10.2 Trigonometric, Cosine, and Sine Fourier Transforms

A. Purpose

This subroutine computes discrete trigonometric, cosine, and sine transforms in up to six dimensions using the FFT. In the one dimensional case, the values y and the Fourier coefficients α and β are related by:

$$y_j = \frac{1}{2}\alpha_0 + \sum_{k=1}^{(N/2)-1}\left[\alpha_k \cos\frac{2\pi jk}{N} + \beta_k \sin\frac{2\pi jk}{N}\right] + \frac{1}{2}\alpha_{N/2}(-1)^j, \quad j = 0, 1, ..., N-1 \quad \text{(1S)}$$

$$\alpha_k = \frac{2}{N}\sum_{j=0}^{N-1} y_j \cos\frac{2\pi jk}{N}, \quad k = 0, 1, ..., \frac{N}{2}$$
$$\beta_k = \frac{2}{N}\sum_{j=0}^{N-1} y_j \sin\frac{2\pi jk}{N}, \quad k = 1, 2, ..., \frac{N}{2}-1 \quad \text{(1A)}$$

$$y_j = \frac{1}{2}\alpha_0 + \sum_{k=1}^{N-1}\alpha_k \cos\frac{\pi jk}{N} + \frac{1}{2}\alpha_N(-1)^j, \quad j = 0, 1, ..., N \quad \text{(2S)}$$

$$\alpha_k = \frac{2}{N}\left[\frac{1}{2}y_0 + \sum_{j=1}^{N-1} y_j \cos\frac{\pi jk}{N} + \frac{1}{2}y_N(-1)^k\right], \quad k = 0, 1, ..., N \quad \text{(2A)}$$

$$y_j = \sum_{k=1}^{N-1}\beta_k \sin\frac{\pi jk}{N}, \quad j = 1, 2, ..., N-1 \quad \text{(3S)}$$

$$\beta_k = \frac{2}{N}\sum_{j=1}^{N-1} y_j \sin\frac{\pi jk}{N}, \quad k = 1, 2, ..., N-1 \quad \text{(3A)}$$

where $N = 2^{\mathrm{M}(1)}$, ($N_i = 2^{\mathrm{M}(i)}$, $i = 1, ..., \mathrm{ND}$ in the multi-dimensional case.) In the equation labels above, the numeral 1, 2, or 3 denotes the Trigonometric, Cosine, or Sine transform, respectively, while the letter S or A denotes Synthesis or Analysis.

B. Usage

B.1 Program Prototype, Single Precision, One-Dimensional Transform

INTEGER M(1), ND, MS

REAL A($\geq \mu$) [$\mu = 2^{\mathrm{M}(1)} + 1$ for a Cosine transform, and $= 2^{\mathrm{M}(1)}$ for a Trigonometric transform or Sine transform.]

REAL S($\geq \nu-1$) [$\nu = 2^{\mathrm{M}(1)-2}$ for a Trigonometric transform and $= 2^{\mathrm{M}(1)-1}$ for a Cosine or Sine transform.]

CHARACTER TCS, MODE

On the initial call set MS to 0 to indicate the array S() does not yet contain a sine table. Assign values to A(), TCS, MODE, M(), and ND = 1 for a one-dimensional transform.

CALL STCST(A, TCS, MODE, M, ND, MS, S)

On return A() will contain computed values. S() will contain the sine table used in computing the Fourier transform. MS may have been changed.

B.2 Argument Definitions

A() [inout] If MODE selects analysis, A() contains y on input and α and/or β on output. If MODE selects synthesis, A() contains α and/or β on input and y on output. Let $N = 2^{\mathrm{M}(1)}$.

When A() contains y's, the element y_j is stored in A($j+1$). The range of j is $[0, N-1]$ for the Trigonometric transform, $[0, N]$ for the Cosine transform, and $[1, N-1]$ for the Sine transform.

For the Trigonometric transform, the α's and β's are stored as A(1) $= \alpha_0$, A(2) $= \alpha_{N/2}$, and $A(2k+1) = \alpha_k$, A($2k+2$) $= \beta_k$, $k = 1, 2, ..., (N/2)-1$.

For the Cosine transform, the α's are stored as A($k+1$) $= \alpha_k$, $k = 0, 1, ..., N$.

For the Sine transform, the β's are stored as A($k+1$) $= \beta_k$, $k = 1, 2, ..., N-1$. Although the value contained in A(1) is irrelevant for the Sine transform, the subroutine does access this location, so A(1) must contain a valid floating-point number on entry and its value will generally be changed on return.

TCS [in] The character variable TCS selects the type of transform to be done.

'T' or 't' selects the Trigonometric transform (Formula 1A or 1S).

'C' or 'c' selects the Cosine transform (Formula 2A or 2S).

'S' or 's' selects the Sine transform (Formula 3A or 3S).

MODE [in] The character variable MODE selects Analysis or Synthesis.

'A' or 'a' selects Analysis (Formula 1A, 2A, or 3A).

'S' or 's' selects Synthesis (Formula 1S, 2S, or 3S).

M() [in] Defines $N = 2^{\mathrm{M}(1)}$. The number of real data points is $N+1$, N, and $N-1$, for the Cosine, Trigonometric, and Sine transforms respectively. Require $0 \le \mathrm{M}(1) \le 31$ (≤ 30 for the cosine or sine transform). If M(1) is 0, no action is taken.

ND [in] Number of dimensions, =1 for one-dimensional transforms.

MS [inout] Gives the state of the sine table in S(). Let MS_{in} and MS_{out} denote the values of MS on entry and return respectively. If the sine table has not previously been computed, set $\mathrm{MS}_{in} = 0$ or -1 before the call. Otherwise the value of MS_{out} from the previous call using the same S() array can be used as MS_{in} for the current call.

Certain error conditions described in Section E cause the subroutine to set $\mathrm{MS}_{out} = -2$ and return. Otherwise, with $\mathrm{M}(1) > 0$, the subroutine sets $\mathrm{MS}_{out} = \max(\mathrm{M}(1), \mathrm{MS}_{in})$ for the Trigonometric transforms and to $\mathrm{MS}_{out} = \max(\mathrm{M}(1)+1, \mathrm{MS}_{in})$ for the Cosine or Sine transforms.

If $\mathrm{MS}_{out} > \max(2, \mathrm{MS}_{in})$, the subroutine sets $\mathrm{NT} = 2^{\mathrm{MS}_{out}-2}$ and fills S() with $\mathrm{NT}-1$ sine values.

If $\mathrm{MS}_{in} = -1$, the subroutine returns after the above actions, not transforming the data in A(). This is intended to allow the use of the sine table for data alteration before a subsequent Fourier transform, as discussed in Section G of Chapter 16.0.

S() [inout] When the sine table has been computed, $\mathrm{S}(j) = \sin \pi j/(2 \times \mathrm{NT})$, $j = 1, 2, ..., \mathrm{NT}-1$, see MS above.

B.3 Program Prototype, Multi-dimensional Transforms

INTEGER M($\ge$ ND)**, ND, MS**

REAL A(μ_1, μ_2, ..., $\ge \mu_{ND}$) [$\mu_k = 2^{\mathrm{M}(k)} + 1$ for a Cosine transform and $= 2^{\mathrm{M}(k)}$ for a Trigonometric or Sine transform.]

REAL S($\ge \max(\nu_1, \nu_2, ..., \nu_{ND}) - 1$) [$\nu_k = 2^{\mathrm{M}(k)-2}$ for a Trigonometric transform and $= 2^{\mathrm{M}(k)-1}$ for a Cosine or Sine transform.]

CHARACTER TCS*($\ge$ ND), MODE*($\ge$ ND)

On the initial call set MS to 0 to indicate the array S() does not yet contain a sine table. Assign values to A(), TCS, MODE, M(), and ND.

CALL STCST(A, TCS, MODE, M, ND, MS, S)

On return A() contains the transformed data and S() contains the sine table used in computing the Fourier transform. The value of MS may have been changed.

B.4 Argument Definitions

A() [inout] Array used for input and output data, see Functional Description below for specification of the storage of items in A().

TCS [in] The character TCS(k:k) selects the type of transform to be done in the k^{th} dimension.

'T' or 't' selects the Trigonometric transform (Formula 1A or 1S).

'C' or 'c' selects the Cosine transform (Formula 2A or 2S).

'S' or 's' selects the Sine transform (Formula 3A or 3S).

MODE [in] The character MODE(k:k) selects Analysis or Synthesis in the k^{th} dimension.

'A' or 'a' selects Analysis (Formula 1A, 2A, or 3A).

'S' or 's' selects Synthesis (Formula 1S, 2S, or 3S).

M() [in] In the k^{th} dimension, $k = 1$, ..., ND, define $N_k = 2^{\mathrm{M}(k)}$. The index range for the values y_j in the k^{th} dimension is $[0, N_k - 1]$ for the Trigonometric transform, $[0, N_k]$ for the Cosine transform, and $[1, N_k - 1]$ for the Sine transform. Require $0 \le \mathrm{M}(k) \le 31$ (≤ 30 for the cosine or sine transform). If $\mathrm{M}(k) = 0$, no action is taken with respect to dimension k.

ND [in] Number of dimensions, $1 \le \mathrm{ND} \le 6$.

MS [inout] As for MS in the one dimensional case above, except, if there is no error and $\mathrm{M}(k) > 0$ for some k, the subroutine sets $\mathrm{MS}_{out} = \max(\mu_1, \mu_2, ..., \mu_{ND}), \mathrm{MS}_{in})$, where $\mu_k = \mathrm{M}(k)$ for the Trigonometric transform, and $= \mathrm{M}(k) + 1$ for the Cosine or Sine transform. NT is defined in terms of MS_{out} as described in Section B.2 above.

S() [inout] When the sine table has been computed, $\mathrm{S}(j) = \sin \pi j/(2 \times \mathrm{NT})$, $j = 1, 2, ..., \mathrm{NT}-1$, see MS above.

B.5 Modifications for Double Precision

Change STCST to DTCST, and the REAL type statements to DOUBLE PRECISION.

C. Examples and Remarks

Given

$$f(t) = \frac{2 \sinh t}{\sinh \pi t}$$

obtain an estimate of $\varphi(\omega) = \int_0^T f(t) \cos \omega t \, dt$ and compare it with $\varphi(\omega) = \sin(1)/(\cosh \omega + \cos(1))$, the true solution. Let T and Ω denote the largest values of t and ω to be used in the computation. As was done below

Eq. (18) in Chapter 16.0, we introduce the approximations

$$\begin{aligned}\varphi(\omega) &= \int_0^T f(t)\cos\omega t\,dt \\ &\approx \frac{T}{N}\left[\frac{1}{2}f(t_0) + \sum_{j=1}^{N-1} f(t_j)\cos\omega t_j + \frac{1}{2}f(t_N)\cos\omega t_N\right]\end{aligned}$$

where $t_j = jT/N$. With $\omega_k = k\pi/T$, we get

$$\varphi(\omega_k) \approx \frac{T}{2}\frac{2}{N}\left[\frac{1}{2}f(t_0) + \sum_{j=1}^{N-1} f(t_j)\cos\frac{\pi jk}{N} + \frac{1}{2}f(t_N)(-1)^k\right]$$

which except for the factor $T/2$ has the form of Eq. (2A) above. Good results are obtained by balancing the error due to a finite T with the error due to aliasing (*i.e.* use of a finite Ω). Thus we want to select T and Ω so that

$$\int_T^\infty |f(t)|\,dt \approx \frac{2}{\pi-1}e^{-(\pi-1)T}, \text{ and} \tag{4}$$

$$\int_\Omega^\infty |\varphi(\omega)|\,d\omega \approx 2\sin(1)e^{-\Omega} \tag{5}$$

are of the same order of magnitude. Since

$$\int_T^\infty |f(t)|\,dt < 10^{-4}\int_0^T |f(t)|\,dt \quad \text{for } T \geq 10.$$

there is no point in choosing $T > 10$. With $T = 10$, $N = 64$ gives $\Omega = (N\pi/T) = 6.4\pi$. The right hand sides of Equations (4) and (5) are then $\approx 4.7\times10^{-10}$ and 3.1×10^{-9} respectively.

The program at the end of this chapter carries out the above calculations, but prints results only for every 10^{th} value of k to save space.

D. Functional Description

The one-dimensional transforms computed by this subroutine are given by equations (1S), (1A), ..., (3S), (3A). The multi-dimensional transform is accomplished by applying the appropriate one-dimensional transforms in each dimension. To define the relation between input and output contents of the A() array in the multi-dimensional case we introduce a 4-argument function, T(*,*,*,*), in which the first argument can take the values ′T′,′C′, or ′S′ to denote Trigonometric, Cosine, or Sine, and the second argument can take the values ′A′ or ′S′ to denote Analysis or Synthesis. The third and fourth arguments are integers. There is also an implied argument, M, which together with the first argument contributes to the definition of N and μ.

$$N = 2^M$$

$$\mathrm{T}('\mathrm{T}','\mathrm{S}',j,k) = \begin{cases} 1/2 & k=0 \\ \cos(\pi jk/N) & k=2,4,...,N-2 \\ \sin(\pi j(k-1)/N) & k=3,5,...,N-1 \\ (1/2)(-1)^j & k=1 \end{cases}$$

$$\mathrm{T}('\mathrm{T}','\mathrm{A}',j,k) = \frac{2}{N}\begin{cases} \cos(\pi jk/N) & j=0,2,...,N-2 \\ \sin(\pi(j-1)k/N) & j=3,5,...,N-1 \\ (-1)^k, & j=1 \end{cases}$$

$$\mathrm{T}('\mathrm{C}','\mathrm{S}',j,k) = \begin{cases} 1/2 & k=0 \\ \cos(\pi jk/N) & k=1,2,...,N-1 \\ (1/2)(-1)^j & k=N \end{cases}$$

$$\mathrm{T}('\mathrm{C}','\mathrm{A}',j,k) = \frac{2}{N}\mathrm{T}('\mathrm{C}','\mathrm{S}',j,k)$$

$$\mathrm{T}('\mathrm{S}','\mathrm{S}',j,k) = \sin\frac{\pi jk}{N}$$

$$\mathrm{T}('\mathrm{S}','\mathrm{A}',j,k) = \frac{2}{N}\mathrm{T}('\mathrm{S}','\mathrm{S}',j,k)$$

$$\mu_i = \begin{cases} 2^{\mathrm{M}(i)} & \text{for a Trigonometric or Sine transform} \\ 2^{\mathrm{M}(i)}+1 & \text{for a Cosine transform} \end{cases}$$

The array A has its contents replaced according to the formula

$$\begin{aligned} A(j_1,j_2,...,j_{ND}) = \sum_{k_1=1}^{\mu_1}\cdots\sum_{k_{ND}=1}^{\mu_{ND}} A(k_1,k_2,...,k_{ND}) \\ \times\,\mathrm{T}(\mathrm{TCS(1:1)},\ \mathrm{MODE(1:1)},\ j_1-1, k_1-1)\times\cdots \\ \times\,\mathrm{T}(\mathrm{TCS(ND:ND)},\ \mathrm{MODE(ND:ND)},\ j_{ND}-1, k_{ND}-1),\end{aligned}$$

where $j_i = 1,2,...,\mu_i$ and storage conventions with respect to each dimension are defined as for the one dimensional transforms.

The computational procedure for the trigonometric transform in the one-dimensional case is almost identical to the procedure used in SRFT1. (See the second paragraph below Eq. 10 in Chapter 16.0 to see why this is so.)

The procedure for the cosine transform uses the trigonometric transform and the identity

$$\cos\frac{\pi j(2k+1)}{N} = \frac{\sin(2\pi j(k+1)/N) - \sin(2\pi jk/N)}{2\sin(\pi j/N)}.$$

This is used to transform

$$y_j = \frac{1}{2}\eta_0 + \frac{1}{2}\eta_N(-1)^j + \sum_{k=1}^{N-1}\eta_k\cos\frac{\pi jk}{N}$$

to

$$\begin{aligned} y_j = \frac{1}{2}\eta_0 + \frac{1}{2}\eta_N(-1)^j + \sum_{k=1}^{(N/2)-1}\eta_{2k}\cos\frac{2\pi jk}{N} \\ + \frac{1}{2\sin(\pi j/N)}\sum_{k=1}^{(N/2)-1}(\eta_{2k-1}-\eta_{2k+1})\sin\frac{2\pi jk}{N}.\end{aligned}$$

Let Y_j be the result from the trigonometric transform with $\alpha_k = \eta_{2k}$, $k = 0, 1, ..., N/2$ and $\beta_k = \eta_{2k-1} - \eta_{2k+1}$, $k = 1, 2, ..., (N/2) - 1$. It follows that

$$\begin{aligned} y_j + y_{N-j} &= Y_j + Y_{N-j} \\ y_j - y_{N-j} &= \frac{Y_j - Y_{N-j}}{2\sin(\pi j/N)} \end{aligned}$$

and thus one can compute y_j from Y_j. The computational procedure is to compute the α's and β's from the η's (the α's require no computation), use the trigonometric transform to get the Y's, and the Y's to compute the y's. The inverse transform is exactly the same except for a factor of $2/N$ as is clear from Eqs. (2S) and (2A).

The sine transform is obtained in a very similar way to the cosine transform. Details can be found in [1].

References

1. Fred T. Krogh, **SCT—Multi-dimensional Sine, Cosine, and Sine-Cosine Transforms**. TU Doc. CP-2313, NPO 11652, Jet Propulsion Laboratory, Pasadena, CA (1970).

E. Error Procedures and Restrictions

Require $1 \le \text{ND} \le 6$, and $0 \le \text{M}(k) \le 31$ for the Trigonometric transform, $0 \le \text{M}(k) \le 30$ for the Cosine and Sine transforms. Require that TCS and MODE have only the allowed values. On violation of any of these conditions the subroutine issues an error message using the error processing procedures of Chapter 19.2 with severity level = 2 to cause execution to stop. A return will be made with MS = −2 instead of stopping if the statement "CALL ERMSET(−1)" is executed before calling this subroutine.

If the sine table does not appear to have valid data, an error message is printed, and the sine table and then the transform are computed.

F. Supporting Information

The source language is ANSI Fortran 77.

Entry	Required Files
DTCST	DFFT, DTCST, ERFIN, ERMSG, IERM1, IERV1
STCST	ERFIN, ERMSG, IERM1, IERV1, SFFT, STCST

Subroutine designed and written by: Fred T. Krogh, JPL, August 1969, revised January 1988.

DRSTCST

```
c      program DRSTCST
c>> 1996-06-05 DRSTCST Krogh Fixes for conversion to C.
c>> 1996-05-28 DRSTCST Krogh Moved formats up.
c>> 1994-10-19 DRSTCST Krogh  Changes to use M77CON
c>> 1994-08-09 DRSTCST WVS Remove '0' from format
c>> 1992-04-22 DRSTCST CAO commented program statement
c>> 1989-05-08 DRSTCST FTK, CLL
c>> 1989-05-04 DRSTCST FTK, CLL
c Driver to demonstrate STCST
c     ------------------------------------------------------------------
c--S replaces "?": DR?TCST, ?TCST
c     ------------------------------------------------------------------
      real             F(65), S(32), T, TTIME
      real             DELTAT, DELTAO, OMEGA, FTRUE, SIN1,COS1
      real             PI, ZERO, ONE
      integer    K, KSKIP, M, MS, N, ND, MA(1)
      parameter (M = 6)
c      parameter (N = 2 ** M + 1)
      parameter (N = 65)
      parameter (ND = 1)
      parameter (KSKIP = 10)
      parameter (PI = 3.14159265358979323846E0)
      parameter (ZERO = 0.E0)
      parameter (ONE = 1.E0)
      data TTIME / 10.E0 /
      data MA / M /
 1000 format (/'  K', 4X, 'OMEGA', 8X, 'COMPUTED', 9X, 'TRUE')
```

```
 1001 format (1X, I3, 1P,E13.5, 2E15.7)
c     ------------------------------------------------------------------------
      SIN1 = SIN (ONE)
      COS1 = COS (ONE)
      DELTAT = TTIME / REAL(N - 1)
      DELTAO = REAL(KSKIP) * (PI / TTIME)
      T = ZERO
c                 Compute (TTIME / 2 ) * F(T)
      F(1) = TTIME / PI
      do 10 K = 2, N
         T = T + DELTAT
         F(K) = TTIME * SINH(T) / SINH(PI * T)
   10 continue
c
      MS = 0
      call STCST (F, 'C', 'A', MA, ND, MS, S)
c
      OMEGA = ZERO
      write (*, 1000)
      do 20 K = 1, N, KSKIP
         FTRUE = SIN1 / (COSH(OMEGA) + COS1)
         write (*, 1001) K, OMEGA, F(K), FTRUE
         OMEGA = OMEGA + DELTAO
   20 continue
      stop
      end
```

ODSTCST

K	OMEGA	COMPUTED	TRUE
1	0.00000E+00	5.4630244E−01	5.4630250E−01
11	3.14159E+00	6.9358177E−02	6.9358155E−02
21	6.28319E+00	3.1364569E−03	3.1364565E−03
31	9.42478E+00	1.3579165E−04	1.3580074E−04
41	1.25664E+01	5.8656733E−06	5.8689702E−06
51	1.57080E+01	2.6077032E−07	2.5362203E−07
61	1.88496E+01	1.4901161E−08	1.0960011E−08