ESD-TR-67-462



ESD RECORD COPY RETURN TO SCIENTIFIC & TECHNICAL INFORMATION DIVISION (ESTI), BUILDING 1211

ESD ACCESSION LIST ESTI Call No AL 57645

Technical Note

Three Fortran Programs that Perform the Cooley-Tukey Fourier Transform N. M. Brenner

1967-2

28 July 1967

Prepared under Electronic Systems Division Contract AF 19(628)-5167 by

Lincoln Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Lexington, Massachusetts



AD657019

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology, with the support of the U.S. Air Force under Contract AF 19(628)-5167.

.

1

This report may be reproduced to satisfy needs of U.S. Government agencies.

This document has been approved for public release and sale; its distribution is unlimited.

ERRATA SHEET

for Technical Note 1967-2

Because of unclear printing in Technical Note 1967-2 (N.M. Brenner, "Three Fortran Programs that Perform the Cooley-Tukey Fourier Transform, " 28 July 1967), the distinction between + and * was often lost. A list of clarifications follows on the attached sheets.

> Publications M.I.T. Lincoln Laboratory P. O. Box 73 Lexington, Massachusetts 02173

7 September 1967



THE FOLLOWING THREE PATTERNS OCCUR FREQUENTLY. BR=WR*AR-WI*AI BI=AI*WR+AR*WI DATA(J)=DATA(I)-TEMPR DATA(J+1)=DATA(I+1)-TEMPI DATA(I)=DATA(I)+TEMPR DATA(I+1)=DATA(I+1)+TEMPI INDEX2MAX=INDEX1+N1-N2 P. 15, L. 7 ISTEP=2*MMAX P. 21, L. 2 AND P. 17, L. 2 NTOT=NTOT*NN(IDIM) P. 22, L. 5-2 AND P. 17, L. 100-2 NP2=NP1*N P. 22, L. 12 AND L. 51 12 OR 51 NTWO=NTWO+NTWO P. 22, L. 70+2 I1RNG=NP1 IF(IDIM-4)71,100,100 P. 23, L. 72+1 I1RNG=NP0*(1+NPREV/2) P. 23, L. 120 AND P. 17, L.110 110 OR 120 I1MAX=I2+NP1-2 P. 23, L. 120+3 AND P. 17, L. 110+3 J3=J+I3-I2 P. 23, L. 200 200 NWORK=2*N

> P. 23, L.210-1 IF(ICASE-3)210,220,210

4

8

7

2

P. 23, L. 240+1 J=J+IFP1 IF(J-I3-IFP2)260,250,250

P. 24, L. 420+1 AND P. 18, L. 420+1 KMIN=IPAR*M+I1

P. 24, L. 440 AND P. 18, L. 440 440 KDIF=IPAR*MMAX 450 KSTEP=4*KDIF

> P. 24, L. 520+1 AND P. 18, L. 520+1 KMIN=4*(KMIN-I1)+I1KDIF=KSTEP IF(KDIF-NP2HF)450,450,530

P. 25, L. 550+1 AND P. 19, L. 550+1 WR=(WR+WI)*RTHLF



P. 25, L. 560+2 AND P. 19, L. 560+2 WI=(TEMPR+WI)*RTHLF P. 25, L. 570+2 AND P. 19, L. 570+2 MMAX=MMAX+MMAX P. 26, L. 650+2 J2RNG=IFP1*(1+IFACT(IF)/2) P. 26, L. 655-2 I=1+(J3-I3)/NP1HF P. 26, L. 665 665 ICONJ=1+(IFP2-2*J2+I3+J3)/NP1HF P. 27, L. 670+1 TEMPI=SUMI SUMR=TWOWR*SUMR-OLDSR+DATA(J) SUMI=TWOWR*SUMI-OLDSI+DATA(J+1) OLDSR=TEMPR OLDSI=TEMPI J=J-IFP1 IF(J-JMIN)675,675,670 675 TEMPR=WR*SUMR-OLDSR+DATA(J) TEMPI=WI*SUMI WORK(I)=TEMPR-TEMPI WORK (ICONJ) = TEMPR+TEMPI TEMPR=WR*SUMI-OLDSI+DATA(J+1) TEMPI=WI*SUMR WORK(I+1)=TEMPR+TEMPI WORK (ICONJ+1) = TEMPR-TEMPI P. 27 L. 690+2 I2MAX=I3+NP2-NP1 P. 27, L. 710-2 JMIN=2*NHALF-1 P. 28, L. 740 740 NP2=NP2+NP2 P. 28, L. 745-1 IMAX=NTOT/2+1 745 IMIN=IMAX-2*NHALF P. 28, L. 805+1 I2MAX=I3+NP2-NP1 P. 28, L. 805+3 IMIN=I2+I1RNG IMAX=I2+NP1-2 JMAX=2*I3+NP1-IMIN P. 28, L. 810 810 JMAX=JMAX+NP2 820 IF(IDIM-2)850,850,830 830 J=JMAX+NP0 P. 28, L. 840 840 J=J-2

P. 28, L. 860

J=J-NP0

860



MASSACHUSETTS INSTITUTE OF TECHNOLOGY LINCOLN LABORATORY

THREE FORTRAN PROGRAMS THAT PERFORM THE COOLEY-TUKEY FOURIER TRANSFORM

N. M. BRENNER

Group 31

TECHNICAL NOTE 1967-2

28 JULY 1967

LEXINGTON

4

MASSACHUSETTS



ABSTRACT

This note describes and lists three programs, all written in USASI Basic Fortran, which perform the discrete Fourier transform upon a multidimensional array of floating point data. The data may be either real or complex, with a savings in running time for real over complex. The transform values are always complex and are returned in the array used to carry the original data. The running time is much shorter than that of any program performing a direct summation, even when sine and cosine values are precalculated and stored in a table. For example, on a CDC 3300 with floating point add time of six microseconds, a complex array of size 80 × 80 can be transformed in 19.2 seconds. Besides the main array, only a working storage array of size 160 need be supplied.

Accepted for the Air Force Franklin C. Hudson Chief, Lincoln Laboratory Office



This note describes and lists three programs, all written in USASI Basic Fortran, which perform the discrete Fourier transform upon a multi-dimensional array of floating point data. The data may be either real or complex, with a savings in running time for real over complex (see Timing). The transform values are always complex and are returned in the array used to carry the original data. The running time is much shorter than that of any program performing a direct summation, even when sine and cosine values are precalculated and stored in a table. For example, on a CDC 3300 with floating point add time of six microseconds, a complex array of size 80×80 can be transformed in 19.2 seconds. Besides the main array, only a working storage array of size 160 need be supplied.

The exact operation performed is called finite discrete Fourier transformation, also known as harmonic analysis or trigonometric interpolation. Given an array of data DATA(II,I2,...),

TRANSFORM(J1,J2,...) =
$$\Sigma$$
 [DATA(I1,I2,...) W1^(I1-1)(J1-1)
W2^(I2-1)(J2-1)...]

where $Wl = exp(-2\pi i/Nl)$, $W2 = exp(-2\pi i/N2)$,... and Il and Jl run from l to Nl, I2 and J2 run from l to N2, etc. The Fortran convention of subscripts beginning at one is adhered to. This summation possesses many of the properties of the more usual infinite integral

$$F(y) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x y} dx$$
.

By interpreting the subscripts modulo N1, N2, etc. and requiring the data to represent equispaced points, we can easily prove the usual properties about linearity, orthogonality, inverse transform and relationship to convolution. See Gentleman and Sande ([3], 1966).

l

There is no limit on the dimensionality (number of subscripts) of the data array. A three-dimensional transform can be performed as easily as a onedimensional transform, though in a proportionately greater time. An inverse transform can be performed, in which the sign in the exponentials is +, instead of - . If an inverse transform is performed upon an array of transformed data, the original data will reappear multiplied by NL*N2*....

The length of each dimension may be any integer, and as large as storage will permit. However, the program runs faster on composite integers than on primes, and is particularly fast on numbers rich in factors of two. For example, on the CDC 3300, the following timings for a one-dimensional transform have been calculated from the timing formula:

N	Factorization	Time for Complex Transform (sec)
4094	2 x 23 x 89	80
4095	3 ² x 5 x 7 x 13	214
4096	212	6.2
4097	17 × 241	180
4098	2 × 3 × 683	480
4099	prime	2868
4100	$2^2 \times 5^2 \times 41$	39

Calling Sequence

The listings of three programs are given in the appendices. FOURL is a subset of FOUR2, which in turn is a subset of FOURT. FOURT is the most general, accepting multidimensional arrays of any size. FOUR2 is the same speed as FOURT but accepts only complex multidimensional arrays whose dimensions are powers of two. FOURL is much slower than FOURT or FOUR2, and performs only one-dimensional transforms on complex arrays whose lengths are powers of two. FOURL is intended mainly for pedagogical purposes; it is half a page of Fortran, the others being much longer.

The calling sequences are:

the sectors

CALL FOURT (DATA, NN, NDIM, ISIGN, IFORM, WORK)

CALL FOUR2 (DATA, NN, NDIM, ISIGN)

CALL FOURL (DATA, NN, ISIGN)

In all cases, DATA is the array used to hold the real and imaginary parts of the input data and the transform values on output. The real and imaginary parts of a datum must be placed into immediately adjacent locations in storage. This is the form of storage used by Fortran IV, and may be accomplished in Fortran II by making the first dimension of DATA of length two, referring to the real and imaginary parts. If the data placed in DATA on input are real, they must have imaginary parts of zero appended. The transform values are always complex and replace the input data. Hence, the array DATA must always be of complex format.

For FOUR1, array DATA must be one-dimensional, of length NN. For FOUR2 and FOURT, it may be multidimensional. The extent of each dimension (except for the possible first dimension referring to the real and imaginary parts) is given in the integer array NN, which is of length NDIM, the number of dimensions. That is, NN(1) = N1, NN(2) = N2, etc. *

ISIGN is an integer used to indicate the direction of the transform. It is minus one to indicate a forward transform (exponential sign is -) and plus one to indicate an inverse transform (sign is +). The scale factor 1/(N1*N2*...)frequently seen in definitions of the Fourier transform must be applied by the user.

If the data being passed to FOURT are real (i.e., have zero imaginary parts), the integer IFORM should be set to zero. This will speed execution (see Timing). For complex data, IFORM must be plus one.

WORK is an array used by FOURT when any of the dimensions of DATA is not a power of two. Since FOUR2 and FOUR1 are restricted to powers of two, WORK is not needed. If the dimensions of DATA are all powers of two in FOURT, WORK may be replaced by a zero in the calling sequence. Otherwise, it must be

As usual, the first subscript varies the fastest in storage order.

supplied, a real floating point array of length twice the longest dimension of DATA which is not a power of two. In one dimension, for the length not a power of two, WORK occupies as many storage locations as DATA. If given, it may not be the same array as DATA.

Double precision versions of these programs may be obtained by changing the names to DFOURT, DFOUR2, and DFOUR1, declaring double precision all variables not beginning with the letters I, J, K, L, M or N, changing the references to COS and SIN to DCOS and DSIN and assigning the correct precision constants to TWOPI (2π) and RTHLF ($0.5^{\frac{1}{2}}$). DATA and WORK must then be double precision arrays.

Storage and Common

No common of any kind is used. An integer array of length thirty-two is used by FOURT. FOURT is about four hundred Fortran statements long, FOUR2 about one hundred and twenty and FOURL thirty-seven.

Return and Error Messages

There are no error messages, error halts or error returns in this program. If NDIM or any NN(I) is less than one, the program returns immediately.

Algorithm

A heavily modified version of the algorithm discovered independently by Danielson and Lanczos ([2], 1942), Good ([4], 1958), and Cooley and Tukey ([1], 1965) is used. The following example is an application to a one-dimensional transform of length six.

Let $w = e^{-2\pi i/6}$. The transformation is written

 $t_{0} = d_{0} + d_{1} + d_{2} + d_{3} + d_{4} + d_{5}$ $t_{1} = d_{0} + wd_{1} + w^{2}d_{2} + w^{3}d_{3} + w^{4}d_{4} + w^{5}d_{5}$ $t_{2} = d_{0} + w^{2}d_{1} + w^{4}d_{2} + w^{6}d_{3} + w^{8}d_{4} + w^{10}d_{5}$

$$t_{3} = d_{0} + w^{3}d_{1} + w^{6}d_{2} + w^{9}d_{3} + w^{12}d_{1} + w^{15}d_{5}$$

$$t_{4} = d_{0} + w^{4}d_{1} + w^{8}d_{2} + w^{12}d_{3} + w^{16}d_{4} + w^{20}d_{5}$$

$$t_{5} = d_{0} + w^{5}d_{1} + w^{10}d_{2} + w^{15}d_{3} + w^{20}d_{4} + w^{25}d_{5}$$

Straightforward computation requires 25 complex multiplications and 30 complex additions. The fast Fourier transform computes as follows:

 $u_{0} = d_{0} + d_{3}$ $u_{1} = d_{0} + w^{3}d_{3}$ $u_{2} = d_{1} + d_{4}$ $u_{3} = d_{1} + w^{3}d_{4}$ $u_{4} = d_{2} + d_{5}$ $u_{5} = d_{2} + w^{3}d_{5}$ $t_{0} = u_{0} + u_{2} + u_{4}$ $t_{1} = u_{1} + wu_{3} + w^{2}u_{5}$ $t_{2} = u_{0} + w^{2}u_{2} + w^{4}u_{4}$ $t_{3} = u_{1} + w^{3}u_{3} + w^{6}u_{5}$ $t_{4} = u_{0} + w^{4}u_{2} + w^{8}u_{4}$ $t_{5} = u_{1} + w^{5}u_{3} + w^{1}u_{5}$

which requires only 13 complex multiplications and 18 complex additions. Note that $w^3 = -1$ and $w^6 = 1$.

Such a reduction in computation can be found for any length which is a composite integer. The algebraic proof may be found in the appendix. Also, the various techniques for performing multidimensional transforms, real transforms, etc. are discussed there.

Special Cautions and Features

The finite discrete Fourier transform places three restrictions upon the data:

- 1. The data must form one cycle of a periodic function. Alternately stated, the subscripts are interpreted modulo N.
- 2. The number of input data and the number of transform values must be the same.

3. The data must be equispaced in each dimension (though, of course, the interval need not be the same for each dimension). Further, if in any dimension the input data are spaced at interval dt, the resulting transform values will be spaced from 0 to 2π(N-1)/(Ndt) at interval 2π/(Ndt) as I runs from 1 to N. By periodicity, the upper limit is identified with -2π/(Ndt) and in fact all points above the "foldover frequency" π/(Ndt) are to be identified with the corresponding negative frequency.

Those familiar with other implementations of the fast Fourier transform may be aware that the order of the data is scrambled in the course of execution. Unscrambling is performed automatically, however, and both the input and output values are placed in ordinary sequential arrangement.

Timing

Let N_{total} be the total number of points in the data array. That is, $N_{total} = N1*N2*...$ Decompose N_{total} into its prime factors, such as $2^{K2}3^{K3}5^{K5}$ Let Σ_2 be the sum of all the factors of two in N_{total} , that is, $\Sigma_2 = 2*K2$. Let Σ_f be the sum of all the other factors, $\Sigma_f = 3*K3 + 5*K5 + ...$ The time taken for a multidimensional transform is

 $T = T_0 + N_{total} [T_1 + T_2 \Sigma_2 + T_f \Sigma_f] .$

For the CDC 3300,

 $T = 3000 + N_{total} [600 + 40\Sigma_2 + 175\Sigma_f]$ microseconds.

The greater optimization apparent for factors of two is due to

- 1. The eight-fold symmetry of the trigonometric functions from 0 to 2π .
- 2. The fact that Fourier transforms of length two and four require

fewer complex multiplies than transforms of other lengths.

The above timing formula is accurate for complex data.

The use of real data (IFORM = 0) can reduce running time by as much as forty percent. On the CDC 3300, a 64×64 complex array was transformed in

6.1 seconds; a 64×64 real array took 4.2 seconds. A complex array 1500 long took 6.1 seconds, while a real 1500 array ran only 3.4 seconds.

Accuracy

The simplistic idea about accuracy is apparently correct: because the fast Fourier transform takes fewer steps in execution, less error creeps in. Gentleman and Sande ([3], 1966) show theoretically that the root-mean-square relative error is bounded by

1.06
$$N_{\text{total}}^{\frac{1}{2}} 2^{-b} \Sigma_{j} [2f_{j}]^{3/2}$$

where b is the number of bits in the floating-point fraction and f are the factors of $N_{\mbox{total}}.$

Further error is introduced in this particular program by the use of recursive generation of sines and cosines for factors of N_{total} other than two. Sines and cosines needed for factors of two are computed precisely. In actual practice, out of eleven and a half digits representable on the CDC 3300, about four were lost on long one-dimensional sequences like 1500 and 4096.

Applications

Besides all the direct uses of discrete Fourier transforms in signal processing, lens design, crystallography, seismic studies, etc., Fourier transforms find application in techniques of correlation and convolution. The principal tool here is the convolution theorem. Denoting the convolution of two discrete functions f and g by f*g

$$(f*g)_k = \Sigma_j f_j g_{k-j}$$
,

where both j and k run from 1 to N and subscripts are interpreted modulo N, and denoting the discrete Fourier transform of f by F(f), the convolution theorem states

F(f*g) = F(f) F(g).

The difficulties here are that cyclical interpretation of subscripts may not be desirable and that N may not be convenient for fastest processing via the fast Fourier transform. Appendage of zeroes to the ends of the sequences solves both problems. See Stockham ([5], 1966) and Gentleman and Sande ([3], 1966).

Examples of Use

A. FOURT

1.	Fe	prward transform of complex 50 \times 40 array in Fortran II
		DIMENSION DATA (2,50,40), WORK (100), NN (2)
		NN $(l) = 50$
		NN $(2) = 40$
		DO 1 $I = 1, 50$
		DO l J = l, 40
		DATA (1,I,J) = real part
	l	DATA $(2,I,J) = imaginary part$
		CALL FOURT (DATA, NN, 2, -1, 1, WORK)
2.	Sa	ame example as 1, but in Fortran IV
		DIMENSION DATA (50,40), WORK (100), NN (2)
		COMPLEX DATA
		DATA NN/50, 40/
		DO 1 $I = 1, 50$
		DO 1 $J = 1, 40$
	l	DATA (I,J) = complex value
		CALL FOURT (DATA,NN,2,-1,1,WORK)
3.	Sa	ame example as 2, but in double precision
		Add the following statement:
		DOUBLE PRECISION DATA, WORK
	Χ.	Change the call to:
		CALL DFOURT (DATA,NN,2,-1,1,WORK)

4. Inverse transform of real 64×32 array in Fortran IV

DIMENSION DATA (64,32), NN(2) COMPLEX DATA DATA NN/64,32/ DO l I = 1,64

- DO 1 J = 1, 32
- l DATA(I,J) = real value CALL FOURT (DATA,NN,2,+1,0,0)

CALL FOUR2 (DATA, NN, 2,+1)

B. FOUR2

Inverse transform of real 64 x 32 array in Fortran IV
DIMENSION DATA (64,32), NN(2)
COMPLEX DATA
DATA NN/64,32/
D0 l I = 1, 64
D0 l J = 1, 32
l DATA(I,J) = real value

C. FOURL

Forward transform of real array of length 2048 in Fortran II
DIMENSION DATA (2,2048)
D0 l I = l, 2048
DATA(l,I) = real part
l DATA(2,I) = 0
CALL FOURL (DATA,2048,-1)

Acknowledgments

The author's interest in the fast Fourier transform was sparked by Thomas Stockham. The original program was written by Charles Rader, and the idea for digit reversal was contributed by Ralph Alter. Additional ideas were gleaned from papers by Langdon and Sande, and Bingham.

BIBLIOGRAPHY

- Cooley, J.W. and Tukey, J.W., "An Algorithm for the Machine Calculation of Complex Fourier Series," <u>Mathematics of Computation</u>, <u>19</u>, 90 (April 1965), 297-301.
- 2. Danielson, G.C. and Lanczos, C., "Some Improvements in Practical Fourier Analysis and Their Application to X-ray Scattering from Liquids," <u>Journal of the Franklin Institute</u>, 233, April 1942, 365-380, and May 1942, 435-452.

「「「ない」」

- Gentleman, W.M. and Sande, G., "Fast Fourier Transforms for Fun and Profit," AFIPS, <u>29</u>, <u>1966 Fall Joint Computer Conference</u>, Spartan Books, Washington 1966.
- Good, I.J., "The Interaction Algorithm and Practical Fourier Analysis," <u>Journal of the Royal Statistical Society</u>, B, <u>20</u> (1958), 361-373, <u>22</u> (1960), 372-375.
- 5. Stockham, T., "High-speed Convolution and Correlation," AFIPS, <u>28</u>, <u>1966 Spring Joint Computer Conference</u>, Spartan Books, Washington.
- 6. Runge, C., <u>Zeitschrift für Math. und Physik.</u>, 48 (1903), 443, 52 (1905), 117.
- 7. Runge, C. and König, H., <u>Vorlesungen über Numerisches Rechnen</u>, 211-2**3**1, Springer, Berlin, 1924.
- 8. Wittaker, E.T. and Robinson, G., <u>The Calculus of Observations</u>, 260-284, Blackie and Son, London 1944.
- 9. Yates, F., <u>The Design and Analysis of Factorial Experiments</u>, 15, Commonwealth Agricultural Bureaux, Farnham Royal, 1937.
- 10. Box, G.E.P., Connor, L.R., Cousins, W.R., Davies, O.L. (Ed.), Himsworth, F.R., and Sillitto, G.P., <u>The Design and Analysis of Industrial</u> <u>Experiments</u>, 363-366, Oliver and Boyd, London, 1954.
- 11. Cooley, J.W., Miller, R.E., and Winograd, S., "Harmonic Analyzer," IBM Watson Research Center, Yorktown Heights, 1963.
- 12. Bergland, G.D. and Hale, H.W., "Digital Real-time Spectral Analysis," <u>IEEE Trans. on Electronic Computers</u>, EC-16 (April 1967), No. 2, 180-185.

- 13. Bowles, K.L., "Signal Processing An Untapped Major Market for the Computer Industry," University of California, San Diego, 1967.
- 14. Rader, C., Lincoln Laboratory Memorandum, 1965.
- 15. Good, I.J., "Random Motion on a Finite Abelian Group," <u>Proc. Camb. Phil</u>. <u>Soc.</u>, <u>47</u> (1951), 756-762.
- 16. Goertzel, G., "An Algorithm for the Evaluation of Finite Trigonometric Series," <u>The American Mathematical Monthly</u>, <u>65</u> (1958), 34-35.

Appendix I

Historical Sketch

In 1903 Runge published schemes for the optimal computation of twelve and twenty-four point Fourier transforms ([6]). They involved grouping and regrouping of values in a manner similar to the modern FFT. Runge's schemes are well known and appear in many works on numerical analysis, including Runge and König ([7], 1924) and Whittaker and Robinson ([8], 1944). Nevertheless, no one thought of generalizing Runge's ideas until 1942 when Danielson and Lanczos ([2]) published an optimal algorithm for N $\cdot 2^{k}$ point transforms. Their paper passed unnoticed.

Meanwhile, in 1937 Yates ([9]) had devised an algorithm for the efficient computation of the interactions of 2^n factorial experiments. This involves sums of the form

$$\mathbf{t}_{j} = \Sigma d_{i}(-1)^{i_{0}j_{0}+i_{1}j_{1}+\cdots}$$

where $i_{0}i_{1}$... and $j_{0}j_{1}$... are the binary representations of i and j. Davies <u>et al</u> extended the method to 3ⁿ experiments ([10], 1954); three years later, Good, in an abstruse paper, extended it to general factorial experiments ([4], 1958). In the same paper, Good devised analogous algorithms for N point Fourier transforms, where N is decomposable into mutually prime factors. Cooley and Tukey removed this restriction and clarified Good's argument ([1], 1965). They also wrote what was probably the first computer program to perform FFT.

Cooley and Tukey's paper sparked a resurgence of interest in the Fourier transform. Despite its indispensability in many areas of signal processing, the Fourier transform had long been avoided for reasons of long computation time. The FFT revived interest to such an extent that the IEEE Audio Transactions has devoted an entire issue to it (June 1967) and three groups have proposed implementing it in hardware ([11], 1963; [12], 1967; [13], 1967).

Appendix II

The Mathematics of the Fast Fourier Transform

Mathematical descriptions of the algorithms used in the Fast Fourier Transform subroutines will be published in the near future.

Punched decks for these three subroutines are available from J. J. Fitzgerald, J-105, or from SHARE.

Appendix III

Listing of the Fortran Subroutines

The listings of the three subroutines FOUR1, FOUR2, and FOURT are given on the following pages. All three are written in USASI Basic Fortran, and, as such are compatible with the great majority of Fortran compilers.

SUBROUTINE FOURIDATAINNISIGN) C THE COOLEY-TUKEY FAST FOURIER TRANSFORM IN USASI BASIC FORTRAN C TRANSFORM(J) = SUM(DATA(I)+W+*((J=1)*(J=1))); WHERE I AND J RUN C FROM 1 TO NN AND W = EXP(ISIGN*2*P]*SQRT(=1)/NN), DATA IS A ON C DIMENSIONAL COMPLEX ARRAY (I,E,: THE REAL AND IMAGINARY PARTS O C THE DATA ARE LOCATED IMMEDIATELY ADJACENT IN STORAGE; SUCH AS C FORTRAN IV PLACES THEM? WHOSE LENGTH NN IS A POWER OF THO, ISI C IS *1 OR *1; GIVING THE SIGN OF THE TRANSFORM, TRANSFORM VALUE C ARE RETURNED IN ARRAY DATA; REFLACING THE INPUT DATA. THE TIME C PROPORTIONAL TO N*LOG2(N); RATMER THAN THE USUAL N*2; WRITTEN NORMAN BRENNER, JUNE 1967, THIS IS THE SHORTEST VERSION	E = P GN S I S
C THE COOLEY-TUKEY FAST FOURIER TRANSFORM IN USASI BASIC FORTRAN C TRANSFORM(J) = SUM(DATA(I)+W+*((I=1))); WHERE I AND J RUN C FROM 1 TO NN AND W = EXP(ISIGN+2+P]+SQRT(=1)/NN), DATA IS A ON C DIMENSIONAL COMPLEX ARRAY (I,E., THE REAL AND IMAGINARY PARTS O C THE DATA ARE LOCATED IMMEDIATELY ADJACENT IN STORAGE; SUCH AS C FORTRAN IV PLACES THEM) WHOSE LENGTH NN IS A POWER OF TWO, ISI C IS +1 OR =1; GIVING THE SIGN OF THE TRANSFORM, TRANSFORM VALUE C ARE RETURNED IN ARRAY DATA; REFLACING THE INFUT DATA. THE TIME C PROPORTIONAL TO N+LOG2(N), RATMER THAN THE USUAL N+42, WRITTEN C NORMAN BRENNER, JUNE 1967, THIS IS THE SHORTEST VERSION	E = P QN S I S
C TRANSFORM(J) = SUM(DATA(I)+W+*(()-1)); WHERE I AND J RUN C FROM 1 TO NN AND W = EXP(ISIGN+2+PI+SGRT(=1)/N), DATA IS A ON C DIMENSIONAL COMPLEX ARRAY (I,E, THE REAL AND IMAGINARY PARTS O C THE DATA ARE LOCATED IMMEDIATELY ADJACENT IN STORAGE, SUCH AS C FORTRAN IV PLACES THEM) WHOSE LENGTH NN IS A POWER OF TWO, ISI C IS +1 OR +1; GIVING THE SIGN OF THE TRANSFORM, TRANSFORM VALUE C ARE RETURNED IN ARRAY DATA, REFLACING THE INPUT DATA. THE TIME C PROPORTIONAL TO N+LOG2(N), RATMER THAN THE USUAL N+42, WRITTEN NORMAN BRENNER, JUNE 1967, THIS IS THE SWORTEST VERSION	E - P QN B I S
C FROM 1 TO NN AND W B EXP(ISIGN*2*PI*SQRT(G1)/NN), DATA IS A ON C DIMENSIONAL COMPLEX ARRAY (I.E., THE REAL AND IMAGINARY PARTS O C THE DATA ARE LOCATED IMMEDIATELY ADJACENT IN STORAGE, SUCH AS C FORTRAN IV PLACES THEM! WHOSE LENGTH NN IS A POWER OF TWO, ISI C IS +1 OR +1; GIVING THE SIGN OF THE TRANSFORM, TRANSFORM VALUE C ARE RETURNED IN ARRAY DATA; REPLACING THE INPUT DATA. THE TIME C PROPORTIONAL TO N+LOG2(N), RATMER THAN THE USUAL N+42, WRITTEN C NORMAN BRENNER, JUNE 1967, THIS IS THE SHORTEST VERSION	E = QN S 1 S
C DIMENSIONAL COMPLEX ARRAY (I.E., THE REAL AND IMAGINARY PARTS O C THE DATA ARE LOCATED IMMEDIATELY ADJACENT IN STORAGE, SUCH AS C FORTRAN IV PLACES THEM) WHOSE LENGTH NN IS A POWER OF TWO, ISI C IS +1 OR +1, GIVING THE SIGN OF THE TRANSFORM, TRANSFORM VALUE C ARE RETURNED IN ARRAY DATA, REFLACING THE INPUT DATA. THE TIME C PROPORTIONAL TO N+LOG2(N), RATMER THAN THE USUAL N+*2, WRITTEN NORMAN BRENNER, JUNE 1967, THIS IS THE SMORTEST VERSION	QN 5 18
C THE DATA ARE LOCATED IMMEDIATELY ADJACENT IN STORAGE, SUCH AS C FORTRAN IV PLACES THEM) WHOSE LENGTH INN IS A POWER OF THO, ISI C IS +1 OR +1, GIVING THE SIGN OF THE TRANSFORM, TRANSFORM VALUE C ARE RETURNED IN ARRAY DATA, REFLACING THE INPUT DATA. THE TIME C PROPORTIONAL TO N+LOG2(N), RATMER THAN THE USUAL N++2, WRITTEN C NORMAN BRENNER, JUNE 1967, THIS IS THE SHORTEST VERSION	GN S I S
C FORTRAN IV PLACES THEM) WHOSE LENGTH NN IS A POWER OF THO, ISI C IS *1 OR *1, GIVING THE SIGN OF THE TRANSFORM, TRANSFORM VALUE C ARE RETURNED IN ARRAY DATA, REFLACING THE INPUT DATA, THE TIME C PROPORTIONAL TO N*LOG2(N), RATMER THAN THE USUAL N**2, WRITTEN C NORMAN BRENNER, JUNE 1967, THIS IS THE SHORTEST VERSION	GN 5 18
C IS +1 OR +1; GIVING THE SIGN OF THE TRANSFORM, TRANSFORM VALUE C ARE RETURNED IN ARRAY DATA; REFLACING THE INDUT DATA. THE TIME C PROPORTIONAL TO N+LOG2(N); RATMER THAN THE USUAL N+42, WRITTEN C NORMAN BRENNER, JUNE 1967, THIS IS THE SMORTEST VERSION	5
C ARE RETURNED IN ARRAY DATA; REPLACING THE INPUT DATA, THE TIME C PROPORTIONAL TO N+LOGINJ; RATHER THAN THE USUAL N+42, WRITTEN C NORMAN BRENNER, JUNE 1967, THIS IS THE SHORTEST VERSION	18
C ARE RETURNED IN ARRAY DATA; REFLACING THE INPUT DATA; THE TIME C PROPORTIONAL TO NELOG2(N); RATHER THAN THE USUAL NEE2, WRITTEN C NORMAN BRENNER; JUNE 1967; THIS IS THE SHORTEST VERSION	1.0
C PROPORTIONAL TO NELOGE(N), MAIMER THAN THE USUAL NERAL WRITTEN C NORMAN BRENNER, JUNE 1967, THIS IS THE SHORTEST VERSION	10 14
C NORMAN BRENNER, JUNE \$967, THIS IS THE SHORTEST VERSION	BY
C OF FFT KNOWN TO THE AUTHOR, AND IS INTENDED MAINLY FOR	
C DEMONSTRATION, PROGRAMS FOUR2 AND FOURT ARE AVAILABLE THAT RUN	
C TWICE AS PAST AND OPERATE ON MULTIDIMENSIONAL ARRAYS WHOSE	
C DIMENSIONS ARE NOT RESTRICTED TO POWERS OF TWO. (LOOKING UP SI	NES
C AND COSINES IN A TABLE WILL OUT RUNNING TIME OF FOURI BY A THIR	0.)
C SEG IFEE AUDIO TRANSACTIONS CJUNG 1967), SPECIAL ISSUE ON FFT	
DIMENSION DATA(1)	
Nage NN	
na E tel N.2	
1 F 1 [J 4 6 5 5 4 6	
	1
DATASJEDATASSE17	
- ALALIATACA	
2 Man/a	
J IF(J=M)>2>2	
4 IJRJ=0	
Mah/2	
IL(W=5)24243	
M+Uselline	
MMAX#2	
6 [F(MMAX=N)7,9,9	
7 İSTEPRZAMMAX	1
DO 8 Mª1. MMAX. 2	
THETA#3.1419265354FLQAT(\$8:GN+(M+1)}/FLQAT(MMAX)	
HR=COS(THETA)	
IW2=52N2PNETA2	
5 6 F 1 1 1 1 1 1 5 5 5 5	1
DO O SAMANAJOTER	
IJERAMMAX	
IDU O JERINIJOTEP IJETOMMAX TEMPRENRODATA(J)=NIODATA(J+1)	
DU G IMINIJOTEP JJIJETOMMAX "TEMPREWRODATA(J)=WIODATA(J=1) "TEMPREWRODATA(J=1)=DATA(J=1)	
DQ G SMANASOTEP JSTAMAX TEMPREWRODATA(J)=WIODATA(J=3) TEMPREWRODATA(J=5)=WIODATA(J=3) DATA(J)=DATA(J=5)=WIODATA(J=3) DATA(J)=DATA(J)=TEMPR	
DQ G SMANJOTEP JJSTAMMAX TEMPREWRODATA(J)=WIODATA(J=1) TEMPSEWRODATA(J+1)=WIODATA(J=1) DATA(J)=DATA(S)=TEMPR DATA(J+1)=DATA(S)=TEMPS	
DQ G SMANJOTEP JSTOMMAX TEMPREWREDATA(J)=WISDATA(J+1) TEMPSWREDATA(J+1)=WISDATA(J+1) DATA(J)=DATA(S)=TEMPR DATA(J=DATA(S+1)=TEMPS DATA(J=DATA(S)=TEMPR	
DQ G 2 MANIA PTEP JJ# 1 MMAX TEMPREWR®DATA(J)=W1®DATA(J+2) TEMPREWR®DATA(J+2)=W1®DATA(J+2) DATA(J)=DATA(J+2)=W1®DATA(J) DATA(J=DATA(1)=TEMP1 DATA(J=S)=DATA(1)=TEMP1 DATA(J=S)=DATA(1)=TEMP1	
DQ G 2 MT2N230TEP J#10MMAX TEMPREWR®DATA(J)=W10DATA(J+2) TEMPREWR®DATA(J+2)=W10DATA(J+2) DATA(J)=DATA(1)=TEMPR DATA(J+2)=DATA(1)=TEMPR DATA(1)=13=DATA(1)=TEMPR B IDATA(1)=13=DATA(1)=TEMPR MMAYERTETE	
DQ 0 2 MAX J# 1 0 MAX TEMPREWROATA(J) 0 MI0DATA(J 1) TEMPREWROATA(J 0) 0 MI0DATA(J 1) DATA(J) 0 DATA(J 0) 0 MTA(J 1) DATA(J 0) 0 DATA(J 1) 0 TEMPR DATA(J 0) 0 ATA(J 1) 0 TEMPR 0 ATA(J 1) 0 DATA(J 1) 0 TEMPR 0 ATA(J 1) 0 ATA(J 1) 0 TEMPR 0 ATA(J 1) 0 ATA(J 1) 0 TEMPR 0 ATA(J 1) 0 ATA(J 1) 0 TEMPR	
IDQ 0 2 MINISTORP IJ# 1 + MMAX TEMPREWR*DATA(J) = W1 = DATA(J = 1) TEMPREWR*DATA(J) = W1 = DATA(J = 1) DATA(J) = DATA(J = 1) = TEMPR DATA(J) = DATA(1) = TEMPR IDATA(1) = TEMPR<	

	Sheunnitue Lanuelbeiveuntuntus behaut
	THE COOLEY-TUKEY FAST FOURIER TRANSFORM IN USASI BASIC FORTRAN
	TRANSFORM(J1+J2+) = SUN(DATA(11+12+)+W2++((11+1++))
	LUBBE IN AND IL DIN FRAM I TA NN/IL AND ULEVALTANDEL
	WHERE IS AND US RUN FROM ST IN MILES AND REPERTS FOUNDARY
	SURTVALIANNALIA ETG
	DATA IS A MULTIDIMENSIONAL FLOATING ROINT ARRAY ALL OF WHOSE
	DIMENSIONS ARE POWERS OF TWO, THE LENGTH OF EACH DIMENSION IS
	STORED IN THE INTEGER ARRAY NN, OF LENGTH NDIM, ISION IS
	+1 OR 1.4 GIVING THE SIGN OF THE TRANSFORM. THE REAL
	AND IMAGINARY PARTS OF A DATUM ARE IMMEDIATELY ADJACENT IN STORAG
	(SUCH AS FORTRAN IV PLACES THEM), TRANSFORM RESULTS ARE RETURNED
	IN ARRAY DATA, REPLACING THE ORIGINAL DATA, TIME IS PROPORTIONAL
	TO N*LOGR(N), RATHER THAN THE USUAL N**2, NOTE THAT IF A FORWARD
	TRANSFORM IS FOLLOWED BY AN INVERSE TRANSFORM, THE ORIGINAL DATA
- Contraction	WILL REAPPEAR MULTIPLIED BY NN(1)*NN(2)* EXAMPLE *
÷	FORWARD FOURIER TRANSFORM OF A TWO-DIMENSIONAL ARRAY IN FORTRAN I
	DIMENSION DATA(2,64,32), NN(2)
in the	NN(1)=64
	NN(2)=32
	DD 1 1=1:64
,	DO 1 J=1:32
	DATA(1/J/J/REAL PART
1	DATA(2/1+J)FIMAGINARY PART
	CALL FOUR2(DATA, NN, 2, •1)
	SAME EXAMPLE IN FORTRAN IV
	DIMENSION DATA(64, J2), NN(2)
	COMPLEX DATA
	DATA NN/64,32/
1.77	DO 1 1*1+64
	DO 1 J=1+32
1	DATA(I)J)=COMPLEX VALUE
	CALL FOUR2(DATA, NN. 2. 1)
	PROGRAM BY NORMAN BRENNER FROM THE BASIC PROGRAM BY CHARLES
	RADER, MAY 1967, THE IDEA FOR THE DIGIT REVERSAL WAS SUGGESTED
	BY RALPH ALTER:
	THIS VERSION OF THE FAST FOURIER TRANSFORM IS THE FASTEST KNOWN
	TO THE AUTHOR. LOOKING UP STARS AND COSTARS IN A TARUE INSTEAD OF
	COMPUTING THEM WOULD DECREASE RUNNING TIME SEVEN REROENT.
	DOADDING CAURT AND FAURI ARE AVAILARIS FROM THE AUTHOR THAT ALSO
	DEPENDENTING FAST FOURIER TRANSPORM AND ARE WRITTEN IN USASI RASIC
	COPTRAN. FOURT IS THREE TIMES AS LONGA IS NOT RESTRICTED TO
	DOWERS OF THO. AND RUNS UP TO FORTY PERCENT FASTER ON REAL DATA.
	FOUR1 IS ONE FOURTH AS LONG, ONE HALF AS FAST, AND IS RESTRICTED
	TO ONE DIMENSION AND POWERS OF THO.
	SEE LERE AUDIO TRANSACTIONS (JUNE 1967), SPECIAL ISSUE ON FFT.
	DIMENSION DATA(1), NN(1)
	IF(ND[M=1)700,1,1
	NTOT=2
	DO 2 IDIMEL/NDIM

	NTOTALITOTALNI () DIMA
4	NIGLANIGLANNSIDIAL
	RTHLF#,70710 67812
	TWOP1=6228318 53070
1	
~	MAIN LOOP FOR FACH DIMENSION
6	HAIN LOU LACH DIRENSION
C.	
-	NP1#2
	DO 600 IDIMES NDIM
	NENNS JUIN/
	NPZENPIAN
	IF(N-1)700,600,100
0	
č	CHIEFLE DATA DY RIT REVERSAL, SINCE NERSER, AS THE SHIFFLING
v.	THE PART OF STATES INTERCUINCE, NA UNBEING IDEAU TO NEEDED
C	CAN BE DONE BY STAFLE INTERVANCES NO HORKING ARRAT IS NEEDED
Ç	
100-	NP2HF=NP2/2
	1=1
	0-4 148 19-1 NO2 NO1
	DO TOO TEATMART
	16(1-15)110)130
110	11MAX#12*NP1"2
	DO 120 I1=12+11MAX+2
	DO 120 13-11-NTOT-NP2
	TEMPREDATA(IS)
	TEMPI=DATA(I3+1)
	DATA(13) BATA(J3)
	DATA(1341) BDATA(1341)
	DATAJJJTEMPR
120	DATA(J3+1)=TEMPI
130	MENP2HF
140	ff(j=M)160,160,150
150	
230	
	1F(M=NP17100)1400140
160	M + U = U
C	
10	MAIN LOOP, PERFORM FOURIER TRANSFORMS OF LENGTH FOUR, WITH ONE OF
1.0	LENGTH THE IF NEEDED. THE THIDDLE FACTOR HEEVELISION-PARTA
	LENGIN IN IT REEDEDI THE INTOLE FROM A REPAIL OF THE PLANE AND THE AND THE REPAIL OF THE PLANE AND T
0	SORIGELS MALANAMAXII CHECK FOR THE SPECIAL CASE WEISIGN SWRITTELS
C	AND REPEAT FOR WAW+(1+ISIGN+SGRT(+1))/SGRT(2)
C	
· .	NP(TWENP1+NP1
	IFAR N
310	IFTIPAR023300,330,320
320	1PAR=1PAR/4
	GO TO 310
338	DO 340 11=1,NR1,2
	DO 340 MARTI NTOT NRITH
	DO GAD REPORT OF A CALL
	RZ=KI=NPI
	TEMPRADATA(KZ)
	TEMPIEDATA(K2+1)
	DATA(K2)=DATA(K1)=TEMPR
	DATA (KPAS JEDATA (KIAI) - TEMPI
	DATAINA/TDATAINA/TTENTA
340	DATASK1+1PDATASK1+1PTEMPI
350	MMAX*NP1
360	TF(MMAX=NP2HF)370,600,600
344	I MINEMANA/ND19W.MMAY/21
010	
	DO DYO LENKIALMAKANYI W
	Mac
	IF(MMAX=NP1)420,420,380
TRA	TUSTA-TUODITCIOAT(N)/FLOAT(ASNNAX)

TP(1\$10N)400+390+390
TAA PURTARATHETA
HISTN(THETA)
410 W2R=WR=WI=WI
W21#2,*WR+WI
WJR=WZR+WR=W21+W1
W31=W2R+W1+W21+WR
420 DO 530 [1=1,NP1,2
KHINFIPAREM+11
IF CMMAARNP1 / 30 / 30 / 470
DO 520 KLEKMINANTOTAKSTEP
K2=K1+KDIF
KJ=KB+KDIF
KeskaskDir
IF (MMAX=NP22460,460,460
460 USREDAJALKS/ DATALKS/
ULI PURIAL KEYA JUNTAL KEYA J
HISTADATA (W3+1 S+DATA(K4+1)
USBADATA(K1)=DATA(K2)
USI + DATA(KE+1) = DATA(K2+1)
[P(1818N)470+475+475
470 UARBOATAIKE+LINDATA(K4+L)
U43=DATA(K43=DATA(K3)
GO TO PLD
INTERPRIATE DATALAT
480 T28-W28+DATA(K2)+W21+DATA(K2+1)
T21=W2R+DATA(K2+1)+W21+DATA(K2)
TJR=WR*DATA(K3)=WI*DATA(K3+1)
TJINKADATA(KI+13+WI+DATA(NJ)
Tennamedata(Ke)=Maledata(Me+2)
TAISHARADATALKASLISHNJISUATALKAJ
ULR-DATACKA/TIAN
USBETSROTAR
1021#731#741
USRODATAIKE)=T2R
UJI=DATA(K2+1)=T21
1P(151GN)490+500+500
1800 HARRTALATSI
UATATARETAR
BLO DATALKLIBUER+UER
DATAIKEPE)=ULI=UZI
DATACKEPPUJRPUAR
TER DATALKAALJAUSTAUAL
HMENDA+ CHASH-12. HEL

830	CONTINUE	111
	MENALMAX	
	TF(H=MMAX)540,540,97	0
540	IF(18)0N1550.560.560	
550	TEMPRENR	
	WR#(WR#WE)#RTHLF	
	WININI-TEMPRINCP	
	80 10 410	*
560	TEMPROWR	
÷ 11	WRACHRANTSCRIPT	
	WSSCTEMPROWICHTHLF	
	GO TO 410	
970	CONTINUE	
	MMAXEMMAX	
400	60 10 300	
200		
100	A DOWN	

	SUBROUTINE FOURT(DATA:NN:NDIM: SECON: FORM: WORK)
C	THE COOLEY-TUKEY FAST FOURTER TRANSFORM IN USASI BASIC FORTRAN
0	TRANSFORM(JI)J/J// W BUMCDATALLA (K / JANK W C (1281 / 4 U) - 17
	HIMTOR TE AND IS DIN POAM E TA NUIS LAG DE VIETAITE 1/1
	HERE IL AND DI HUN FROM IL DU HRIET AND REFERILISIONS AND THE
	JUNITER//NNSA////CONS. INERS 30 NO. LANSI VON PRE DARGOARDATT
-	INUNDER OF SUBJECT IS OF DEPENDEN HEADER IN AN AN ARRAY OF TRANSFORMED
N	(TEGNER'S DATA, THE OFFICIAL BETA UTI REAPPEAR.
	MILTIPLIED BY NN(1)+NN(2)+ THE ARRAY OF INPUT DATA MUST BE
	IN COMPLEX FORMAT. HOWEVER, IF ALL IMAGINARY RARTS ARE ZERO (1.E.
	THE DATA ARE DISQUISED REAL! RUNNING THE IS CUT UP TO PORTY PER-
c	CENT. (FOR FASTEST TRANSFORM OF REAL DATA, NN(1) SHOULD BE EVEN.)
0	THE TRANSFORM VALUES ARE ALWAYS COMPLEX, AND ARE RETURNED IN THE
	ORIGINAL ARRAY OF DATA, REPLACING THE INPUT DATA, THE LENGTH
	OF EACH DIMENSION OF THE DATA ARRAY MAY BE ANY INTEGER. THE
C	PROGRAM RUNS FASTER ON COMPOSITE INTEGERS THAN ON PRIMES, AND IS
C	PARTICULARLY FAST ON NUMBERS RICH IN FACTORS OF TWO
C	
6	TIMING IS IN FACT GIVEN BY THE FOLLOWING FORMULA, LET NTOT BE THE
0	TOTAL NUMBER OF POINTS (REAL OR COMPLEX) IN THE DATA ARRAY, THAT
C	IS, NTOTENN(1)*NN(2)*1.1 DECOMPOSE NTOT INTO ITS PRIME FACTORS.
C	SUCH AS 2++K2 + 3++K3 + 5++K5 + 11. LET SUM2 BE THE SUM OF ALL
0	THE FACTORS OF THO IN NTOTA THAT IS, SUME + 29K2, LET SUMP BE
0	THE SUM OF ALL OTHER FACTORS OF NTOT: THAT IS: SUMF = 3483+D485+1
	THE TIME TAKEN BY A MULTIDIMENSIONAL TRANSFURM UN TRASE NTOI DATA
	IS I TO THE TO THE CONTRACT OF THE CONTRACT. THE CONTRACT OF THE CONTRACT OF THE CONTRACT OF THE CONTRACT OF T
	POINT ADD TIME . STANTS ON ADDRESS 1. SOUR . ATCT CONTACT STANTS
	TARABULA WICKOSCOURS ON CONCERS BASH
	INCLEMENTATION OF THE REFINITION BY SUMMATION WILL BUN IN A TIME
	PROPORTIONAL TO NTOTY(NN(1)+NN(2)+). FOR MIGHLY CONROSITE NTOT
	THE SAVINGS OFFERED BY THIS PROGRAM CAN BE DRAMATIC. A ONE-DIMEN-
	STONAL ARRAY 4000 IN LENGTH WILL BE TRANSFORMED IN 4000+(600+
Ċ	40+(2+2+2+2+2+175+(5+5+5)) = 14,5 SECONDS CERSUS ABOUT 4000+
C	4000+175 = 2800 SECONDS FOR THE STRAIGHTFORWARD TECHNIQUE,
0	
0	THE FAST FOURIER TRANSFORM PLACES THREE RESTRICTIONS UPON THE
0	DATA .
	1. THE NUMBER OF INPUT DATA AND THE NUMBER OF TRANSFORM VALUES
	MUST BE THE SAME,
C	2. BOTH THE INPUT DATA AND THE TRANSFORM VALUES MUST REPRESENT
2	EQUISPACED POINTS IN THEIR RESPECTIVE DOMAINS OF TIME AND
0	FREQUENCY, CALLING THESE SPACINGS DELTAT AND DELTAF, IT HUST BE
	THUE THAT DELTAP 22 PI/INNII/ DELTATI OF COURSES DELTAT NEED NUT
	C INC SAME FUN EXENT UTHENSIONS THE INDUT PERA AND THE TRANSBOOM AND THE
	DEDDESENT SINGLE OVCIES OF PERIODIC SUNCTIONS.
	LELUMARILI AFIGRE ALARMA AL LEUTARIA LAIAITANAÎ
Ē	THE CALLING SEQUENCE IS .
ē	CALL FOURT (DATA, NN, NDIM, ISIGN, IFORM, WORK)
0	
¢	DATA IS THE ARRAY USED TO HOLD THE REAL AND IMAGINARY PARTS
C	OF THE DATA ON INPUT AND THE TRANSFORM VALUES ON OUTPUT, IT
C	IS A MULTIDIMENSIONAL FLOATING POINT ARRAY, WITH THE REAL AND
C	IMAGINARY PARTS OF A DATUM STORED IMMEDIATELY ADJACENT IN STORAGE
	ACTOR AC PORTOAN AV BLACES THEM) INCOMAL PORTOAN ACTORS IN

EXPECTED, THE FIRST SUBSCRIPT CHANGING FASTEST. THE DIMENSIONS C ARE GIVEN IN THE INTEGER ARRAY INN. OF LENGTH NDIM. ISIGN 15 C TO INDICATE A FORRARD TRANSFORM (EXPONENTIAL SIGN IS -) AND -1 C FOR AN INVERSE TRANSFORM (SIGN IS +), IFORM IS +1 IF THE DATA ARE COMPLEX: O IF THE DATA ARE REAL: IF IT IS 0, THE IMAGINARY PARTS OF THE DATA MUST BE SET TO ZERO; AS EXPLAINED ABOVE, THE C C C TRANSFORM VALUES ARE ALWAYS COMPLEX AND ARE STORED IN ARRAY DATA. C WORK IS AN ARRAY USED FOR WORKING STORAGE, IT IS FLOATING POINT C REAL, ONE DIMENSIONAL OF LENGTH EQUAL TO TWICE THE LARGEST ARRAY DIMENSION NN(1) THAT IS NOT A POWER OF THO, IF ALL NN(1) ARE C Ċ POWERS OF TWO, IT IS NOT NEEDED AND MAY BE REPLACED BY ZERO IN THE C CALLING SEQUENCE, THUS, FOR A ONE-DIMENSIONAL ARRAY, INN(1) ODD, C WORK OCCUPIES AS MANY STORAGE LOCATIONS AS DATA, IF SUPPLIED, Work must not be the same array as data, all subscripts of all 10 C ARRAYS BEGIN AT ONE. C C THREE-DIMENSIONAL FORWARD FOURIER TRANSFORM OF A C EXAMPLE 1. COMPLEX ARRAY DIMENSIONED 32 BY 25 BY 13 IN FORTRAN IV. C DIMENSION DATA(32,25,13), HORK(50), NN(3) 10 COMPLEX DATA C DATA NN/32,25,13/ C C D0-1-1-1-32 DO 1 J=1:25 C C DO 1 K=1:13 1 DATA(1.J.K)=COMPLEX VALUE : C CALL FOURT (DATA, NN, 3, -1, 1, WORK) C C EXAMPLE 2. ONE-DIMENSIONAL FORWARD TRANSFORM OF A REAL ARRAY OF LENGTH 64 IN FORTRAN II. Ĉ C C DIMENSION DATA(2,64) C DO 2 1=1+64 DATA(1,1)=REAL PART 2 DATA(2,1)=0, :0 C CALL FOURT (DATA, 64, 1, -1, 0, 0) C 0 THERE ARE NO ERROR MESSAGES OR ERROR HALTS IN THIS PROGRAM. THE PROGRAM RETURNS IMMEDIATELY IF NDIM OR ANY NN(1) IS LESS THAN ONE. C CC PROGRAM BY NORMAN BRENNER FROM THE BASIC PROGRAM BY CHARLES C RADER, JUNE 1967, THE IDEA FOR THE DIGIT REVERSAL WAS SUGGESTED BY RALPH ALTER. C C THIS IS THE FASTEST AND MOST VERSATILE VERSION OF THE FFT KNOWN =C TO THE AUTHOR, A PROGRAM CALLED FOURS IS AVAILABLE THAT ALSO C PERFORMS THE FAST FOURIER TRANSFORM AND IS WRITTEN IN USASI BASIC Ċ FORTRAN, IT IS ABOUT ONE THIRD AS LONG AND RESTRICTS THE DIMENSIONS OF THE INPUT ARRAY (WHICH MUST BE COMPLEX) TO BE FOWERS :C 'C OF THO, ANOTHER PROGRAM, CALLED FOUR1, IS ONE TENTH AS LONG AND C RUNS THO THIRDS AS FAST ON A ONE-DIMENSIONAL COMPLEX ARRAY WHOSE Ċ LENGTH IS A POWER OF TWO, :0 Ç REFERENCE --:0 TEEE AUDIO TRANSACTIONS (JUNE 1967), SPECIAL ISSUE ON 'THE FPT. 10 C DIMENSION DATA(1), NN(1), [FACT(32), WORK(1) TWOP1=6,283185307 RTHLF=.70710 67812 1F(NDIMEL)920,1,1 NTOT=2 1 DO 2 1DIM=1, NDIM IF(NN(IDIM))920,920,2 NTOT=NTOT=NN(IDIM)

0	
-	NATH LODD FOR FACH DIMENSION
	Lift Pack to med Stradator
	ND4 # 2
	NP4"
	Newissing
	NF62NF62N
- (6)	1.114-91200120020
-0	TS N & DOLED OF THE AND TE NOT. WHAT ARE TTS FACTORS
	to it w comparison of the weather the thereas
	Meda
1	1910
	5 F # & a mac / # 2
1.0	
10	
	10170-10170-10171-10001
4.4	17 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
++	
76	
/	18841844 17 NO 1147 (8400)
	ir air an Anna Anna Anna Anna Anna Anna Anna
: 9 R	
3.0	tourself for V
9.9	
	3 N 10 N 17 N 3 N 3 N 3 N 3 N 3 N 3 N 3 N 3 N 3 N
	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
30	fracerite and tv
· w 56	TRATES
	MarculoT
	AO TO 30
40	10TV=10TV=2
	00 10 30
50	INONZETE
	TF(1REM)60,51,60
51	NTWO=NTWO+NTWO
	100 TO 70
60	fpact(fp)=M
0	
:¢	SEPARATE FOUR CASES?*
0	1. COMPLEX TRANSFORM OR REAL: TRANSFORM FOR THE 4TH . STH SET .
:C	DIMENSIONS
C	2. REAL TRANSFORM FOR THE 2ND OR 3RD DIMENSION, METHOD .
10	TRANSFORM HALF THE DATA: SUPRLYING THE OTHER HALF BY DON-
10	UUQATE SYMMETRY:
°C	3, REAL TRANSFORM FOR THE IST DIMENSION, N ODD, METHOD **
1 C	SET THE IMAGINARY PARTS TO ZERO,
0	4. REAL TRANSFORM FOR THE SET DIMENSION, N HVAN, METHODA
C	TRANSFORM A COMPLEX ARRAY OF LENGTH N/2 WHOSE REAL PARTE
0	AND THE EVEN NUMBERED REAL VALUES AND WHOSE FRADERARY BARTS
10	AND THE ODD NUMBERED INEAL VALUES. ISPARATE AND BUPPLY
C	THE SECOND HALF BY CONJUGATE STANKETRY
C	
70	
	王王明王刘帝承
	164103444432747657266

-	A A A A A A A A A A A A A A A A A A A	 A second s
1.0	1211178172121200	
100	A D A D D A D	
1.6		
	140000000000011400000012	
	P S M IS M M IS P S S S S S TO COM S S S S S S S S S S S S S S S S S S S	
	TFCIDIMOL 173/73/100	
1 100 100	**********	
13	1.CV main a	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
	** BADANDE	
	19000000	
	*##NTWO#W##\$\$\$00,500,74	
7.6		
2.000	A REAL PROPERTY OF THE PROPERT	
	107000000000000000000000000000000000000	
	HIT WW-HIT HWF W	
	NeW/2	
	IN DIAL OF	
	NP2#NP2ZZ	
		1
	NTOTENIUTER	
	DO SO USL NTOT	
	DATARJJUDATALI	
	1.1.1.0	
00	2.02.04	
100		
M	second mining of the second	and the second
6		
1.00	ALL OF BANG ON CLUBIC INCOM	MANDE, NO MADE MO ADDAM TO MERCH
		TATALAN A AND L'ANG, L'AND WAL AN INTAL RADAR & AND L. 1 11. 21
18		
100	ILINIMOGNEESEOAF7785770	and the second
4 4 8	N0012-5002/0	· · · · · · · · · · · · · · · · · · ·
110	NEEDLANEEVE	
	(通知機) 一方方 古人 人名法尔尔 人名	
	Was in the same west	State of the second
to the second		
420	14447412448449	
	1 BURNEL PROVIDE BURNEL	
	DO 129 LLUESILSMAXIS	
	A	
	DO TWO IGORYTHNIOLFIGHT	
	13844453689	
		and the second
	N274182200274133	
que a		and the second
	IDATAS SOURCE SOURCE SOUTH STATE	
		the second s
	DATANJAITTENT	
496	navit iset tuybupt	
	MULANTAL LAURA	
130		
240	111000010000000000000000000000000000000	and the second
		the standard
943	G4970	
	MAMZ	
	2014 Mark Bild BO . 4 40 . 4 40	그는 것 같은 것 같
	INTHUMPTIZAATAATAA	그는 것 같은 것 같은 것 같은 것 같은 것 같은 것 같이 없다.
150	Lin in M	
- 10 H V		
	00 10 300	
3.100	00 10 300	
Ģ	00 10 300	
10 10	SHUFFLE DATA BY DIGIT REVERS	AL FOR IGENERAL IN
Ċ	SO TO SPE Shuffle Data by Digit Reveal	AL FOR ISBNERAL IN
C	GO TO PPE SHUFFUE DATA BY DIGIT RECENT	AL FOR IGENERAL IN
C	SO TO BOS SHUFFLE DATA BY DIGIT REVEN	AL FOR IGENERAL IN
500 50 50 50 50 50 50 50 50 50 50 50 50	NWORKSEN	AL FOR IGENERAL IN
500 500 500	SO TO SOC SHUFFLE DATA SY DIGIT REVERI NWORK#SEN DD 270 (151.NP1.2	AL FOR IGENERAL IN
500 C	NHORKUSAN DD 270 (1914) NP1.2	AL FOR IDENERAL IN
500 C	GO TO PPO SHUFFUE DATA BY DIGIT REPERI NWORKWEAN DD 270 [istanp1/2 DD 270 [istanp1/2 DD 270]attantot.np2	AL FOR IGENERAL IN
500 C	00 TO 300 SHUFFLE DATA BY DIGIT REVERS NWORKW20N DO 270 [isiaNP1/2 DO 270 [isiaNP1/2 DO 270]isiaNP1/2	AL FOR IGENERAL IN
500 C	IGO TO POP SMUFFLE DATA BY DIGIT REPERI NWORK#8+N DD 270 [isi/NP1/8 DD 270 [isi/NP1/8 JD 270 [isi/NP1/8 JJ 13	AL FOR IGENERAL IN
500 C C	00 TO 300 SHUFFLE DATA BY DIGIT REVERS NWORK#20N DD 270 [isi/NP1/2 D0 270 [isi/NP1/2 D0 270 [isi/NP1/2 J#13 D0 260 [fi/NWORK#2	AL FOR IGENERAL IN
500 500	00 TO 300 SHUFFLE DATA SY DIGIT REQUAL NWORK#20N DD 270 [isi/NP1/2 DD 270 [isi/NP1/2 J0 270 [isi/NP1/2 J0 260 [si/NHORK/2	AL FOR IGENERAL IN
500 500	00 TO 300 SHUPPLE DATA BY DIGIT REVERS NWORK#2*N DD 270 [isi/NP1/2 DD 270 [isi/NP1/2	AL FOR IGENERAL IN
C C 200	00 TO 300 SHUFFLE DATA BY DIGIT REVEAU NWORK#20N DO 270 [isi.NP1.2 DO 270 [isi.NP1.2 DO 270] To 1.NP1.2 J#13 DO 260 [FL:NWORK:2 IF([CASE#3/250/280,210 WORK(1) #8474(1)	AL FOR IGENERAL IN
C C 200	00 TO 300 SHUFFLE DATA BY DIGIT REVEAU NWORK#20N DD 270 [isi/NP1,2 DD 270 [isi/NP1,2	AL FOR IGENERAL IN
200	00 TO 300 SHUFFUE DATA BY DIGIT REVERS NWORKW20N DO 270 [isi/NP1/2 DO 270 [isi/NP1/2 DO 270 [isi/NP1/2 DO 270 [isi/NP1/2 DO 260 [si/NWORK/2 IF(ICASES)210/220,210 WORK([)00ATA(]) WORK([)520DATA(])	AL FOR IGENERAL IN
210	GO TO 300 SHUFFLE DATA BY DIGIT REVEAU NWORK#20N DD 270 [iss.NP1,2 DO 270 [iss.NP1,2	AL FOR IGENERAL IN
200	00 T0 300 SHUFFLE DATA SY DIGIT REVEAU NWORKW20N D0 270 [isiaNP1/2 D0 270 [isiaNP1/2 D0 270 [isiaNP1/2 J#13 J#13 D0 260 [siaNNORK:2 IF([cases]210:220:210 WORK([:siaDATA(j:1) G0 T0 330	AL FOR IGENERAL IN
210	00 TO 300 SHUFFLE DATA SY DIGIT REVEAL NWORK#20N DD 270 [iss.NP1.2 DD 270 [iss.NP1.2	AL FOR IGENERAL IN
2 2 2 2 2 0 0 2 2 0 0 2 2 0 0 2 2 0 0 2 2 0 0 2 2 0 0 2 2 0 0 2 2 2 0 0 2 2 2 0 0 2 2 2 0 0 2 2 0 2 2 0 0 2 2 0 0 2 2 0 0 2 2 0 0 2 2 0 0 2 2 0 0 2 2 0 2 2 0 0 2 2 0 2 2 0 2 2 2 0 2 2 2 0 2	00 TO 300 SHUFFLE DATA SY DIGIT REVEAU DD 270 [isiaNP1/2 DD 270 [isiaNP1/2 DD 270] StiaNP1/2 DD 260 [siaND07,NP2 Jat3 DD 260 [siaND07,NP2 IF([CASE(3)210/220,210 WORK(]) SDATA(J) WORK(]) SDATA(J) WORK(]) SDATA(J)	AL FOR IGENERAL IN
C C 200 210	GO TO 300 SHUFFLE DATA SY DIGIT REVEAL NWORK#8*N DD 270 [1=1,NF1,2 DD 270]3=11×NTOT,NP2 J#13 D0 260]=1,NWORK;2 IF([CASE#37250;210 WORK(]*5)=DATA(J) WORK(]*5)=DATA(J) WORK(]*5]=DATA(J) WORK(]*5]=D	AL TOR IGENERAL IN
220 2210	00 T0 300 SHUFFLE DATA BY DIGIT REVEAU D0 270 [isi/NP1/2 D0 270 [isi/NP1/2 D0 270 [isi/NP1/2 D0 260 [fi/NNORK:2 If(ICASE\$3)210/220/210 WORK(])8BATA(J) WORK(])8BATA(J) WORK(])19BATA(J) WORK(])19BATA(J) WORK(])19BATA(J) WORK(])19BATA(J)	AL <u>for iobnural in</u>
2200 2210 2220 2220	00 T0 300 SHUPPLE DATA SY DIGIT REVEAL NWORKWSEN DD 270 [1.5.,NP1,2 D0 270 [3:1.ND7,NP2 J#13 D0 260 [3:1.ND7,NP2 J#14 D0 260 [3:1.ND7,NP2 J#15 D0	AL FOR IGENERAL IN
220 2210 2220 2230	00 TO 300 SHUFFLE DATA BY DIGIT REVEAU DD 270 (isi/NP1/2 DD 270 (isi/NP1/2 DD 270 (isi/NP1/2 DD 270 (isi/NHORK:2 IF(icases)2207280:210 WORK(i)08ATA(J) WORK(i)08ATA(J) WORK(i)108ATA(J)	AL FOR IOBNERAL IN
200 2200 2210 2220 2230	GO TO POP SHUFFLE DATA BY DIGIT REVERING NWORKWSEN DD 270 [1913:NP1;2 DD 270 [1913:ND7,NP2 J#13 DO 260 [05,NWORK;2 IF([CASE\$32107220;210 WORK(]25374(J) WORK(]25374(J	AL FOR IGENERAL IN
220 2200 2210 2220 2220 2230 2230	00 T0 300 SHUFFLE DATA SY DIGIT REVEAU DD 270 [isi/NP1/2 DD 270 [isi/NP1/2 DD 270 [isi/NP1/2 DD 270 [isi/NB0RK/2 TF([CASE\$37250/220/210 WORK(])SBATA(J) WORK(]>SBATA(J) WORK(]>SBATA(J) WORK(]>SBATA(J) WORK(]>SBATA(J) WORK(]>SBATA(J) WORK(]>SBATA(J) WORK(]>SBATA(J) WORK(]>SBATA(J) WORK(]>SBATA(J) WORK(]>SBATA(J) WORK(]>SBATA(J) WORK(]>SBATA(J) WORK(]>SBATA(J) WORK(]>SBATA(J) WORK(]>SBATA(J) S	AL FOR IOBNIRAL IN
210 220 220 220 220 220 220 220 220 220	00 T0 300 SHUFFLE DATA SY DIGIT REVEAU D0 270 [isiaNP1/2 D0 270 [isiaNP1/2 D0 270 [sianT07,NP2 J#13 J#13 D0 260 [sianT07,NP2 IF(1CASE(12002000000000000000000000000000000000	AL FOR IGENERAL IN
220 2210 2220 2220 2220 2220 2220 2220	00 T0 300 SHUFFLE DATA SY DIGIT REVEAU DD 270 [isi/NP1/2 DD 270 [isi/NP1/2 DD 270 [isi/NB/2 J#13 D0 260 [#1/NB/2 TF([CASE\$32250/280/210 WORK([)SEATA(J) WORK([)SEATA(J) WORK([)SEATA(J) WORK([)SEATA(J) WORK([)SEATA(J) WORK([)SEATA(J) WORK([SE]SEATA	AL FOR IORNIRAL IN
220 2210 2220 2230 2230 2240	00 T0 300 SHUFFLE DATA SY DIGIT REVEAU D0 270 [isi/NP1/2 D0 260 [isi/NP1/2 D0 70 230 WORK([isi/NP1/2 D0 70 230 WORK([isi/NP1/2 P0 70 230 WORK([isi/NP1/2 P	AL TORIGENERAL IN

250	jajel/P2
	trp2=ifp1
	TFATFAL
-	1F(1FP2=NP1)260,260,240
260	CONTINUE
	12MAY#134NP2=NP1
	LENARA VENDERNINE
	174 796 19413.12NAV.N01
	DATATICALITY
270	2 # 2 # 4
C	
Ç	MAIN LOOP FOR FACTORS OF THO, PERFORM FOURIER TRANSFORMS OF
C	LENGTH FOUR, WITH ONE OF LENGTH TWO IF NEEDED. THE TWIDDLE FACTOR
C	WHEXP(ISIGN+2+PI+SQRT(+1)+M/(++MMAX)), OHECK FOR WHISIGN+SURT(+1)
C	AND REPEAT FOR W#W#(1+1SIGN+SGRT(=1))/SGRT(2)1
Ċ	
300	IF (NTWO=NP1)600,600,309
305	NP1TW=NP1+NP1
	IPAR INTWO/NP1
310	TF(1PAR=2)350,330,320
320	TRAREIPAR/4
	GO TO 310
330	DO 340 11s1, 11RNG, 2
	DO 340 KIRII.NTOT.NPITW
340	DATA KATS JEDATA KATS JETTERPI
350	MMAX3NF1
360	IF(MMAX=NTW0/2/370,000,000
370	LMAXEMAXU(NPITH&MMAAK2)
	DO 570 LENPI, LMAX, NPITW
	MeL
-	IF(MMAX=NP1)=20,420,380
380	THETAS TWOPITEUDAT(L)/ELOAT(4*MMAX)
	IF(ISIGN)400,390,390
390	THETASTHETA
400	WR=COS(THETA)
	WI=SIN(THETA)
410	
	W2I=2,*WR+WI
	W3R=W2R+WR-W2I*WI
	W31=W2R+W1+W21+WR
420	DO 530 [1=1, I1RNG, 2
	KMIN=11+1PAR*M
	IF (MMAX0NP1)430,430,440
430	KMIN=11
440	KDIFFIPAR+MMAX
450	KSTEP=4*KDIF
	IF (KSTEP=NTWO)460,460,530
460	DO 520 KIEKMINANTOTAKSTEP
	K2aK1+KDIF
	K3aK2+KDIF
1.000 Juni -	K4=K3+KD1F
	15(MMAX5NP1)470,470,480
470	110 EDATA (K1) + DATA (K2)
410	VAR-MAINTAN(WINING WINING)
	ARG2801010A4.NG101011

<mark>24</mark>

a gu	U21=DATA(K3+1)+DATA(K4+1)	
	U3R=DATA(K1)=DATA(K2)	
	U31=DATA(K1+1)+DATA(K2+1)	그는 그는 것 같은 것 같은 것 같은 것 같이 많이 많이 없다.
	IF(ISIGN)471,472,472	
471	U4R=DATA(K3+1)=DATA(K4+1)	
	U41=DATA(K4)=DATA(K3)	
	GO TO 510	
472	UAR=DATA(K4+1)=DATA(K3+1)	
	DATEDATA(K3)*DATA(K4)	
	00 TO 510	
480	T288W28+DATA(K2)=W21+DATA(K2+1)	
00	T21=W2R+DATA(#2+1)+W21+DATA(#2)	
	TTORWORDATALKTANTODATALKAAT	
	TRANSPORTATION AND AND AND AND AND AND AND AND AND AN	
	TOIPARTUALAINOTA/THI WAININY	
	IA TANON UNA IA NATE / TAN IA NATA NATA NATA NATA NATA NATA NA	
	UIRADATAKETTER	
	U1I=DATAIK1=1)+TZI	
	UZRETSHETAR	
	UZITJICTAL	
	UJREDATA(K1) TER	
	U31=DATA(K1+1)=T21	in the second
	IF(ISIGN)490,500,500	
490	U4R=T31=T41	
	U41=T4R=T3R	
	GO TO 510	
500	UAR#T41cT31	
	UAT=T3R=T4R	
510	DATA(K1)=U1R+U2R	
	DATA (K1+1)=U11+U21	
	DATA(K2)=U3R+U4R	
	DATA(K2+1)=U31+U41	
	DATA (K3) BUS REUS R	
	DATA(#3+1)#111+1121	
	DATA (VA) BIISPELIAD	
826	DATA FRAAT THIS FHIAT	
260	UNICENSTED	
	KUIP-KOTCP MANUAATEMINATINATI	
230	CONTINUE	
	MAMALMAX	
	IFICMEMMAX) DAU / DAU / DAU / D70	
540	IF(ISIGN/550/500/560	
550	TEMPRENR	물건 방법은 감독을 들었다. 여러 일을 위한 것이라고 있다.
	WR#(WR*WI)*RTHLF	
	WINCWI-TEMPRIARTHLF	
	GQ TO 410	
560	TEMPREWR	
	WR=(WR=WI)*RTHLF	
	WIN(TEMPR+WI)+RTHLF	
	GO TO 410	
570	CONTINUE	
	TPAR=3-IPAR	
	MMAXEMMAX+MMAX	
	00 10 360	
0	AA IA AAA	
	MAIN LOOP FOR FACTORS NOT BOUNT	TO THO. APPLY THE THINDLE FACTOR
	USEVO/101ANS2-DIACADT/_1141	10 12d 111// 1501 0100011. WUGU
	REALISION & COUDIES SDANSFADM AF I	CNATH TRAFTIES, MARTING HOM AP
G	HENFUNN A FUUNTER TRANSFURM OF .	TOWNER THATTERS WANTUR USE OF
⁰	CONJUGATE STAMETRIES	
000	IF(NTWO+NP2/002,700,700	

605 IFRIANTWO
IF INON2
NR1HF=NR1/2
610 IFP2=IFACT(IF)=IFP1
JIMINENPIAL
IF(J1MIN=1FP1)615,615,640
615 DO 635 JARJIMIN, IFRI, NR1
THETA==TWOPIFFLOAT(J1=1)/FLOAT(IFPR)
IF(ISIGN)625+620+620
620 THETAR THETA
625 WSTPRECOS(THETA)
WSTPI=SIN(THETA)
WRENSTRR
WIEWSTPI
IDMINE.IS .TFR1
12MAXE.IT+IFP2-IFP1
DO 635 J28J2MIN-J2MAX, FFR1
11WAX#J2+11RNA=2
00 630 110/2/11MAY.2
DO 630
THURREDATALISS
DATA (IS INDATAC IS) - UD-DATA (IS+5 YOUT
900 DATA 1996/ TERCATA 19919191919191
030 WINIERRAWSTEITHINNSIER
040 THETAR TROPICFUUATESARUISE
[F[]3]UN/070/072/042
045 THETASTTHETA
650 WSTPHECOS(THETA)
WSTPI#SIN(THETA)
J2RNG#IFP1#(1+IFACT(IF)/2)
DO 695 [181/ IIRNG, 2
DO 695 [JELLANTOT, NP2
J2MAXHI3+J2RNG+IFP1
DO 690 JZEIJ/JZMAX/IPP1
J1MAX#J2+[FP1=NP1
DO 680 JIBJZ/JIMAX/NR1
JJMAX#J1+NR2=IFP2
DO 680 JJEJ1 JJMAX , IFP2
N#1N#12495+12
JMAX=JMIN+IFP2=IFP1
1=t+(J3et3)/NP1HF
1F(J2=131655+655+665
655 SUMR#0.
SUMI#0.
DO 660 JEJMIN, JMAX, [FP1
SUMR#SUMR+DATA(J)
660 SUMISSUMI+DATA(J+1)
WORK (1) #SUMR
WORK (1+1)=SUM1
GO TO 680
665 ICONJ#1#(1FP2=2*J2#13*J3)/NP1HP
XAMLEL
SUMRADATA(J)
SUME BDATA(J+1)
OLDSR.D.
0105140.

Section Section 197	
10 Mar - 1	「王書詞即是是是是書書」
Sec.	
and the second second	
the second second second	THE PART PROPERTY AND A PARTY
	·····································
13 HE	
MUSTER CHAIN	17886348398543487554755475
	Linkschutzenfinsenfinselen under under
	UARP/ STORES, TEMPI
	その形式をきたない」と言葉語がある本語がある。
	- 中国経動動産経動を通貨換支
	HORV? TARUGA & Secondar tenic ?
680	CONTINUE

4.0.0	
655	· · · · · · · · · · · · · · · · · · ·
and the surface of	w te water to the second se
	100 IO 10 10 10
686	
690	"平城台段府府城段市城 段
	了混沌点天用李泰中和PPI品用NPI名
	nn AGB 19a13/12MAX.NR1
	DATASSEAWURNSS
	DATA(1241)==WORK(5+1)
405	0.00
042	2 W 1 W 1
	TEDISTED9
	1F(1FP1=NP2)610,700,700
c	1F(1FR1=NP2)010,700,700
C	IF(IFRIENP2)610,700,700
CC	COMPLETE & REAL TRANSFORM IN THE IST DIMENSION, N'EVEN, BY CON-
000	IF([FRINP2]010,700,700 COMPLETE & REAL TRANSFORM IN THE IST DIMENSION, N'EVEN, BY CON- JUGATE SYMMETRIES,
C	IF(IFRIENP2)610,700,700 COMPLETE A REAL TRANSFORM IN THE 157 DIMENSION, NIEVEN, BY CON- JUGATE SYMMETRIES,
C C C	IF(IFRIENP2)610,700,700 COMPLETE & REAL TRANSFORM IN THE IST DIMENSION, NIEVEN, BY CON- JUGATE SYMMETRIES,
C C C 700	IF([FRIENP2]010,700,700 COMPLETE & REAL TRANSFORM IN THE IST DIMENSION, N'EVEN, BY CON- JUGATE SYMMETRIES, GO TO (900,800,900,701), ICASE
C C C 700 701	IF(IFRIENP2)610,700,700 COMPLETE A REAL TRANSFORM IN THE IST DIMENSION, NEVEN, BY CON- JUGATE SYMMETRIES, GO TO (900,800,900,701), ICASE NHALFEN
C C C 700 701	IF(IFRINNP2)010,700,700 COMPLETE A REAL TRANSFORM IN THE IST DIMENSION, N'EVEN, BY CON- JUGATE SYMMETRIES, GO TO (900,800,900,701), ICASE NHALFEN NENEN
C C C 700 701	IF(IFRIENP2)010,700,700 COMPLETE & REAL TRANSFORM IN THE IST DIMENSION, NIEVEN, BY CON- JUGATE SYMMETRIES, GO TO (900,800,900,701), IGASE NHALFEN NEN+N
C C C 700 701	IF(IFRIENP2)610,700,700 COMPLETE A REAL TRANSFORM IN THE IST DIMENSION, NEVEN, BY CON- JUGATE SYMMETRIES, GO TO (900,800,900,701), IGASE NHALFEN NENAN THETA=TWORI/FLOAT(N)
C C C 700 701	IF(IFPI=NP2)010,700,700 ICOMPLETE & REAL TRANSFORM IN THE IST DIMENSION, N EVEN, BY CON- JUGATE SYMMETRIES, GO TO (900,800,900,701), IGASE NMALF=N N=N+N THETA==TWORI/FLOAT(N) IF(ISIGN)703,702,702
C C C 700 701	IF(IFRIENP2)010,700,700 ICOMPLETE & REAL TRANSFORM IN THE 157 DIMENSION, N EVEN, BY CON- JUGATE SYMMETRIES, GO TO (900,800,900,701),IGASE NHALFEN NUN+N THETA==TWORI/FLOAT(N) IF(ISIGN)703,702,702
C C 700 701 702	IF(IFP1=NP2)010,700,700 IF(IFP1=NP2)010,700,700 COMPLETE A REAL TRANSFORM IN THE 157 DIMENSION, N EVEN, BY CON- JUGATE SYMMETRIES, GO TO (900,800,900,701), IGASE NHALF=N NUN+N THETA==TWORI/FLOAT(N) IF(ISIGN)703,702,702 THETA==THETA
C C 700 701 702 703	IF(IFPI=NP2)010,700,700 IP(IFPI=NP2)010,700,700 COMPLETE A REAL TRANSFORM IN TWE STIDIMENSION, NEVEN, BY CON- JUGATE SYMMETRIES, GO TO (900,800,900,701), IGASE NHALF=N NNN+N THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 IF(ISIGN)703,702,702 THETA==THETA WSTPR#COS(THETA)
C C C 700 701 702 703	IF(IFPinnp2)010,700,700 COMPLETE & REAL TRANSFORM IN TWE ST DIMENSION, NEVEN, BY CON- JUGATE SYMMETRIES, GO TO (900,800,900,701), IGASE NMALF=N NNN+N THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA=THETA WSTPR#COS(THETA)
C C C 700 701 702 703	IF(IFPI=NP2)010,700,700 IP(IFPI=NP2)010,700,700 IDUGATE SYMMETRIES, GO TO (900,800,900,701),ICASE NHALF=N NUN+N THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA==THETA WSTPR#COS(THETA) NGTPI#SIN(THETA)
C C C 700 701 702 703	IF(IFP1=NP2)010,700,700 COMPLETE & REAL TRANSFORM IN TWE 15T DIMENSION, NEVEN, BY CON- JUGATE SYMMETRIES, GO TO (900,800,900,701), IGASE NMALF=N NNN+N THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 IF(ISIGN)703,702,702 THETA==TMETA WSTPR#COS(THETA) WR=WSTPR
C C 700 701 702 703	IF(IFPinNP2)010,700,700 COMPLETE & REAL TRANSFORM IN TWE ST DIMENSION, NEVEN, BY CON- UUGATE SYMMETRIES, GO TO (900,800,900,701), IGASE NMALF=N NNN+N THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA=THETA WSTPR#COS(THETA) WR=NSTPR WSTPI
C C 700 701 702 703	IF(IFPINNP2)010,700,700 IP(IFPINNP2)010,700,700 IDUGATE SYMMETRIES, GO TO (900,800,900,701),IGASE NHALFON NUNAN THETADOTWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETADOTHETA WSTPRWCOS(THETA) WSTPIMETAN WROWSTPR WIGHTSTPI IMINAS
C C C 700 701 702 703	IF(IFPINNP2)010,700,700 ICOMPLETE & REAL TRANSFORM IN TWE IST DIMENSION, N EVEN, BY CON- JUGATE SYMMETRIES, GO TO (900,800,900,701),IGASE NMALF=N NEN+N THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA=TNETA WSTPR#COS(THETA) WSTPIESIN(THETA) WR=WSTPR WI=WSTPI IMIN=3
C C C 700 701 702 703	IF(IFPinNp2)010,700,700 COMPLETE & REAL TRANSFORM IN TWE ST DIMENSION, NEVEN, BY CON- JUGATE SYMMETRIES, GO TO (900,800,900,701), IGASE NNALF=N NNN+N THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA==TWETA WSTPR#COS(THETA) NR=NSTPR WINNS UNINS IMINS JMINS2*NHALF=1
C C C 700 701 702 703	IF(IFPI=NP2)010,700,700 IF(IFPI=NP2)010,700,700 COMPLETE & REAL TRANSFORM IN THE IST DIMENSION, N EVEN, BY CON- JUGATE SYMMETRIES, GO TO (900,800,900,701),IGASE NMALF=N NETA==TWORI/FLOAT(N) IF(ISIGN)703,702,702 THETA==THETA WSTPR#COS(THETA) WSTPI#SIN(THETA) WR=WSTPR W[swSTPR W[swSTPR] IM[N=3. JMIN=2=NHALF=1 GO TO 725
C C C 700 701 702 702	IF(IFPinnp2)010,700,700 IF(IFPinnp2)010,700,700 UOMPLETE & REAL TRANSFORM IN TWE 157 DIMENSION, N EVEN, BY CON- JUGATE SYMMETRIES, GO TO (900,800,900,701),IGASE NMALF=N NEN+N THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA==TMETA WSTPR#COS(THETA) WSTPI#SIN(THETA) WR=WSTPN WI=SIN(THETA) WR=WSTPN JMINE2=NWALF=1 GO TO 725 Im IN=
C C C 700 701 702 703 710	IF(IFPinnp2)010,700,700 COMPLETE & REAL TRANSFORM IN TWE ST DIMENSION, NEVEN, BY CON- JUGATE SYMMETRIES, GO TO (900,800,900,701), IGASE NWALF=N NUMAFN THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA==TWETA WSTPR#COS(THETA) WSTPR#COS(THETA) NR=WSTPI IMIN=3 JMIN=2*NHALF=1 GO TO 725 JEJMIN
C C C 700 701 702 702 703	IF(IFPI=NP2)010,700,700 IP(IFPI=NP2)010,700,700 COMPLETE & REAL TRANSFORM IN THE IST DIMENSION, N EVEN, BY CON- JUGATE SYMMETRIES, GO TO (900,800,900,701),IGASE NMALF=N NETA==TWORI/FLOAT(N) IF(ISIGN)703,702,702 THETA==THETA WSTPR#COS(THETA) WSTPI#SIN(THETA) WR=WSTPR WI=NSTPR WI=NSTPR JMIN=2=NHALF=1 GO TO 725 J#JMIN DO 720 J#IMIN,NTOT,NR2
C C C 700 701 702 703 702 703	IF(IFPinNP2)010,700,700 IF(IFPinNP2)010,700,700 UOMPLETE & REAL TRANSFORM IN TWE 157 DIMENSION, N EVEN, BY CON- JUGATE SYMMETRIES, 00 TO (900,800,900,701),IGASE NMALF=N NEN+N THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA==TMETA WSTPR#COS(THETA) WSTPI#SIN(THETA) WR=WSTPR WI=SIN(THETA) WR=WSTPR J#JNNS J#JNNN DO 720 J#JMIN_NTOT,NP2 SUMR=(DATA(I)+DATA(J))/2.
C C 700 701 702 703 710	IF(IFPI=NP2)010,700,700 IP(IFPI=NP2)010,700,700 ID(ATE SYMMETRIES, GO TO (000,800,000,701),ICASE NHALF=N NWN+N THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA==THETA WSTPR#COS(THETA) WSTPR#COS(THETA) WR=WSTPR WR=WSTPR IM(N=3 JHIN=2=NHALF=1 GO TO 725 JHJMIN DO 720 IFIMIN,NTOT,NR2 SUMRE(DATA(I))/2,
C C C 700 701 702 703 702	IF(IFPInNP2)010,700,700 IF(IFPInNP2)010,700,700 COMPLETE & REAL TRANSFORM IN TWE IST DIMENSION, N EVEN, BY CON- JUGATE SYMMETRIES, GO TO (900,800,900,701),IGASE NMALF=N NETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA==THETA WSTPR#COS(THETA) WSTPR#COS(THETA) WSTPI#SIN(THETA) WR=WSTPR WI=WSTPR WI=NMALF=1 GO TO 725 J#JMIN DO 720 IFIMIN,NTOT,NP2 SUMR=(DATA(I)+DATA(J))/2, SUMI=(DATA(I+1)+DATA(J))/2,
C C 700 701 702 703 710	IF(IFPinNP2)010,700,700 IF(IFPinNP2)010,700,700 COMPLETE & REAL TRANSFORM IN TWE 157 DIMENSION, N EVEN, BY CON- JUGATE SYMMETRIES, GO TO (900,800,900,701),IGASE NMALF=N NEN+N THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA=THETA WSTPIMETA WSTPIMETA WSTPIMETA WSTPIMETA WSTPIMETA WSTPIMETA WSTPIMETA WSTPIMETA JMINE2=NHALF=1 GO TO 725 JMJMIN DO 720 IFIMIN,NTOT,NP2 SUMR=(DATA(I)>DATA(J))/2, DIFR=(DATA(I)>DATA(J))/2, DIFR=(DATA(I)>DATA(J))/2,
C C 700 701 702 703 710	IF(IFPInNP2)010,700,700 IF(IFPInNP2)010,700,700 COMPLETE A REAL TRANSFORM IN THE IST DIMENSION, N EVEN, BY CON- JUGATE SYMMETRIES, 00 TO (900,800,900,701),IGASE NHALFEN NEN+N THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA=THETA WSTPR#COS(THETA) WSTPI#SIN(THETA) WSTPI#SIN(THETA) WREWSTPR VIEWSTPI IMIN=2=NHALF=1 00 TO 725 JEJMIN DO 720 IFIMIN,NTOT,NP2 SUMR*(DATA(I)=DATA(J)/2, DIFR*(DATA(I==DATA(J)/2, DIFR*(DATA(I==DATA(J)/2, DIFR*(DATA(I==DATA(J)/2, DIFR*(DATA(I==DATA(J)/2, DIFR*(DATA(I==DATA(J)/2, DIFR*(DATA(I==DATA(J)/2, DIFR*(DATA(I==DATA(J)/2, DIFR=(DATA(I==DATA(J)/2, DIFR=(DATA(I==DATA(J)/2, DIFR=(DATA(I==DATA(J)/2, DIFR=(DATA(I==DATA(J)/2, DIFR=(DATA(I==DATA(J)/2, DIFR=(DATA(I==DATA(J)/2, DIFR=(DATA(I==DATA(J)/2, DIFR=(DATA(I==DATA(J)/2, DIFR=(DATA(I==DATA(J)/2, DIFN=(DATA(I=DATA(J)/2, DIFN=(DATA(I=DATA(J)/2, DIFN=(DATA(I=DATA(J)/2, DIFN=(DATA(I=DATA(J)/2, DIFN=(DATA(I=DATA(J)/2, DIFN=(DATA(I=DATA(J)/2, DIFN=(DATA(I=DATA(J)/2, DIFN=(DATA(I=DATA(J)/2, DIFN=(DATA(J)
C C C 700 701 702 703	IF(IFPInNP2)010,700,700 IF(IFPInNP2)010,700,700 COMPLETE A REAL TRANSFORM IN THE 15T DIMENSION, N EVEN, BY CON- JUGATE SYMMETRIES, CO TO 1900,800,900,701),IGASE NHALF=N NHN+N THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA==THETA WSTPRWCOS(THETA) WSTPRWCOS(THETA) WSTPI=SIN(THETA) WSTPI=SIN(THETA) WSTPI=SIN(THETA) WSTPI=SIN(THETA) UNINE2=NHALF=1 CO TO 725 J#JMIN DO 720 IFIMIN,NTOT,NP2 SUMME(DATA(I)=DATA(J))/2, SUMI=(DATA(I)=DATA(J))/2, DIFR=(DATA(I)=DATA(J))/2, DIFI=(DATA(I)=DATA
C C 700 701 702 703 710	IF(IFPInNP2)610,700,700 IF(IFPInNP2)610,700,700 COMPLETE & REAL TRANSFORM IN THE IST DIMENSION, N EVEN, BY CON- JUGATE SYMMETRIES, 00 TO 1900,800,900,701,,IGASE NHALF#N NHN*N NHN*N NHN*N NHN*N NHN*N NHN*N NHTHETA==TWOPI/FLOAT(N) IF(ISION)703,702,702 THETA==TWETA WSTPR#COSITHETA) WSTP
C C 700 701 702 703 710	IF(IFPInNP2)610,700,700 IF(IFPInNP2)610,700,700 COMPLETE & REAL TRANSFORM IN THE 15T DIMENSION, N EVEN, BY CON- JUGATE SYMMETRIES, CO TO 1900,800,900,7013,3CASE NHALF=N NHN=N THETA==TWOPI/FLOAT(N) IF(ISIGN)7D3,702,702 THETA==THETA WSTPR#COS(THETA) WSTPI#SIN(THETA) WSTPI#SIN(THETA) WR=WSTPR W1=WSTPR W1=WSTPI IMIN#S JMIN#2=NHALF=1 CO TO 720 IFIMIN,NTOT,NP2 SUMR=(DATA(I)=DATA(J))/2, SUMR=(DATA(I)=DATA(J))/2, DIFR=(DATA(I)=DATA(J))/2, DIFR=(DATA(I)=DATA(J))/2, DIFR=NA+SUMI=#IPIFR
C C C 700 701 702 703	IF(IF#1:1/* IF(IF#1:N#2:610,700,700 COMPLETE & REAL TRANSFORM IN THE 157 DIMENSION, N EVEN, BY CON- JUGATE SYMMETRIES, 00 TO (900,800,900,701),IGASE NHALF=N NWN+N THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 IF(ISIGN)70,702
C C 700 701 702 703 710	IF(IF#1:1/* IF(IF#1:NP2)610,700,700 COMPLETE A REAL TRANSFORM IN THE 157 DIMENSION, N EVEN, BY CON- JUGATE SYMMETRIES, CO TO (900,800,900,701), IGASE NMALF=N NWN+N THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA==THETA WSTPR#COS(THETA) WSTPI=SIN(THETA) WSTPI=SIN(THETA) WSTPR#COS(THETA) WSTPI=SIN(THETA) WSTPR#COS(THETA) WSTPI=SIN(THETA)
C C 700 701 702 703 710	<pre>IF(IF#E=NP2}610,700,700 COMPLETE A REAL TRANSFORM IN THE 1ST DIMENSION, N EVEN, BY CON- JUGATE SYMMETRIES, CO TO (900,800,900,701),IGASE NHALF=N NWN+N THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA==THOPI/FLOAT(N) IF(ISIGN)703,702,702 IF(ISIGN)70,725 IF(</pre>
C C C 700 701 702 703	<pre>IF(IF#1://***********************************</pre>
C C 700 701 702 703 710	<pre>IF(IF#1:1:#2;610,700,700) IF(IF#1:1:#2;0)MENSION, N'EVEN, BY CON- JUGATE SYMMETRIES, GO TO (900,800,900,701),IGASE NHALF=N NWN+N THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA==TWOPI/FLOAT(N) IF(ISIGN)703,702,702 THETA==TWETA WSTPR#COS(THETA) WSTPR#COS(THETA) WSTPR#COS(THETA) WSTPR#COS(THETA) WSTPR#COS(THETA) UNIN#2*NHALF=1 GO TO 725 J#JMIN DO 720 IFIMIN,NTOT,NR2 SUMR=CDATA(1,20DATA(J))/2, DIFR=CDATA(1,20DATA(J))/2, DIFR=CDATA(1,20DATA(J))/2, DIFI=CDATA(1,20DATA(J))/2, DIFI=SUMI=WENDIFER TEMP3#WI=SUMI=WI=DIFR DATA(1)=SUMI=WENDIFER DATA(1)=SUFF=1 DA</pre>

720	Jag Jan Market and Andreas a
	1M1N#1M1N+2
	JMIN#JMIN#2
	TEMPRAWR
	WRAWRWSTPROWINWSTPI
	UTATEMPROUSTPTOUTOWSTPR
-	TO THY NH MINI 210, 730, 740
762	
130	IF (ISIUN//SI//TUI/TU
731	DO 100 SEIMINENTOTENEE
735	DATASTELIANDATASTES
740	NBSENBS
	NTOTANTOT+NTOT
	JUNTOT+1
	IMAX=NTOT/2+1
745	ININ#IMAX-2*NHALF
	2 m 2 M 2 N
	GO TO 759
758	DATA(J)BDATA(J)
100	
122	
	IF(I-1MAX)/DU,/DU,/DU
760	DATA QJEDATA SIMINI-DATA SIMINALI
	DATA(J+2)s0;
	IF(1=J)770,780,780
765	DATA(J)=DATA(I)
	CATACJ+L2ATAC +1
770	2 m # # # #
	tania 2
	*#/***********************************
	TATA AND ATA (PMTNIA DATA (IMTNA)
112	
	SMAAF ITSN
780	DATALIJODATALIJODATALEJ
	DATA(2)EU,
	100 TO 900
Ç	
C	COMPLETE A REAL TRANSFORM FOR THE 2ND OR 3RD DIMENSION BY
C	CONJUGATE SYMMETRIES.
0	· · · · · · · · · · · · · · · · · · ·
800	TF(11RNG=NR1)805,900,900
805	DO 860 1381,NTOT.NP2
***	12MAYET JANBONNO1
	DO 860 1913.12MAY.NP1
	MUNET2-JERNO
	JAARKTIJENMASSININ
	IA(18410)98019801910
410	JNAXEJNAXANNX
950	11(101Weg1000'000'000
	JHJMAX+NPO
	DO BAO ESTMENA DMAXAR
	DATA(J)#BATA(J)
	DATA(}+1) a+DATA(J+1)
844	
	THE BAR SETHIN, SHAY JUDD
a she	
11.	DATASTASTATUATASUASI
	· · · · · · · · · · · · · · · · · · ·

C C	END OF LOOP	ON EACH DIMENSION
900	NPOSNP1	
910	NPREVEN	
920	END	

UNCLASSIFIED

Security Classification

(Security classification of title, be	ody of abstract and indexing annotation n	ust be entered when the ove	rall report is classified)
ORIGINATING ACTIVITY (Corporate a	uthor)	2a, REPORT SEC	URITY CLASSIFICATION
Lincoln Laboratory, M.I.T.		Unclas	ssified
		2b. GROUP None	
. REPORT TITLE			
Three Fortran Programs that	t Perform the Cooley-Tukey Four	ier Transform	
. DESCRIPTIVE NOTES (Type of report	and inclusive dates)		
lechnical Note			
Brenner, Norman M.	(181)		
, REPORT DATE	7a.	TOTAL NO. OF PAGES	7b. NO. OF REFS
28 July 1967		34	16
a. CONTRACT OR GRANT NO.	9a.	ORIGINATOR'S REPORT	NUMBER(S)
AF 19(628)5167 b. project no.		Technical Note 196	57-2
c. 649L	96.	OTHER REPORT NO(5) (A assigned this report)	Any other numbers that may be
d.		ESD-TR-67-462	
This document has been appro-	s oved for public release and sale; 12.	its distribution is unlin	nited.
This document has been appro-	s oved for public release and sale; 12.	its distribution is unlin sponsoring military of Air Force Systems	nited. ACTIVITY Command, USAF
This document has been appr supplementary notes None	s oved for public release and sale; 12.	its distribution is unlin Sponsoring military of Air Force Systems	nited. ACTIVITY Command, USAF
This document has been appr . SUPPLEMENTARY NOTES None	s oved for public release and sale; 12.	its distribution is unlin sponsoring military Air Force Systems	nited. ACTIVITY Command, USAF
This document has been approved This document has been approved None Three programs and form the discrete Four The data may be either The transform values and original data. The run summation, even when example, on a CDC 3300 size 80 × 80 can be transforage array of size 1	s oved for public release and sale; 12. 12. 12. 12. 12. 12. 12. 12	its distribution is unlin SPONSORING MILITARY Air Force Systems Air Force Systems in running time for rea ned in the array used to at of any program perfor- culated and stored in a bix microseconds, a con- es the main array, only	an, which per- point data. l over complex. o carry the orming a direct table. For mplex array of a working
This document has been approved SUPPLEMENTARY NOTES None ABSTRACT ABSTRACT The data may be either The data may be either The transform values a original data. The run summation, even when example, on a CDC 3300 size 80 × 80 can be tran storage array of size 1	s oved for public release and sale; 12. 12. 12. re described and listed, all writted ier transform upon a multidimens real or complex, with a savings ure always complex and are return ning time is much shorter than the sine and cosine values are precation 00 with floating point add time of so nsformed in 19.2 seconds. Beside 60 need be supplied.	its distribution is unlin SPONSORING MILITARY Air Force Systems Air Force Systems in running time for rea ned in the array used to at of any program perfor- culated and stored in a ix microseconds, a con- es the main array, only	an, which per- point data. l over complex. o carry the orming a direct table. For mplex array of ' a working
This document has been approved This document has been approved SUPPLEMENTARY NOTES None ABSTRACT Three programs and form the discrete Four The data may be either The transform values and original data. The run summation, even when example, on a CDC 3300 size 80 × 80 can be transitor storage array of size 1 KEY WORDS	s oved for public release and sale; 12. re described and listed, all writted ier transform upon a multidimens real or complex, with a savings ure always complex and are return ning time is much shorter than the sine and cosine values are precation 00 with floating point add time of so nsformed in 19.2 seconds. Beside 60 need be supplied.	its distribution is unlin SPONSORING MILITARY A Air Force Systems din Torce Systems in running time for rea ned in the array used to at of any program perfor- culated and stored in a ix microseconds, a con- es the main array, only	an, which per- point data. l over complex. o carry the orming a direct table. For mplex array of r a working

UNCLASSIFIED Security Classification

Printed by United States Air Force L. G. Hanscom Field Bedford, Massachusetts

See .

.