FFT

PURPOSE

Compute the discrete fast Fourier transform of a variable.

DESCRIPTION

The Fourier transform converts a time domain function into a frequency domain function while the inverse Fourier transform converts a frequency domain function into a time domain function. If h(t) represents the function in the time domain and H(f) represents the function in the frequency domain, then the definitions of the Fourier transform and the inverse Fourier transformation respectively are:

$$H(f) = \int_{-\infty}^{\infty} h(t)e^{2\pi ift}dt \qquad \textbf{(EQ 3-34)}$$

$$h(t) = \int_{-\infty}^{\infty} H(f)e^{-2\pi ift}df \qquad \textbf{(EQ 3-35)}$$

In practice, functions are sampled at equally spaced discrete points. The discrete Fourier transform and the discrete inverse Fourier transforms respectively are:

$$H_n = \sum_{k=0}^{N-1} h_k e^{\frac{2\pi ikn}{N}} \qquad \textbf{(EQ 3-36)}$$

$$h_k = \frac{1}{N}\sum_{n=0}^{N-1} H_n e^{\frac{-2\pi ikn}{N}} \qquad \textbf{(EQ 3-37)}$$

where k represents the sampled points in the time domain, lower case n represents the sampled points in the frequency domain, and N is the number of sampled points. DATAPLOT calculates the discrete Fourier and inverse Fourier transforms. If you wish to calculate these transforms for a function, then evaluate this function at a series of points. This can be accomplished with something like the following:

```
LET FUNCTION F = <define the function in terms of X1>
LET X1 = SEQUENCE 0 0.5 40
LET Y1 = F
LET R2 C2 = FOURIER TRANSFORM Y1
```

The fast Fourier and the inverse fast Fourier transforms are more computationally efficient ways to calculate the Fourier and inverse Fourier transforms. See the REFERENCE section below for references which give a more detailed explanation of Fourier transforms. There are different definitions for the Fourier and inverse Fourier transforms in the literature. The NOTE sections below discuss this.

SYNTAX 1

LET <r1> <c1> = FFT <y1>   <SUBSET/EXCEPT/FOR qualification>

where <y1> is the real component of the input variable;
    <r1> is a variable containing the real component of the calculated FFT;
    <c1> is a variable containing the complex component of the calculated FFT;
and where the <SUBSET/EXCEPT/FOR qualification> is optional.

SYNTAX 2

LET <r1> <c1> = FFT <y1> <y2>   <SUBSET/EXCEPT/FOR qualification>

where <y1> is the real component of the input variable;
    <y2> is the complex component of the input variable;
    <r1> is a variable containing the real component of the calculated FFT;
    <c1> is a variable containing the complex component of the calculated FFT;
and where the <SUBSET/EXCEPT/FOR qualification> is optional.

EXAMPLES

LET RFFT CFFT = FFT Y1

NOTE 1

DATAPLOT uses the CFFTF, CFFTB, and CFFTI routines, written by Paul Swartztrauber of the Nation Center for Atmospheric Research, from the SLATEC Common Mathematical Library to compute this function. SLATEC is a large set of high quality, portable public domain Fortran routines for various mathematical capabilities maintained by seven federal laboratories. Versions of DATAPLOT prior to 95/8 use the FOUR1 routine from the Numerical Recipes book (see the REFERENCE section below) to calculate the FFT.

One difference is that the SLATEC routines do not require zero padding to a power of two for the number of elements in the data sample (DATAPLOT did this automatically when using the FOUR1 routine). However, the SLATEC routines are most efficient when the number of elements is a product of small prime factors. When this is not the case, the algortihm loses its speed advantage over the standard Fourier transform. In this case, you may want to zero pad the variable yourself. DATAPLOT no longer does it automatically.

NOTE 2

Some references reverse the roles of the Fourier transform and the inverse Fourier transform. From the formulas, they differ only in whether the exponent is positive or negative and the scaling constant used. The choice of convention is usually determined by the field of application. The command SET FOURIER EXPONENT +, the default, uses the convention of the FFT having a positive exponent and the inverse FFT having a negative exponent. The command SET FOURIER EXPONENT - uses the convention of a minus exponent for the FFT and a positive exponent for the inverse FFT. See the documentation for this command in the SET Subcommands chapter in Volume I for details. Versions that use the FOUR1 routine (i.e., prior to 95/8) should not use the SET FOURIER EXPONENT command as it does not work as intended.

NOTE 3

Some references use different scaling factors for the FFT and the inverse FFT. One common convention is to divide both the FFT and the inverse FFT by the square root of N. Another is to not divide either the FFT or inverse FFT by any scale factor. If you prefer a different scaling, simply use the LET command to multiply or divide the FFT or inverse FFT by the desired scaling factor.

NOTE 4

By default, DATAPLOT returns the frequencies in the following order: the first point corresponds to the aliased frequency (i.e., the most positive and most negative frequency), points 2 thru N/2 are the negative frequencies arranged from largest magnitude to smallest magnitude, point (N/2)+1 is the zero frequency, and points (N/2)+2 thru N are the positive frequencies arranged from smallest magnitude to largest magnitude.

Most FFT algorithms return the frequencies in the following order: the first point is the zero frequency, points 2 thru (N/2) are the positive frequencies from arranged from smallest magnitude to the largest magnitude, the point (N/2)+1 is the aliased point, and the points (N/2)+2 thru N are the negative frequencies arranged from largest magnitude to smallest magnitude.

Versions starting with 95/8 can specify the desired order using the SET FOURIER ORDER command. SET FOURIER ORDER DATAPLOT uses the default DATAPLOT order and SET FOURIER ORDER STANDARD uses the second order. Be aware that the generated values will differ somewhat since the order of the data input into the FFT algorithm may be different.

NOTE 5

An alternate (mathematically equivalent) formulation specifies these transformations in terms of the angular frequency (radians per second) ω, where ω = 2πf. The equations then become:

$$H(\omega) = \int_{-\infty}^{\infty} h(t)e^{i\omega t}dt \tag{EQ 3-38}$$

$$h(t) = \left(\frac{1}{2\pi}\right)\int_{-\infty}^{\infty} H(\omega)e^{-i\omega t}d\omega \tag{EQ 3-39}$$

DEFAULT

None

SYNONYMS

None

RELATED COMMANDS

INVERSE FOURIER TRANSFORM	=	Compute the inverse Fourier transform.
FOURIER TRANSFORM	=	Compute the Fourier transform.

INVERSE FFT	=	Compute the inverse FFT.
COSINE TRANSFORM	=	Compute the cosine transformation.
SINE TRANSFORM	=	Compute the sine transformation.
SPECTRAL PLOT	=	Generate a spectral plot.

REFERENCE

“Numerical Methods and Software,” Kahaner, Moler, and Nash, Prentice Hall, 1989 (chapter 11).

“Numerical Recipes: The Art of Scientific Computing (FORTRAN Version),” Press, Flannery, Teukolsky, and Vetterling, Cambridge University Press, 1989 (chapter 12).

APPLICATIONS

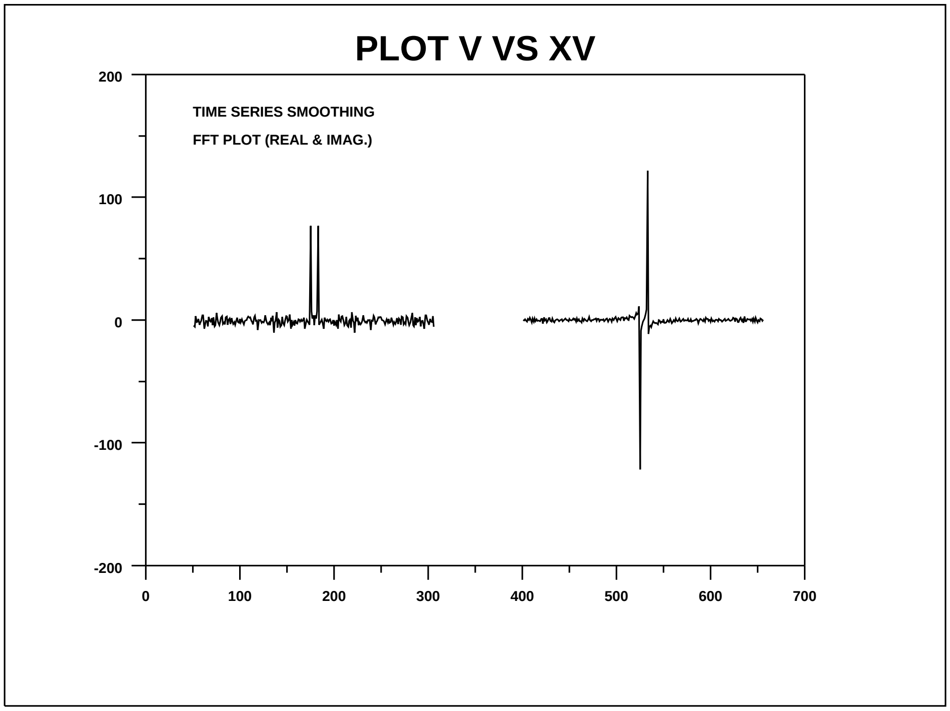
Frequency analysis of time series, signal processing

IMPLEMENTATION DATE

87/5, algorithm replaced 95/8, SET FOURIER EXPONENT, SET FOURIER ORDER added 95/8

PROGRAM

```
LEGEND 1 TIME SERIES SMOOTHING; LEGEND 2 FFT PLOT (REAL & IMAG.)
TITLE AUTOMATIC
LET X = SEQUENCE 0 .1 25.55; LET YS = SIN(X)
LET YN = NORMAL RANDOM NUMBERS FOR I = 1 1 256; LET YN = YN/10
LET Y = YS+YN
.
LET U V = FFT Y
LET NU = NUMBER U; LET XU = SEQUENCE 1 1 NU
LET XU = XU+50
LET XV = XU+350
.
LET U = 2.5*U
PLOT U VS XU AND
PLOT V VS XV
```


PLOT V VS XV
TIME SERIES SMOOTHING
FFT PLOT (REAL & IMAG.)