

Differentiation by Fourier Transform—DIFR

The program DIFR (Differentiation by Fourier Transform) is a subroutine subprogram that calculates the derivative of data \dot{x}_m ($m = 0, 1, 2, \dots, N - 1$) by using the Fourier transform and Fourier inverse transform of N discrete-valued data x_m ($m = 0, 1, 2, \dots, N - 1$) given at equal interval in the time domain.

Note that if there are large discontinuities in the time history to be differentiated, the error will be large. An example of the use of this program is given in the example of the subroutine INFR.

DIFR (Differentiation by Fourier Transform)

【Purpose】

To differentiate equally spaced time history data using the Fast Fourier Transform.

【Usage】

(1) How to connect

CALL DIFR (N, X, ND, DT)

| Argument | Type | Parameter in calling program | Return Parameter |
|----------|--------------------------|--|----------------------------------|
| N | I | Total number of 1-D array X | Unchanged |
| X | R 1-D array (ND) | Time history data to be differentiated | Differentiated time history data |
| ND | I | Dimension size of X in calling program (ND.GE. N) | Unchanged |
| DT | R | Time increment (unit: sec) | Unchanged |

(2) Necessary subroutines and function subprograms

FAST

(3) Remarks

N must be less than or equal to 8192. (But it can be changed easily.)

【Calculation Method】

When N discrete-valued data x_m ($m=0, 1, 2, \dots, N-1$) at equal interval Δt are given in the time domain, first Fourier transform them to obtain the complex Fourier coefficients C_k in the frequency domain. Then,

after performing the following operations, the derivative \dot{x}_m ($m = 0, 1, 2, \dots, N - 1$) can be obtained by performing the inverse Fourier transform.

$$\left. \begin{aligned} D_0 &= 0 \\ D_k &= ikC_k, \quad D_{N-k} = D_k^* \quad k=1, 2, \dots, N/2-1 \\ D_{N/2} &= 0 \end{aligned} \right\}$$

$$\dot{x}_m = \frac{2\pi}{N\Delta t} \sum_{k=0}^{N-1} D_k e^{i(2\pi km/N)} \quad m = 0, 1, 2, \dots, N-1$$

The fast Fourier transform program FAST is used for the Fourier transform and the inverse Fourier transform. To make it convenient to use the Fourier, add a trailing zero to make the number of data a power of 2, and then complex the data by the 'INITIALIZATION' block in the program. At the end of the Fourier transform, the complex Fourier coefficients C_k are all multiplied by N , but the correction is done after the inverse Fourier transform is completed.

【Program List】

| | | | |
|---|--|------|----|
| C | ***** | DIFR | 1 |
| C | SUBROUTINE FOR DIFFERENTIATION BY FOURIER TRANSFORM | DIFR | 2 |
| C | ***** | DIFR | 3 |
| C | | DIFR | 4 |
| C | CODED BY Y. OHSAKI | DIFR | 5 |
| C | | DIFR | 6 |
| C | PURPOSE | DIFR | 7 |
| C | TO DIFFERENTIATE AN EQUI-SPACED TIME HISTORY BY APPLICATION OF | DIFR | 8 |
| C | FAST FOURIER TRANSFORM | DIFR | 9 |
| C | | DIFR | 10 |
| C | USAGE | DIFR | 11 |
| C | CALL DIFR(N, X, ND, DT) | DIFR | 12 |
| C | | DIFR | 13 |
| C | DESCRIPTION OF ARGUMENTS | DIFR | 14 |
| C | N - TOTAL NUMBER OF DATA N.LE. 8192 | DIFR | 15 |
| C | X(ND) - ORIGINAL/DIFFERENTIATED DATA AT CALL/RETURN | DIFR | 16 |
| C | ND - DIMENSION OF X IN CALLING PROGRAM | DIFR | 17 |
| C | DT - TIME INCREMENT IN DATA | DIFR | 18 |
| C | | DIFR | 19 |
| C | SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED | DIFR | 20 |
| C | FAST | DIFR | 21 |
| C | | DIFR | 22 |
| C | SUBROUTINE DIFR(N, X, ND, DT) | DIFR | 23 |
| C | | DIFR | 24 |
| C | COMPLEX C(8192) | DIFR | 25 |
| C | DIMENSION X(ND) | DIFR | 26 |
| C | PARAMETER (P2=6.283185) | DIFR | 27 |
| C | | DIFR | 28 |
| C | INITIALIZATION | DIFR | 29 |
| C | | DIFR | 30 |
| C | DO 110 M=1, N | DIFR | 31 |
| C | C(M)=CPLX(X(M), 0.) | DIFR | 32 |

| | | | |
|-----|--------------------------------|------|----|
| 110 | CONTINUE | DIFR | 33 |
| | NT=2 | DIFR | 34 |
| 120 | IF (NT. GE. N) GO TO 130 | DIFR | 35 |
| | NT=NT*2 | DIFR | 36 |
| | GO TO 120 | DIFR | 37 |
| 130 | IF (NT. EQ. N) GO TO 150 | DIFR | 38 |
| | DO 140 M=N+1, NT | DIFR | 39 |
| | C(M)=(0., 0.) | DIFR | 40 |
| 140 | CONTINUE | DIFR | 41 |
| 150 | NFOLD=NT/2+1 | DIFR | 42 |
| C | | DIFR | 43 |
| C | FOURIER TRANSFORM | DIFR | 44 |
| C | | DIFR | 45 |
| | CALL FAST (NT, C, 8192, -1) | DIFR | 46 |
| C | | DIFR | 47 |
| C | DIFFERENTIATION | DIFR | 48 |
| C | | DIFR | 49 |
| | C(1)=(0., 0.) | DIFR | 50 |
| | DO 160 K=2, NFOLD-1 | DIFR | 51 |
| | C(K)=CMPLX(0., REAL(K-1))*C(K) | DIFR | 52 |
| | C(NT-K+2)=CONJG(C(K)) | DIFR | 53 |
| 160 | CONTINUE | DIFR | 54 |
| | C(NFOLD)=(0., 0.) | DIFR | 55 |
| C | | DIFR | 56 |
| C | FOURIER INVERSE TRANSFORM | DIFR | 57 |
| C | | DIFR | 58 |
| | CALL FAST (NT, C, 8192, +1) | DIFR | 59 |
| | P=P2/REAL(NT)**2/DT | DIFR | 60 |
| | DO 170 M=1, N | DIFR | 61 |
| | X(M)=REAL(C(M))*P | DIFR | 62 |
| 170 | CONTINUE | DIFR | 63 |
| | RETURN | DIFR | 64 |
| | END | DIFR | 65 |