



1 SUMMARY

Calculates **discrete Fourier transforms**. Given equally spaced complex data $f(n)$, $n=0,1,2,..,N-1$ of period N , and assuming $f(N)=f(0)$ it calculates the direct transform in the form of the discrete sum

$$\tilde{f}(m) = \sum_{n=0}^{N-1} f(n) \exp\left\{2\pi i m \frac{n}{N}\right\}$$

or alternatively given $\tilde{f}(m)$, $m=0,1,2,..,N-1$ calculates the inverse transform in the form of the discrete sum

$$f(n) = \frac{1}{N} \sum_{m=0}^{N-1} \tilde{f}(m) \exp\left\{-\frac{2\pi i m n}{N}\right\}$$

where in both cases N must be a power of 2 (a restriction which may be overcome in some cases). The real case is obtained by setting the relevant imaginary parts to zero.

The 'Fast Fourier Transform' method is used, see W.M. Gentleman and G. Sande, 'Fourier Transforms in Place', Proc. Fall Joint Computer Conference, 1966.

ATTRIBUTES — **Version:** 1.0.0. **Types:** FT01A; FT01AD. **Original date:** September 1967. **Origin:** A.R.Curtis, Harwell.

2 HOW TO USE THE PACKAGE

2.1 Argument list and calling sequence

The single precision version

```
CALL FT01A(N, INV, FREAL, FIMAG)
```

The double precision version

```
CALL FT01AD(N, INV, FREAL, FIMAG)
```

N is an **INTEGER** variable which must be set by the user to N the number of data points and its value must be a power of 2 (see §2.2 to overcome this restriction). If the routine finds that N is not a power of 2, the argument **INV** will be set to -1 and an immediate return to the calling program is made.

INV is an **INTEGER** variable which must be set by the user to indicate whether the routine is to calculate the direct transform (the first formula in §1), or the inverse transform (the second formula in §1). The only two possible values for **INV** are

- 1 for the inverse transform
- 2 for the direct transform

INV may be altered by the routine, see argument **N**.

FREAL is a **REAL** (**DOUBLE PRECISION** in the **D** version) array of length at least N . For **INV**=2 the user must set **FREAL(I)**, $I=1, N$ to the real part of $f(n)$ $n=0,1,..,N-1$; and on return the real part of $f(m)$ $m=0,1,..,N-1$ will be found in **FREAL(I)**, $I=1, N$. For **INV**=1 the reverse applies, i.e. the user provides the real parts of $f(m)$ and receives back those of $f(n)$.

FIMAG is a **REAL** (**DOUBLE PRECISION** in the **D** version) array of length at least N . For **INV**=2 the user must set **FIMAG(I)**, $I=1, N$ to the imaginary part of $f(n)$ $n=0,1,..,N-1$; and on return the imaginary part of $f(m)$ $m=0,1,..,N-1$ will be found in **FIMAG(I)**, $I=1, N$. For **INV**=1 the reverse applies, i.e. the user provides the imaginary parts of $f(m)$ and receives back those of $f(n)$.

2.2 Some notes on using the routine

Although called a ‘‘Fourier transform’’, all this routine actually does is to carry out the multiplication of a complex-valued vector by the complex-valued matrix whose (m,n) component is defined by the first equation in §1, or by its inverse as defined in the second equation in §1. The user should take care that the interpretation he puts on this process is consistent with it. In particular, attention should be paid to the following points:

- (i) Discrete sums are calculated, not integrals, and the truncation error may not be negligible.
- (ii) The transform found is that of periodic data (period N), for which $f(n+kN)=f(n)$ for all (positive or negative) integers k . Thus if one wishes to approximate to the transforms of a function which is zero outside a limited range, N should be increased and the necessary extra values supplied as zeros; in this way, the case where the number of data points is not initially a power of 2 can also be handled.
- (iii) If one is going to multiply the transforms of two functions together, so that the inverse function is their convolution, this will be non-zero over the sum of the ranges of the two original functions. Therefore, each of these should first be extended by zeros as described above, to avoid the result ‘‘overlapping’’ itself because of the periodicity.
- (iv) Any discontinuities in the function $f(x)$, from which the values of $f(n)$ are taken, or its derivatives, will dominate the transform $f(m)$; this includes a discontinuity between $f(N-1)$ and $f(N)=f(0)$, because of the periodicity.
- (v) The components of $f(N-j)$, for small values of j , arise from slowly varying components of f , with ‘‘negative frequencies’’ $-j$; if it is desired to weight down high-frequency components of f to smooth noisy data, it is the components around $m=\frac{1}{2}N$ which should be decreased (symmetrically about $m=\frac{1}{2}N$, i.e. about index $\frac{1}{2}N+1$ in the argument arrays).
- (vi) The process described in §1 lacks symmetry; the values of $f(0)$ are used but not the (presumed equal) values of $f(N)$. Thus the data value which is symmetrically related to $f(n)$ is $f(N-n)$, and only if these are equal to all n will the transform $f(m)$ be real. These symmetrically related values are returned at indices $n+1$ and $N+1-n$ in the argument arrays, i.e. the sum of their indices is $N+2$.
- (vii) If the function $f(x)$ is non-zero over a range $-a < x < b$ say, with a and b positive, it will usually be convenient to shift the origin of x to a suitable positive value of n , so long as $x=-a$ corresponds to $n \geq 0$ and $x=b$ corresponds to $n \leq N$. If $x=0$ corresponds to $n=n_0$, then the transform of $f(x)$ (in the ordinary sense) will be

$$f(m) \exp\left(\frac{2\pi i m n_0}{N}\right).$$

Note that if $n_0=N/2$, the multiplying factor is $(-1)^m$.

3 GENERAL INFORMATION

Use of common: none.

Workspace: none.

Other routines: none.

Input/Output: none.

Restrictions:

N must be a power of 2,

INV can only be 1 or 2.

4 METHOD

This routine implements a method for ‘fast Fourier transforms’ called ‘‘Fourier Transforms in Place’’, for the method and applications see

Gentleman and Sande, Proc. Fall Joint Computer Conference (1966).

and further references are

Cooley and Tukey, Math. Computation, **19**, 296 (1965).

Foreman.M.L.,J. Opt. Soc. Amer., **56**, 978 (1966).

Scudder.M.J., G. E. Report No. 66-C-429 (1966).