プログラムのページ

担当 鈴 木 誠 道

7206 計算時間低減を考慮した任意データ数に対する FFT のアルゴリズム*

米澤 洋 (九州工業大学)
高須賀 馨 (旭化成工業(株))

0. 問 題

時系列データ $X_i(i=0,1,\cdots,N-1)$ に対する離散的有限フーリエ変換 (DFT) は次式で定義される.

$$C_k=\sum_{i=0}^{N-1} X_i W^{ik} \qquad (k=0,1,\cdots,N-1) \qquad (1)$$

ただし, $W=\exp(-2\pi j/N)$ で回転因子と呼ばれる. C_k は DFT の k 番目の係数であり, 一般に複素数である.

この C_k を求めようとすると, N^2 回の演算回数 (ここでいう演算回数とは, データ X_i と W^{ik} の積和の回数) を必要とし, 莫大な時間を要する. 最近, Cooley と Tukey[1] が開発した高速フーリエ変換 (FFT) を用いることによって, 演算回数を飛躍的に減小できるようになった. しかし, この FFT もプログラム作成上の簡便さから, 使用するデータ数が2のベキ乗でなければならぬというきびしい条件がある. この条件を緩和したアルゴリズム[2] も発表されているが, データ数に制約のない一般的な FFT のアルゴリズムが必要となる.

また, FFT ではフーリエ係数 C_k を $k=0,1,\cdots,N-1$ まで同時に計算する (すなわち, 折り曲げ周波数の2倍のところまで計算する) ため, 無駄な計算を強いられている. この無駄な計算を省略すれば, 計算所要時間は更に低減できることになる.

1. 任意のデータ数に対する FFT のアルゴリズム

いま, データ数 N が $N=r_1 r_2 \cdots r_m$ のとき,

$$\{k_{m-1}, k_{m-2}, \cdots, k_1, k_0\} \equiv k_{m-1}(r_1 r_2 \cdots r_{m-1})+k_{m-2}(r_1 r_2 \cdots r_{m-2}) + \cdots + k_1 r_1 + k_0.$$

$$\{i_{m-1}, i_{m-2}, \cdots, i_1, i_0\} \equiv i_{m-1}(r_2 r_3 \cdots r_m)+i_{m-2}(r_3 r_4 \cdots r_m) + \cdots + i_1 r_m + i_0.$$

ただし,

$$k_q=0,1,\cdots,r_{q+1}-1,$$
$$i_q=0,1,\cdots,r_{m-q}-1,$$
$$q=0,1,\cdots,m-1$$

とおけば, 式 (1) は

$$C_{\{k_{m-1},\cdots,k_0\}} = \sum_{i_0=0}^{r_{m-1}} \sum_{i_1=0}^{r_{m-1}-1} \cdots \sum_{i_{m-1}=0}^{r_1-1} X_{\{i_{m-1},\cdots,i_0\}} \times W^{\{i_{m-1},\cdots,i_0\}\{k_{m-1},\cdots,k_0\}}$$

$$= \sum_{i_0=0}^{r_{m-1}} \sum_{i_1=0}^{r_{m-1}-1} \cdots \left\{ \sum_{i_{m-1}=0}^{r_1-1} X_{\{i_{m-1},\cdots,i_0\}} \times W^{\{i_{m-1},0,\cdots,0\}\{k_{m-1},\cdots,k_0\}} \right\} \times \{ W^{\{0,i_{m-2},\cdots,i_0\}\{k_{m-1},\cdots,k_0\}} \}. \qquad (2)$$

と書き改めることができる. ところで,

$$W^{\{i_{m-1},0,\cdots,0\}\{k_{m-1},\cdots,k_0\}} = W^{r_1 r_2 \cdots r_m \cdot i_{m-1}\{k_{m-1}(r_2 r_3 \cdots r_{m-1})+k_{m-2}(r_2 r_3 \cdots r_{m-2}) + \cdots + k_1\} + i_{m-1}(r_2 r_3 \cdots r_m)k_0} = W^{i_{m-1}(r_2 r_3 \cdots r_m)k_0}$$

であるから式 (2) は次式のようになる.

$$C_{\{k_{m-1},\cdots,k_0\}} = \sum_{i_0=0}^{r_{m-1}} \cdots \left\{ \sum_{i_{m-1}=0}^{r_1-1} X_{\{i_{m-1},\cdots,i_0\}} \times W^{i_{m-1}(r_2 r_3 \cdots r_m)k_0} \right\} \times W^{\{0,i_{m-2},\cdots,i_0\}\{k_{m-1},\cdots,k_0\}}.$$

上式 { } 内は

$$\sum_{i_{m-1}=0}^{r_1-1} X_{\{i_{m-1},\cdots,i_0\}} W^{i_{m-1}(r_2 r_3 \cdots r_m)k_0} = C_{1\{k_0,i_{m-2},\cdots,i_0\}}$$

と定義できるから, 同様な操作を p 回 ($p=2,3,\cdots,$

---

* An FFT Algorithm for Arbitary Data-Numbers with Reduction in Computing-Time
(Yoo Yonezawa: Kyushu Inst. of Tech.)
(Kaoru Takasuka: Asahi Kasei Kogyo Co., Ltd.)

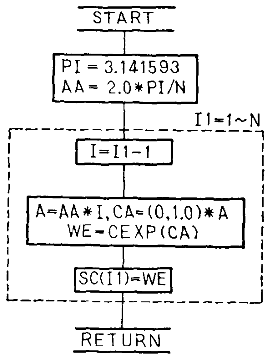

図 1 SUBROUTINE JUNBI

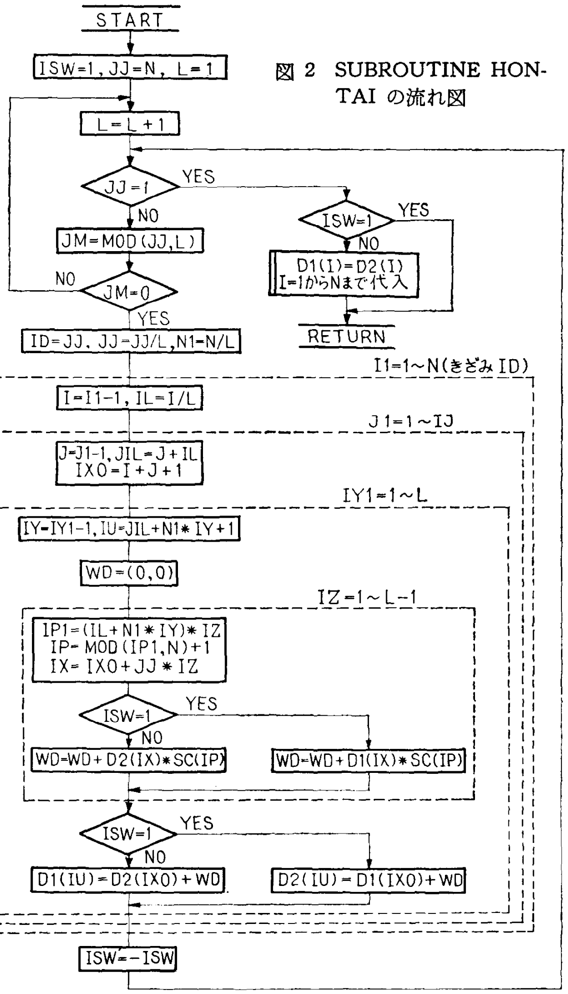

図 2 SUBROUTINE HONTAI の流れ図

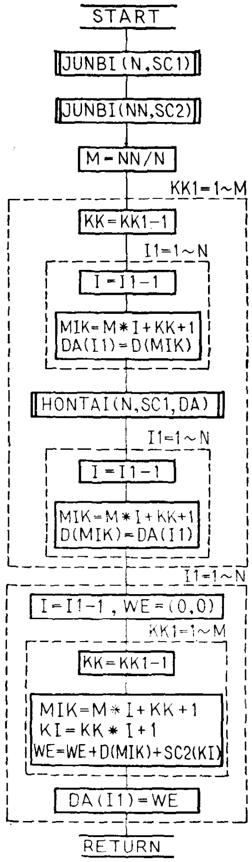

図 3 SUBROUTINE FFT の流れ図

```
      SUBROUTINE JUNBI(N,SC)
      COMPLEX CA,WE,SC(N)
      PI=3.141593
      AA=2.0*PI/N
      DO 10 I1=1,N
      I=I1-1
      A=AA*I
      CA=(0.0,1.0)*A
      WE=CEXP(CA)
      SC(I1)=WE
   10 CONTINUE
      RETURN
      END
```

図 4 SUBROUTINE JUNBI の流れ図

```
      SUBROUTINE HONTAI(N,SC,D1)
      COMPLEX WD,SC(N),D1(N),D2(1000)
      ISW=1
      JJ=N
      L=1
    2 L=L+1
    3 IF(JJ.EQ.1) GO TO 4
      JM=MOD(JJ,L)
      IF(JM)2,5,2
    5 ID=JJ
      JJ=JJ/L
      N1=N/L
      DO 10 I1=1,N,ID
      I=I1-1
   12 IL=I/L
      DO 10 J1=1,JJ
      J=J1-1
      JIL=J+IL
      IX0=I+J+1
      DO 10 IY1=1,L
      IY=IY1-1
      IU=JIL+N1*IY+1
      WD=(0.0,0.0)
      DO 20 IZ=1,L-1
      IP1=(IL+N1*IY)*IZ
      IP=MOD(IP1,N)+1
      IX=IX0+JJ*IZ
      IF(ISW.EQ.1) GO TO 6
      WD=WD+D2(IX)*SC(IP)
      GO TO 20
    6 WD=WD+D1(IX)*SC(IP)
   20 CONTINUE
      IF(ISW.EQ.1) GO TO 7
      D1(IU)=D2(IX0)+WD
      GO TO 10
    7 D2(IU)=D1(IX0)+WD
   10 CONTINUE
   11 ISW=-ISW
      GO TO 3
    4 IF(ISW.EQ.1) GO TO 8
      DO 30 I=1,N
      D1(I)=D2(I)
   30 CONTINUE
    8 RETURN
      END
```

図 5 SUBROUTINE HONTAI のプログラム

```
      SUBROUTINE FFT(NN,N,D,DA)
      COMPLEX D(NN),SC1(100),SC2(100),DA(N),WE
      CALL JUNBI(N,SC1)
      CALL JUNBI(NN,SC2)
      M=NN/N
      DO 10 KK1=1,M
      KK=KK1-1
      DO 20 I1=1,N
      I=I1-1
      MIK=M*I+KK+1
      DA(I1)=D(MIK)
   20 CONTINUE
      CALL HONTAI(N,SC1,DA)
      DO 10 I1=1,N
      I=I1-1
      MIK=M*I+KK+1
      D(MIK)=DA(I1)
   10 CONTINUE
      DO 30 I1=1,N
      I=I1-1
      WE=(0.0,0.0)
      DO 40 KK1=1,M
      KK=KK1-1
      MIK=M*I+KK+1
      KI=KK*I+1
      WE=WE+D(MIK)*SC2(KI)
   40 CONTINUE
      DA(I1)=WE
   30 CONTINUE
      RETURN
      END
```

図 6 SUBROUTINE FFT のプログラム

m) 行なうと，漸化式

$$C_p\{k_0, k_1, \cdots, k_{p-1}, i_{m-p-1}, \cdots, i_0\}$$

$$=\sum_{i_{m-p}=0}^{r_p-1} C_{p-1}\{k_0, k_1, \cdots, k_{p-2}, i_{m-p}, \cdots, i_0\}$$

$$\times W^{i_{m-p}(r_{p+1}, \cdots, r_m)\{k_{p-1}(r_1 r_2 \cdots r_{p-1})+\cdots+k_0\}}$$

が得られるので，$p=m$ のとき DFT を得ることができる．(SUBROUTINE HONTAI (N, SC, D) 参照.)

このとき，前述の演算回数Pはつぎのようになる．

$P=N(r_1+r_2+\cdots+r_m)$　　：r が素数のとき

$=rN\log_r N$　　：$r_1=r_2=\cdots=r_m=r$ のとき

$=N^2$　　：N が素数のとき

2. 演算時間低減のためのアルゴリズム

データ数が $N=m\cdot n$ であり，$(m-1)$ 次の高調波成分まで，すなわち

$$C_k=\sum_{i=0}^{N-1} X_i W^{ik} \qquad (k=0, 1, \cdots, m-1)$$

を求めることを考える．上式を展開すると

$$C_k=\sum_{i=0}^{mn-1} X_i W^{ik}$$

$$=\sum_{i=0}^{m-1} X_{ni} W^{nik}+\sum_{i=0}^{m-1} X_{ni+1} W^{(ni+1)k}+\cdots$$

$$+\sum_{i=0}^{m-1} X_{ni+n-1} W^{(ni+n-1)k}$$

となるが，ここで

$$F_{lk}\equiv\sum_{i=0}^{m-1} X_{ni+l} W^{ink},$$

$$k=0, 1, \cdots, m-1,$$

$$l=0, 1, \cdots, n-1,$$

とおけば，

$$C_k=F_{0k}+F_{1k}W^k+\cdots+F_{lk}W^{lk}+\cdots$$

$$+F_{(n-1)k}W^{(n-1)k}$$

$$=\sum_{l=0}^{n-1} F_{lk} W^{lk} \qquad (3)$$

と表現できるから，F_{lk} を前述の FFT で求め，式(3)の総和をとればよいことになる．(SUBROUTINE FFT (NN, N, D, DA) 参照.)

3. 流れ図およびプログラム

上述の考え方に基づいて作成した流れ図およびプログラムを図1～6に示す．

SUBROUTIN JUNBI (N, SC)

これは回転因子 W^i $(i=0, 1, \cdots, N-1)$ を計算するサブルーチンであり，N にデータ数を与えて呼出せば，SC (複素配列) に回転因子をのせて返ってくる．

SUBROUTIN HONAI (N, SC, D)

式(1)を計算するサブルーチンで，Nにデータ数，SC (複素配列) に回転因子，D (複素配列) に時系列データを与えて呼び出すと，$D(k)$ に C_{k-1} $(k=1, 2,$

…, N) なる解を与えて戻る.

SUBROUTINE FFT (NN, N, D, DA)

NN にデータ数, N に求めたい最高高調波周波数(ただし, N は NN の約数), Dに時系列データを与えて呼ぶと, DA に FFT の解を与えて返る.

以上, いずれも Level 7000 の FORTRAN 言語であり, FACOM 230-60 を使用した.

参考文献

1) J. W. Cooley & J. W. Tukey : "An algorithm for the machine calculation of complex Fourier series," Mathematics Computation, Vol. 19, No. 90, pp. 297〜301 (1965).
2) 小椋靖夫・平松啓二 : "任意の基数を用いた FFT のアルゴリズム," 情報処理, Vol. 13, No. 2, pp. 122〜124 (1971).

(昭和47年5月8日受付)
(昭和47年6月6日再受付)