NAME

FFT − Return in-place the direct or inverse fast Fourier transform of a sequence.

SYNOPSIS

CALL FFT(N,INVDIR,SCALE,SHIFT, DATA,B, RC)

N	is the INTEGER*4 number of samples in DATA, a positive power of 2
INVDIR	is INTEGER*4; +1 => transform the data, −1 => invert the transform
SCALE	is LOGICAL*4; T => scale output or assume input is scaled
SHIFT	is LOGICAL*4; T => frequency-shift result or assume input was frequency-shifted
DATA(N)	is the COMPLEX*16 input sequence, then the result
B	is the REAL*8 upper limit of integration for the transform or inverse
RC	is the INTEGER*4 return code; see below

DESCRIPTION

First the routine sanity-checks its input parameters, sets RC accordingly, and returns without doing anything if RC is not 0. Next it arranges the input sequence by ascending bit-reversed index, and uses the butterfly algorithm to evaluate the signal flow graph. Then it finds the sampling interval and new upper limit of integration. Finally it scales the output values if that has been requested, and if frequency-shifting is specified either shifts the forward transform or fixes up the inverse transform of the shifted input.

SEE ALSO

DFT, which computes the 1-dimensional discrete Fourier transform or inverse
FFT2ST and FFT2TR, which compute the 2-dimensional fast Fourier transform or inverse
DFT2, which computes the 2-dimensional discrete Fourier transform or inverse

DIAGNOSTICS

n output RC=0 if all went well, or certain bits are set to 1 if the following error conditions occur:

1 bit	1 => N is not a positive power of 2
2 bit	1 => INVDIR is not +1 or −1
4 bit	1 => B is not positive

LINKAGE

gfortran source.f −L${HOME}/lib −lmisc

AUTHOR

Michael Kupferschmid

REFERENCES

"Computing Fourier Transforms" by Michael Kupferschmid

EXAMPLE

```
      INTEGER*4 N/8/
      LOGICAL*4 SCALE/.FALSE./,SHIFT/.FALSE./
      COMPLEX*16 DATA(8)/8*(0.0D0,0.0D0)/
      REAL*8 B/3.5D0/
      INTEGER*4 RC
C
C     transform the pulse waveform of [1, Section 4]
      DATA(2)=(1.5D0,0.D0)
      DATA(3)=(3.0D0,0.D0)
      DATA(4)=(1.5D0,0.D0)
      INVDIR=+1
      CALL FFT(N,INVDIR,SCALE,SHIFT, DATA,B, RC)
      WRITE(6,901) RC,DATA
  901 FORMAT(I2/(' (',F5.2,',',F5.2,')'))
C
C     invert the transform
      INVDIR=-1
      CALL FFT(N,INVDIR,SCALE,SHIFT, DATA,B, RC)
      WRITE(6,901) RC,DATA
      STOP
      END
```

This example produced the output below. The inverse recovers the input pulse waveform exactly.

```
unix[1] a.out
 0
( 6.00, 0.00)
( 0.00, 5.12)
(-3.00, 0.00)
( 0.00,-0.88)
( 0.00, 0.00)
( 0.00, 0.88)
(-3.00, 0.00)
(-0.00,-5.12)
 0
( 0.00, 0.00)
( 1.50,-0.00)
( 3.00, 0.00)
( 1.50, 0.00)
( 0.00, 0.00)
( 0.00, 0.00)
( 0.00,-0.00)
( 0.00,-0.00)
unix[2]
```