

フーリエ変換による微分 DIFR

プログラムDIFR (Differentiation by Fourier Transform) は、時間領域において、等間隔で与えられた N 個の離散値データ x_m ($m=0, 1, 2, \dots, N-1$) を、フーリエ変換およびフーリエ逆変換を利用することによって、データの微分値 \dot{x}_m ($m=0, 1, 2, \dots, N-1$) を求めるサブルーチン副プログラムである。

微分する時刻歴に、あまり大きい不連続個所があると、誤差が大きくなることには、注意が必要である。このプログラムの使用例は、別掲のサブルーチン INFRの使用例に出ている。

DIFR (フーリエ変換による微分)

【目的】

高速フーリエ変換を利用して、等時間間隔の時刻歴データを微分する。

【使用法】

(1) 接続方法

CALL DIFR (N, X, ND, DT)

引数	型	プログラムを呼ぶときの内容	プログラムから戻ったときの内容
N	I	データの個数 N 8192	不変
X	R 1次元配列 (ND)	微分されるデータ	微分したデータ
ND	I	配列 X の整合寸法	不変
DT	R	データの時間間隔	不変

(2) 必要なサブルーチンおよび関数副プログラム

FAST

【計算法】

時間領域において、等間隔 Δt で N 個の離散値データ x_m ($m=0, 1, 2, \dots, N-1$) が与えられたとき、まずこれをフーリエ変換して周波数領域における複素フーリエ係数 C_k を求め、これに対して次の演算

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

を行ったのち、フーリエ逆変換

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

を行えば、微分値 \dot{x}_m ($m=0, 1, 2, \dots, N-1$) が得られる。

フーリエ変換およびフーリエ逆変換には、高速フーリエ変換のプログラム FAST を使っている。FAST を使うのに都合がよいう、後続のゼロを付け加えて、データ個数をいったん 2 の累乗数とし、またデータを複素数化する操作を、プログラムの中の INITIALIZATION のブロックで行っている。フーリエ変換が終わった時点で、複素フーリエ係数 C_k は、すべて N 倍されているが、これに対する修正は、フーリエ逆変換が終わった後で行っている。

【プログラム】

```

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 SUBROUTINE DIFR(N,X,ND,DT) DIFR 7
C DIFR 8
C COMPLEX C(8192) DIFR 9
C DIMENSION X(ND) DIFR 10
C PARAMETER (PI2=6.283185) DIFR 11
C DIFR 12
C INITIALIZATION DIFR 13
C DIFR 14
C DO 110 M=1,N DIFR 15
C C(M)=CMPLX(X(M),0.) DIFR 16
110 CONTINUE DIFR 17
C NT=2 DIFR 18
120 IF(NT.GE.N) GO TO 130 DIFR 19
C NT=NT*2 DIFR 20
C GO TO 120 DIFR 21
130 IF(NT.EQ.N) GO TO 150 DIFR 22
C DO 140 M=N+1,NT DIFR 23
C C(M)=(0.,0.) DIFR 24
140 CONTINUE DIFR 25
150 NFOLD=NT/2+1 DIFR 26
C DIFR 27
C FOURIER TRANSFORM DIFR 28
C DIFR 29
C CALL FAST(NT,C,8192,-1) DIFR 30
C DIFR 31
C DIFFERENTIATION DIFR 32
C DIFR 33
C C(1)=(0.,0.) DIFR 34
C DO 160 K=2,NFOLD-1 DIFR 35
C C(K)=CMPLX(0.,REAL(K-1))*C(K) DIFR 36
C C(NT-K+2)=CONJG(C(K)) DIFR 37
160 CONTINUE DIFR 38
C C(NFOLD)=(0.,0.) DIFR 39
C DIFR 40
C FOURIER INVERSE TRANSFORM DIFR 41
C DIFR 42
C CALL FAST(NT,C,8192,+1) DIFR 43
C P=PI2/REAL(NT)**2/DT DIFR 44
C DO 170 M=1,N DIFR 45
C X(M)=REAL(C(M))*P DIFR 46
170 CONTINUE DIFR 47
C RETURN DIFR 48
C END DIFR 49

```