DECONVOLUTION

PURPOSE

Compute the discrete deconvolution of two variables.

DESCRIPTION

Mathematically, the convolution of 2 continuous distributions g and h is defined as:

$$g*h = \int_{-\infty}^{\infty} g(\tau)h(t-\tau)d\tau \qquad \textbf{(EQ 3-32)}$$

In practice, h is typically a data stream while g is a response function. The response function is typically a peaked function that goes to zero in both directions from that peak. The effect of convolution is to smear the data stream with the response function.

DATAPLOT computes the convolution from the functions sampled at discrete points (see the sample program for an example of how to evaluate a function at a discrete set of points). This is referred to as discrete convolution. If X is the data stream with n_x points and Y is the response function with n_y points, then DATAPLOT computes the convolution as:

Z(1) = X(1)*Y(1)
Z(2) = X(1)*Y(2) + X(2)*Y(2)
Z(3) = X(1)*Y(3) + X(2)*Y(3) + X(3)*Y(3)
etc.

This can be written as:

$$Z_i = \sum_{j=1}^{i} X_{i-j+1} Y_j \qquad \textbf{(EQ 3-33)}$$

where i goes from 1 to n_x+n_y-1. Deconvolution is an attempt to “unsmear” a data set given a known response function. DATAPLOT computes the deconvolution of the data stream variable Y and the response variable X as:

Z(1) = X(1)/Y(1)
Z(2) = (X(2) - Y(2)*Z(1))/Y(1)
Z(3) = (X(3) -Y(3)*Z(1) -Y(2)*Z(2))/Y(1)
etc.

The resulting deconvolved variable will have n_y-n_x+1 elements. If the response variable contains more elements than the data stream variable, the roles of X and Y are reversed in the above formulas and the returned number of elements will be n_x-n_y+1.

One potential problem is that division by zero or overflow can occur if the response variable is zero or near enough to zero. The solution is to trucate the response variable so these values are not included.

SYNTAX

LET <z> = DECONVOLUTION <y> <x> <SUBSET/EXCEPT/FOR qualification>

where <y> is the data stream variable with n_y elements;
<x> is the response variable with n_x elements;
<z> is a variable containing the computed deconvolution values (it will contain n_y-n_x+1 elements or n_x-n_y+1 elements if $n_x>n_y$);
and where the <SUBSET/EXCEPT/FOR qualification> is optional.

EXAMPLES

LET Y3 = DECONVOLUTION Y1 Y2

DEFAULT

None

SYNONYMS

None

RELATED COMMANDS

CONVOLUTION	=	Compute the discrete convolution of two variables.
FFT	=	Compute the Fast Fourier Transform of two variables.

REFERENCE

"Numerical Recipes: The Art of Scientific Computing (FORTRAN Version)," Press, Flannery, Teukolsky, and Vetterling, Cambridge University Press, 1989 (chapters 12 and 13).

APPLICATIONS

Mathematics

IMPLEMENTATION DATE

Pre-1987

PROGRAM

```
LET FUNCTION F1 = ((X+1)**2)/((X**2)+1)/2
LET FUNCTION F2 = 2**((-(X2+1.3/2)**2)/(A**2))-2**((-(X2-1.3/2)**2)/(A**2))
LET A = 0.85
LET XMIN1 = -7; LET XINC1 = .1; LET XMAX1 = 7
LET X = SEQUENCE XMIN1 XINC1 XMAX1; LET Y1 = F1
LET XMIN2 = -2; LET XINC2 = .1; LET XMAX2 = 2
LET X2 = SEQUENCE XMIN2 XINC2 XMAX2; LET Y2 = F2
.
MULTIPLOT 2 2; MULTIPLOT CORNER COORDINATES 0 0 100 100
TITLE SAMPLE DATA; PLOT Y1 X
TITLE RESPONSE FUNCTION; PLOT Y2 X2
LET Y3 = CONVOLUTION Y1 Y2
LET X3MIN = 2*XMIN1; LET X3MAX = 2*XMAX1
LET X3 = SEQUENCE X3MIN XINC1 X3MAX; TITLE CONVOLUTION FUNCTION
PLOT Y3 VS X3
.
LET Y4 = DECONVOLUTION Y3 Y2
TITLE DECONVOLUTION FUNCTION
PLOT Y4 VS X
END OF MULTIPLOT
```