switched notation and are now using X for the independent variable which was t previously.) Assume that a second star yields data values $\{Y_m;\ m = 1, 2, \ldots, M\}$. One way to estimate the separation between the two images is to find the lag at which the cross-correlation function of the data arrays is maximum.

Figure 4 demonstrates the computation a cross-correlation function of point data. These data are not quite the coordinate values. We have here simulated data produced by the Multichannel Astrometric Photometer (Gatewood 1987). In this astrometric measuring device a series of slits (comprising a *Ronchi grating*) is swept past the star images, causing the light from each star to be modulated in the form of an approximately sinusoidal light curve. The positional difference between two star images is thus converted to a phase difference between two modulated light curves. The lag at which the peak in the cross-correlation function of the light curves occurs is an estimate of this phase difference. If the detector used is a "time-tagging" one, such as the Multi-Anode Microchannel Array (MAMA; Timothy 1983), the data for each star consist of a set of times at which the photons were detected.

Figure 4 shows is the cross-correlation function of two such data streams, without any binning of the data on the time axis. The symbols represent the unsmoothed correlation function, and the sequence of curves represent successively increased amounts of smoothing. It is seen that the cross-correlation function is well determined, and in turn determines the relative phase of the two light curves well.

V. EPILOGUE: BL LACERTAE

This paper has described a technique for computing the discrete Fourier transform of unevenly spaced time series data. The procedure is applicable to frequency domain analysis, but the major use I contemplate is the computation of time-domain functions such as the auto- and cross-correlation functions.

I close with an example of such application to real data, namely, the Algonquin Radio Observatory data on the variation of the centimeter flux from the highly variable radio source BL Lacertae (Medd *et al.* 1972; Andrew *et al.* 1978). It is clear that there is a stochastic character to the variation, and therefore the random time-domain models described in Paper I, or the chaos models described in (Scargle 1989, [Paper IV]), may be appropriate.

Figure 5*a* and 5*b* show the ACFs of the time series data for the two wavelengths of the Algonquin observations, and Figure 5*c* shows the CCF of the same data. The CCF is of interest since the expanding plasma cloud model for radio source variability predicts that the activity at longer wavelengths will be delayed, because of radiative transfer effects, compared to that at shorter wavelengths. A future paper on BL Lacertae will describe the results of this kind of analysis.

I am grateful to Gary Villere for material incorporated into Appendix B; to Bob Hogan carrying out computations that were used in the example of § IV for point data; to Kent Cullers, William Borucki, and the anonymous referee for useful suggestions; to Donald Percival for pointing out an error in Paper II; and to Phyllis Scargle for stimulating discussions.

APPENDIX A

FORTRAN CODE: THE DISCRETE FOURIER TRANSFORM

This appendix presents source code for a subroutine to compute the Fourier transform of data with arbitrary spacing. The comments and notes in § II explain the solutions to the numerical problems which arise in this computation. The arrays in named common UNEVE are Extended Memory Array, which is peculiar to the Hewlett-Packard system used for the computations and will probably be unnecessary on other machines.

APPENDIX B

FUNDAMENTALS

This appendix discusses the choice of the frequencies to be used in performing inverse Fourier transforms, necessary for comparison with the original time series data (§ II) or for the computation of correlation functions from power spectra (§§ III and IV). For clarity we will often compare the general case with that of even sampling. Assume that we are measuring values of $X(t)$, a random process which is a continuous function of time.

The starting point is the data. We have N data points, each consisting of a time t_n and the value $X_n = X(t_n)$ of the variable at this time; i.e., we have the sets

$$\{X(t_n),\ n = 1, 2, \ldots, N\} \qquad \text{and} \qquad \{t_n,\ n = 1, 2, \ldots, N\}\ . \tag{B1}$$

The times will be assumed to be ordered:

$$t_1 < t_2 < t_3 < \cdots < t_N\ . \tag{B2}$$

In the evenly spaced case, there is a constant interval, $\Delta t = t_{n+1} - t_n$, between the samples. In the general case, the times are arbitrary, and the interval between adjacent points takes on many values.

```
FTN4X
$EMA (UNEVE)
      SUBROUTINE FT(XX,TSAMP,NN,WZ,NFREQ,SI)
C*************************************************************
C*    CALCULATE THE FOURIER TRANSFORM OF UNEVEN DATA         *
C*-----------------------------------------------------------*
C*                                                           *
C*    INPUT: -NN SAMPLE TIMES (TSAMP) AND VALUES (XX)        *
C*           -VALUE OF FUNDAMENTAL FREQUENCY (WZ)            *
C*           -NUMBER OF FREQUENCIES FOR TRANSFORM (NFREQ)    *
C*           -LENGTH OF FREQUENCY ARRAY (LFREQ -in COMMON)   *
C*           -FIDUCIAL ORIGIN OF TIME (TZERO -in COMMON)     *
C*           -SIGN OF TRANSFORM (SI)                         *
C*                                                           *
C*    Note that LFREQ must be >= 2 * NFREQ                   *
C*-----------------------------------------------------------*
C*                                                           *
C*    OUTPUT: -REAL AND IMAGINARY PARTS OF DFT (FTRX,FTIX)   *
C*                                                           *
C*    The transform is embedded in a zero-filled array       *
C*     of length LFREQ, with appropriate symmetries for      *
C*     real data.                                            *
C*-----------------------------------------------------------*
C*    JEFFREY D. SCARGLE                                     *
C*    MAIL STOP 245-3                                        *
C*    NASA-AMES RESEARCH CENTER                              *
C*    MOFFETT FIELD, CA   94035                              *
C*                                                           *
C*    HEWLETT-PACKARD FORTRAN (FTN4X COMPILER)               *
C*************************************************************
      COMMON/ UNEVE / FTRX(1024), FTIX(1024), LFREQ, TZERO


      DIMENSION XX(NN),TSAMP(NN)
      COMPLEX WORK
      TOL1 = 1.0 E -04
      TOL2 = 1.0 E -08
      WUSE = WZ
      FNN = FLOAT( NN )
      CONST1 = 1.0 / SQRT(2.0)
      CONST2 = SI * CONST1
      SUMT = 0.0
      SUMX = 0.0
      DO 100 I=1,NN
         SUMT = SUMT + TSAMP( I )
         SUMX = SUMX + XX( I )
100   CONTINUE
      ISTOP = NFREQ


C**********************************
C*  INITIALIZE FOR ZERO FREQUENCY  *
C**********************************

      TAU0 = SUMT / FNN      ! LIMIT OF TAU AS W-->0
      CSUM = FNN
      SSUM = 0.0
      FTRX(1) = SUMX / SQRT( FNN )
      FTIX(1) = 0.0
      WDEL = WUSE
      WRUN = WUSE
      II = 2
```

```
C*****************************************
C*         START FREQUENCY LOOP          *
C*****************************************
150   CONTINUE

C******************
C*  CALCUATE TAU  *
C******************
      CSUM =  0.0
      SSUM =  0.0
      SUMTC  =  0.0
      SUMTS  =  0.0

      DO 190 I = 1,NN
         TTT = TSAMP( I )
         ARG1 = 2.0 * WRUN * TTT
         ARG = FOLD( ARG1 )
         TCOS = COS ( ARG )
         TSIN = SIN ( ARG )
         CSUM = CSUM + TCOS
         SSUM = SSUM + TSIN
         SUMTC  = SUMTC  + TTT * TCOS
         SUMTS  = SUMTS  + TTT * TSIN
190   CONTINUE

      WATAN = ATAN2( SSUM , CSUM )
      IF(ABS(SSUM).GT.TOL1 .OR. ABS(CSUM).GT.TOL1) GOTO 200
      WATAN = ATAN2( -SUMTC , SUMTS )
200   CONTINUE


      WTAU = 0.5 * WATAN
      WTNEW = WTAU
      SUMR  = 0.0
      SUMI  = 0.0
      SCOS2 = 0.0
      SSIN2 = 0.0
      CROSS = 0.0

C***********************************
C*   SUMMATIONS OVER THE SAMPLES   *
C***********************************

      DO 440 I = 1,NN

         TIM = TSAMP(I)
         ARG1 = WRUN * TIM - WTNEW
         ARG = FOLD( ARG1 )
         TCOS = COS(ARG)
         TSIN = SIN(ARG)

         CROSS = CROSS + TIM * TCOS * TSIN
         SCOS2 = SCOS2 + TCOS * TCOS
         SSIN2 = SSIN2 + TSIN * TSIN

         XD = XX(I)
         SUMR = SUMR + XD * TCOS
         SUMI = SUMI + XD * TSIN

440      CONTINUE

      FTRD = CONST1 * SUMR / SQRT(SCOS2)
      IF ( SSIN2 .LE. TOL1 ) GOTO 450
      FTID = CONST2 * SUMI / SQRT(SSIN2)
      GOTO 460
```

```
450   CONTINUE                        ! USE L'HOPITAL'S RULE
      FTID = CONST2 * SUMX / SQRT( FNN )
      IF ( ABS(CROSS) .GT. TOL2 ) FTID = 0.0

460   CONTINUE

      PHASE1 = WTNEW - WRUN * TZERO
      PHASE = FOLD( PHASE1 )
      WORK = CMPLX( FTRD, FTID )*CEXP( CMPLX(0.0, PHASE ) )
      FTRX(II) = REAL( WORK )
      FTIX(II) = AIMAG( WORK )
      II = II + 1
      WRUN = WRUN + WDEL
      IF( II .LE. ISTOP )GOTO 150

C*********************************************
C* ZERO-FILL TRANSFORM (OVERSAMPLE INVERSE)  *
C* IMPOSE SYMMETRY FOR REAL DATA             *
C*********************************************
      IF( 2 * NFREQ .GT. LFREQ ) GOTO 999
      I1 = NFREQ + 1    ! THIS GIVES GOOD RESULTS ON EVEN DATA

      DO 320 I= I1,LFREQ
         FTRX(I) = 0.0
         FTIX(I) = 0.0
320      CONTINUE

      NSTOP = LFREQ / 2
      DO 340 I=2, NSTOP
         IPUT = LFREQ -I + 2
         FTRX(IPUT)  =  FTRX(I)
         FTIX(IPUT)  = -FTIX(I)
340      CONTINUE
465   CONTINUE

      RETURN
999   CONTINUE
      END


      FUNCTION FOLD( ARG )
C************************************************
C*  This function folds trigonometric arguments  *
C*    to account for the limited range for       *
C*    the trigonometric functions, so the value  *
C*    of ARGMAX will depend on the system.       *
C************************************************
      PI = 3.1415926535898
      ARGMAX = 8000.0 * PI
      FOLD = ARG
10    CONTINUE
      IF( FOLD .LE. ARGMAX) GOTO 20
         FOLD = FOLD - ARGMAX
         GOTO 10
20    CONTINUE
      IF( FOLD .GT. -ARGMAX) GOTO 30
         FOLD = FOLD + ARGMAX
         GOTO 20
30    CONTINUE
      RETURN
      END
```

We call the interval (t_1, t_N) the *sampling interval* and define the sampling range as

$$T_R = t_N - t_1 = (N - 1)\Delta t \,, \tag{B3}$$

not to be confused with the *period* to be introduced shortly. (Note that Δt is defined for the even sampling case only.) The goal is to compute X's frequency space representation and test it by inverse transforming back to the time domain. This is accomplished by computing the DFT at a set of evenly spaced frequencies and performing the inversion in the standard way:

$$X(t) = \sum \exp\,(i\omega t)\mathrm{FT}_X(\omega) \,, \tag{B4}$$

where the sum is over evenly spaced frequencies. Note that the time offset $\tau(\omega)$ is not used in this expression. Once the Fourier transform has been evaluated we can forget about the fact that the data are unevenly sampled, and the inversion back to the time domain can be carried out just as it would be for evenly spaced data.

When this inversion is carried out, it turns out that the function thus represented is not really $X(t)$, but a function periodically replicated outside of the sampling interval. We find, in fact, that for the evenly spaced case

$$X(t_{N+j}) = X(t_j) \,, \tag{B5}$$

for any integer j. Thus the period of the replicated function is not the sampling range, but the slightly larger value

$$T_P = t_{N+1} - t_1 = N\,\Delta t \,, \tag{B6}$$

larger than the range by one of the intervals Δt. We have the following relation giving the period in terms of the range:

$$T_P = T_R N/(N - 1) \,. \tag{B7}$$

Since N and T_R are well defined for arbitrary sampling, even though Δt is not, we use equation (B7) to define the period T_P and the corresponding fundamental frequency

$$\omega_0 = 2\pi/T_P \,. \tag{B8}$$

However, *for the computation of correlation functions it turns out to be better to use a fundamental frequency calculated as though the sampling interval were larger than it actually is by a factor of 2:*

$$\omega_0 = 2\pi/(2T_P) = \pi/T_P \,. \tag{B9}$$

For this choice nicely eliminates an undesirable time-domain effect called "wraparound." The use of equation (B9) is analogous to the standard procedure to avoid wraparound, namely, zero-filling the data array to twice its original length.

Wraparound results from the periodic replication inherent in any finite DFT. What is undesirable in a correlation function is the spillover of the right-hand end of the data into the left-hand end. The effect is eliminated because zero-extension assures that any such spilled data is multiplied by zero, thus producing no effect. (See Paper I, Fig. 15, for a graphic description of wraparound.) Any larger value of ω_0 allows overlap for some value of the lag, while use of a smaller ω_0 only increases the computation time without changing the result. A second and apparently independent reason for using equation (B9) is that, at least for even sampling, this choice approximately minimizes the mean-square error of the ACF estimate (Kay 1981).

All inverse Fourier transforms in this paper were done using a standard Fast Fourier Transform (FFT) algorithm using evenly spaced frequencies based on the value of the fundamental in equation (B9). There is one characteristic of these frequencies that holds for evenly spaced data, but is lost for uneven spacing—the statistical independence of the Fourier Transform at these frequencies. This results from the fact that the frequency functions that are being used to expand in are not orthogonal with respect to the operation of summing over the sample times. The small correlations that exist may be important in some applications, and should be kept in mind, especially in statistical analyses.

The maximum frequency in the sum in equation (B4), $\omega_{\max}$, is less well defined than is the fundamental. For even spacing it is the Nyquist frequency, $2\pi/\Delta t$. But with uneven sampling there is no single value of Δt. In practice a good choice for $\omega_{\max}$ is often 2π divided by some average or effective value of the actual sampling intervals $\Delta t_i = t_i - t_{i-1}$ (assuming the t_i are ordered). But in fact the value of $\omega_{\max}$ should be regarded as a more or less free *smoothing parameter*. The examples in the text show that the standard case, defined to be the choice of $\omega_{\max}$ such that the number of frequencies is equal to the number of data points—as in the evenly spaced case—corresponds to no smoothing, but not quite exact replication of the original data. The choice to use in practice depends on the sampling at hand, the amount of noise in the data, and what the Fourier transform will be used for.

A note about the number of frequencies: In the standard case the number of frequencies is actually twice the number of data points, because of our choice of a fundamental that is half as large (eq. [B9]) as the conventional choice. However, the maximum frequency used—which is what is relevant to how smoothed the reconstructed data are—is the same as it would be, say, in the case of evenly spaced data transformed in the conventional way.

The determination of the fundamental frequency to use in the inversion of the cross-spectrum (to yield the cross-correlation function) is a bit more complex. We need to define a kind of joint fundamental frequency for two data sets. Equation (B9) still applies, but what value is one to use for T_P if the sampling intervals for X and Y are not the same? The following procedure has worked well in several cases: compute the range T_R to be the length of the smallest interval that includes all the samples of both X and Y, i.e.,

$$t_{\min} = \min\,[t_i^{(x)}, t_i^{(y)}] \tag{B10}$$

$$t_{\max} = \max\,[t_i^{(x)}, t_i^{(y)}] \tag{B11}$$

$$T_R = (t_{\max} - t_{\min}) \tag{B12}$$

so that

$$T_P = T_R N/(N-1) = (t_{\max} - t_{\min})N/(N-1)\,, \tag{B13}$$

where $N = (N_X + N_Y)/2$, the average number of points in the samples.

APPENDIX C

EFFECT OF A CONSTANT COMPONENT

A common first step in spectral or correlation analysis is to subtract the mean of the data (i.e., the sample mean) from the time series. Part of the rationale for doing this is a fear that if the mean is left in the resulting zero-frequency spike in the power spectrum will leak to other frequencies, or that the ACF will not behave properly for large t.

A simple example shows that removing the sample mean can give misleading results. Consider the Wold Representation:

$$X(t) = C(t) * R(t) + D\,, \tag{C1}$$

where $C(t)$ is a pulse shape, the process $R(t)$ represents the amplitudes of randomly occurring pulses, and the asterisk is the convolution operator. The process D contains all of the determinism of X. The process R contains all of the randomness of X; it is always at least white, but in physical applications it is often even more random—namely independently distributed. (Paper I, § IIc, is a discussion of this surprisingly general model.) For simplicity we take D to be constant, making X essentially a *moving average* process (i.e., filtered white noise, closely related to "shot noise").

We seek the auto-correlation function of the pulse shape $C(t)$. From equation (C1) the ACF of X is

$$\rho_X(t) = \langle R^2 \rangle S_C(t) + 2D\langle R \rangle \sum C(t)dt + D^2\,, \tag{C2}$$

where $S_C(t)$ is the autoconvolution of $C(t)$:

$$S_C(t) = \sum C(t')C(t-t')dt'\,. \tag{C3}$$

This expression can be interpreted as the correlation of $C(t)$ with its time reverse, $C(-t)$, and is the same as the ACF of the pulse shape if the pulse is time-symmetric.

If $\langle R \rangle = 0$ or the pulse shape has zero area (either of which means that the pulsed component of X, namely, $C * R$, has zero mean value) the above expression simplifies to:

$$\rho_X(t) \sim \text{pulse shape autoconvolution} + \text{constant}\,. \tag{C4}$$

In this case, we could subtract the mean of the data; this would have the effect of forcing $D = 0$, and the constant in equation (C4) would accordingly be zero.

However, this analysis does not work if $\langle C * R \rangle \neq 0$, as is often the case in astronomy. For example, if $C(t)$ represents a pulse of light and $R(t)$ is the amplitude of the pulse at time t, then C and R are both positive definite. In such cases one can calculate the autocorrelation function of $X - K$, where K is an arbitrary constant, and search for that value of K which produces the expected asymptotic behavior of the ACF. In most cases one expects the ACF to decline smoothly to zero for large lags, assuming the samples span a time interval substantially greater than the widths of the pulse.

In effect we are finding the representation (C1) that corresponds to $D = 0$. We could do this directly if there were little or no overlap between the pulses, for then D would be the minimum value of the observed time series.

It may be noted that similar considerations hold for power spectrum analysis. A random process such as in equation (C1) has a continuous power spectrum, which may well be nonzero at zero frequency. Hence it would be a mistake to subtract out the mean value, forcing the zero-frequency power $P_X(0)$ to be zero. More appropriate in this case would be to adjust the subtracted constant so that $P_X(0)$ is equal to the limit of $P_X(\omega)$ as $\omega \to 0$.

APPENDIX D

WEIGHTED DATA

The effect of unequally weighted data points can be found by considering what happens if there are two points that coincide, and realizing that this is the same as a single point of double weight. If w_i is the weight of point i, with the normalization $\sum w_i = N$ (unit weight per point on average), we find

$$\mathrm{FT}_X(\omega) = F_0 \sum (Aw_n X_n \cos \omega t'_n + iBw_n X_n \sin \omega t'_n)\,, \tag{D1}$$

$$A(\omega) = (\sum w_n \cos^2 \omega t'_n)^{-1/2}\,; \qquad B(\omega) = (\sum w_n \sin^2 \omega t'_n)^{-1/2}\,, \tag{D2}$$

$$\tau(\omega) = (1/2\omega) \tan^{-1} (\sum w_n \sin 2\omega t_n / \sum w_n \cos 2\omega t_n)\,. \tag{D3}$$

REFERENCES

Andrew, B. H., MacLeod, J. M., Harvey, G. A., and Medd, W. J. 1978, *A.J.*, **83**, 863.
Box, G. E. P., and Jenkins, G. M. 1970, *Time Series Analysis, Forecasting and Control* (San Francisco: Holden-Day).
Bracewell, R. N. 1965, *The Fourier Transform and Its Applications* (New York: McGraw-Hill).
Brockwell, P. J., and Davis, R. A. 1987, *Time Series: Theory and Methods* (New York: Springer-Verlag).
Edelson, R. A., and Krolik, J. H. 1988, *Ap. J.*, **333**, 646.
Gastner, M., and Roberts, J. B. 1975, *J. Inst. Maths. Appl.*, **15**, 195.
———. 1977, *Proc. Roy. Soc. London A*, **354**, 27.
Gatewood, G. 1987, *A.J.*, **94**, 213.
Jenkins, G. M., and Watts, D. G. 1968, *Spectral Analysis and Its Applications* (San Francisco: Holden-Day).
Kay, S. M. 1981, *IEEE Trans. ASSP*, **ASSP-29**, 859.
Mayo, W. T., Shay, M. T., and Riter, S. 1974, NASA Sci. Tech. Inf. Facility preprint A76-10428, FF No. 672.
Medd, W. J., Andrew, B. H., Harvey, G. A., and Locke, J. L. 1972, *Mem. R.A.S.*, **77**, 109.
Papoulis, A., 1962, *The Fourier Integral and Its Applications* (New York: McGraw-Hill).
Press, W. H., and Rybicki, G. B., 1989, *Ap. J.*, **383**, 277.
Press, W. H., and Teukolsky, S. A. 1988, *Comput. Phys.*, **2**, No. 6, 77.
Scargle, J. D. 1981, *Ap. J. Suppl.*, **451**, (Paper I).
———. 1982, *Ap. J.*, **263**, 835 (Paper II).
———. 1989, *Ap. J.*, submitted.
Scott, P. F., 1976, Ph.D. thesis, Rensselaer Polytechnic Institute, Troy, New York.
Timothy, G., 1983, *Pub. A.S.P.*, **95**, 810.

JEFFREY D. SCARGLE: Mail Stop 245-3, Theoretical Studies Branch, Space Science Division, NASA-Ames Research Center, Moffett Field, CA 94035