プログラムのページ

担当 吉 沢 正

7101 数式処理言語 (FORMAC) による因数分解

桧山 澄子 (東京大学地震研究所)

0. 問 題

z: 有理整数環, $f(x)$: z の元を係数とする n 次多項式 (i. e. $f(x)\in z[x]$)

$$f(x)=x^n+a_1x^{n-1}+\ldots+a_n.$$

を $z[x]$ で素因子 $g_1(x)g_2(x)\ldots g_m(x)$ に分解すること. ただし, $\dim(g_i(x))\leq 6$ とする.

1. 算法のアルゴリズム

全体の流れは**図1**のとおりである. 計算のステップごとにいくつかの internal procedure に分けられるので以下各 Procedure ごとに機能や算法を述べる.

(a) プロシージャー: SHONE

目的　$f(x)$ の1次因子を求める.

計算方法　$f(x)$ の定数項 a_n の素因数分解を行ない, その約数すべてを $f(x)$ に代入し, $f(x)=0$ かどうかをしらべる. もし $f(p_i)=0$ なら, $f(x)$ は $x-p_i$ なる因子をもつといえる.

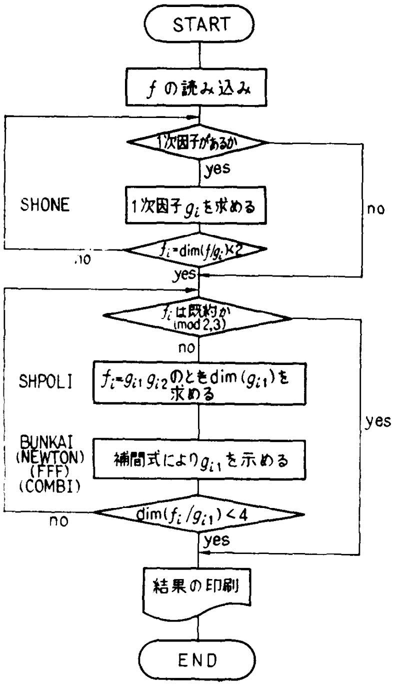

図 1

(b) プロシージャー: SHPOLI

目的　2次以上の因子をもつかどうかを判定し, もつ場合は, その因子の次数を求める.

計算方法　$z/(2)[x]$* で $f(x)$ が既約かどうかを判別する. $z/(2)[x]$ 考えたのは, そこでは与えられた次数の多項式が有限個しかないので, 既約かどうかの判定は, 有限個の既約な多項式で実際に割ってみればよいからである.

mod 2 で n 次の既約因子をもつということは, $z[x]$ で m 次の因子をもつということの必要条件でしかないので, mod 2 で $f(x)$ が m 次の既約因子をもつことがわかったとき, mod 3 でも $f(x)$ が m 次の既約因子をもつかどうかをみる.

ともに m 次の既約因子をもつときは, $z[x]$ でも m 次の因子をもつ可能性があるので, その m 次の多項式を求める次のプロシージャーにうつる.

(c) プロシージャー: BUNKAI

目的　与えられた任意の m 次既約因子を求める.

計算方法　0, ±1, ±2,... なる $m+1$ 個の値を $f(x)$ に代入し, 各 $f(m_i)$ について約数 $q_{i1},\ldots,q_{im}$ を求める.

さて, そこで $f(x)$ が $g(x)$ なる因子をもったとすると, $g(m_i)|f(m_i)$ であるはずだから, $g(m_i)$ は約数 q_{ij} のいずれかと一致する. そこで, $x=m_i$ のとき $g(m_i)=q_{ij}$ $(j=1,\ldots,m)$ の各場合について, ニュートンの補間式を利用して $g(x)$ を求める.

この際 NEWTON なるプロシージャーを使う.

次に求めた $g(x)$ が実際に $f(x)$ の因子になっているかどうかを, 割ることによってたしかめる.

(d) プロシージャー: FFF

目的　与えられた係数と次数に対し, 多項式をつくる.

(e) プロシージャー: COMBI

* $z/(2)[x]$ とは 0, ±1 を係数とする多項式の集まりのこと.

プログラム

```
INPUT TO FORMAC PREPROCESSOR
TESHON:PROC OPTIONS(MAIN);
    FORMAC_OPTIONS;
  DCL DENFMC3 ENTRY(BINARY FIXED(31),BINARY FIXED(31));
  DCL IFC FIXED(15),(IP(15),IPRIM(7,200),NUM(7),MO,
        II,IPR(100),IRAM(31))
                          FIXED DEC,F CHAR(72),(N,M) FIXED BIN(31);
  ON ENDFILE(SYSIN)GO TO OWARI2;
  L1:GET LIST(F);LET(F="F");
     PUT EDIT('POLINOMIAL TO BE FACTORIZED')(SKIP,A);
    PRINT_OUT(F);
        LET(ANS=1;G1=F);
L2:CALL SHONE(ISW);
     IF ISW=0 THEN GO TO LSH1;
     LET(ANS=ANS*G2);
       LET(N=HIGHPOW(G1,X));IF INTEGER(N)>=2 THEN DO;
      LET(F=G1);GO TO L2;END;
  LSH1:LET(N=HIGHPOW(F,X));
     IF INTEGER(N)<=3 THEN DO;
     LET(ANS=ANS*G1);
    GO TO OWARI;END;
     CALL SHPOLI;
     IF M¬=0 THEN GO TO CAL;
     IF M=0 & INTEGER(N)/2 <=3 THEN DO;
       LET(ANS=ANS*G1);GO TO OWARI;END;
      ELSE DO M=4,5,6;
   CAL:CALL BUNKAI;
     LET(ANS=ANS*IG);
      LET(F=G1);
       GO TO LSH1;
SHONE:PROC(ISW);
     DCL IPRIM(100) FIXED DEC,I FIXED BIN;
   ISW=0;
      LET(FO=EVAL(F,X,0));
          IFO=INTEGER(FO);
          IF IFO=0 THEN DO;
   ISW=1;
       LET(G1=F/X;G2=X);RETURN;END;
   ELSE DO;IFO=ABS(IFO);
     CALL PRIM(IFO,IPRIM);
     DO I=1 TO II;
      IPRIM(II+I)=-IPRIM(I);END;
             II=II+II;
      DO I1=1 TO II;
       LET(IPRIM="IPRIM(I1)");
       LET(FO=EVAL(F,X,IPRIM));
        IF INTEGER(FO)=0 THEN DO;
        LET(G1=0;G2=X-IPRIM;FO=F);
  L1:LET(L=HIGHPOW(FO,X);LC=COEFF(FO,X**L);LL=L-1);
    IF INTEGER(LL)>=0 THEN DO;
      LET(LC=LC*X**LL);
     LET(G1=G1+LC;FO=FO-LC*(X-IPRIM);FO=EXPAND(FO));
     GO TO L1;END;
     ISW=1;RETURN;
             END;END;END;
      END SHONE;
SHPOLI:PROC;
     DCL L(3) LABEL;
     DCL (I,J) FIXED BIN;
     NN=2;M=0; /* IF M=0 THEN THIS POLYNOMIAL IS IRREDUCIBLE */
```

```
      LET(P(1,1)=X**2+X+1;P(2,1)=X**3-X*X-1;P(2,2)=X**3-X-1);
      DO I=1 TO 2;
      DO J=1 TO  2;
      IF I=1&J>=2 THEN GO TO L1;
      LET(I="I";J="J";Q=P(I,J);R=F);
      LET(LQ=HIGHPOW(Q,X));
   L2:LET(L=HIGHPOW(R,X);LC=COEFF(R,X**L);LL=L-LQ);
      LL=INTEGER(LL);IF LL>=0 THEN DO;
      LET(R=R-LC*Q*X**LL;R=EXPAND(R));
      GO TO L2;END;
      LET(NO=COEFF(R,X**0);N1=COEFF(R,X);N2=COEFF(R,X**2));
      NO=INTEGER(NO);N1=INTEGER(N1);N2=INTEGER(N2);
        IF MOD(NO,NN)=0 & MOD(N1,NN)=0 & MOD(N2,NN)=0 THEN DO;
        M=INTEGER(LQ);
         GO TO L3;END;
   L1:END;END;
  RETURN;
   L3:NN=3;
     /GO TO L(M);
L(2):LET(P(1,1)=X**2+1;P(1,2)=X**2-X-1;P(1,3)=X**2+X-1);
      DO J=1 TO 3 ;
      LET(J="J"; Q=P(1,J);R=F);
   L4:LET(L=HIGHPOW(R,X);LC=COEFF(R,X**L);LL=L-2 );
      LL=INTEGER(LL);IF LL>=0 THEN DO;
       LET(R=R-LC*Q*X**LL; R=EXPAND(R));
        GO TO L4; END;
      LET(NO=COEFF(R,X**0);N1=COEFF(R,X));
       NO=INTEGER(NO);N1=INTEGER(N1);
      IF MOD(NO,NN)=0 & MOD(N1,NN)=0 THEN GO TO L6;END;
         M=0;RETURN;
   L6:M=2;RETURN;
L(3):LET(P(1,1)=X**3-X*X-X-1;P(1,2)=X**3-X*X+1;P(1,3)=X**3-X*X+X+1;
         P(1,4)=X**3-X-1;P(1,5)=X**3-X+1;P(1,6)=X**3+X*X-X+1;
         P(1,7)=X**3+X*X-1; P(1,8)=X**3+X*X+X-1);
      DO J=1 TO 8;
      LET(J="J"; Q=P(1,J);R=F);
   L5:LET(L=HIGHPOW(R,X);LC=COEFF(R,X**L);LL=L-3);
      LL=INTEGER(LL); IF LL>=0 THEN DO;
      LET(R=R-LC*Q*X**LL; R=EXPAND(R));
      GO TO L5; END;
     LET(NO=COEFF(R,X**0);N1=COEFF(R,X);N2=COEFF(R,X**2));
     NO=INTEGER(NO);N1=INTEGER(N1);N2=INTEGER(N2);
      IF MOD(NO,NN)=0 & MOD(N1,NN)=0 & MOD(N2,NN)=0 THEN
      GO TO L7;END;
       M=0;RETURN;
   L7:M=3;RETURN;
    END SHPOLI;
  BUNKAI:PROC;
      MM=M+1; K=-M/2-1;
      DO I=1 TO MM; K=K+1;
      LET(IFO=EVAL(F,X,"K"));
      IFO=INTEGER(IFO);
      IFO=ABS(IFO);
     CALL PRIM(IFO,IP);
      DO J=1 TO II;IPRIM(I,J)=IP(J);END;
      CALL COMBI;
      NUM(I)=MO+MO;
      JM=II+1;IND=0;
      DO JACK=JM TO MO;
      IND=IND+1;
```

```
    IPRIM(I,JACK)=IPR(IND);END;
    DO J=1 TO MO;KKK=MO+J;
    IPRIM(I,KKK)=-IPRIM(I,J);END;
    III=NUM(I);
  END;
    DO I=7 TO MM+1 BY -1;
       IPRIM(I,1)=0;NUM(I)=1;
    END;
   M4=NUM(7);M5=NUM(6);M6=NUM(5);M7=NUM(4);M8=NUM(3);M9=NUM(2);
    M10=NUM(1);
    DO I4=1 TO M4;IP(7)=IPRIM(7,I4);DO I5=1 TO M5;IP(6)=IPRIM(6,I5);
 DO I6=1 TO M6;IP(5)=IPRIM(5,I6);DO I7=1 TO M7;IP(4)=IPRIM(4,I7);
 DO I8=1 TO M8;IP(3)=IPRIM(3,I8);DO I9=1 TO M9;IP(2)=IPRIM(2,I9);
  DO I10=1 TO M10;IP(1)=IPRIM(1,I10);
    CALL NEWTON(ISW);
    IF ISW=0 THEN GO TO HIYAK;
    CALL FFF;
    LET(IG=EXPAND(IG));
    LET(R=F;G1=0);
 L1:LET(L=HIGHPOW(R,X);LG=HIGHPOW(IG,X);
    LC=COEFF(R,X**L);LL=L-LG);
     N=INTEGER(LL); IF N>=0 THEN DO;
    LET(LC=LC*X**LL;G1=LC+G1;R=R-LC*IG;R=EXPAND(R));
    GO TO L1; END;
    IF IDENT(R;0) THEN RETURN;
HIYAK:END;END;END;END;END;END;END;
COMBI:PROC;
    DCL (J,K,L) FIXED BIN;
    MO=II;IND=0;
    IF II<=2 THEN GO TO HIYAK;
    DO J=2 TO II;DO K=2 TO II;
    IF K<=J THEN GO TO LC1;
    MO=MO+1;IND=IND+1;IPR(IND)=IP(J)*IP(K);
 LC1:END;END;
     IF II<=3 THEN GO TO HIYAK;
    DO J=2 TO II;DO K=2 TO II;DO L=2 TO II;
    IF L<=K|K<=J THEN GO TO LC2;
    MO=MO+1;IND=IND+1;
    IPR(IND)=IP(J)*IP(K)*IP(L);
LC2:END;END;END;
    IF II<=4 THEN GO TO HIYAK;
  IF MOP<=L|L<=K|K<=J THEN GO TO LC3;
    DO J=2 TO II;DO K=2 TO II;DO L=2 TO II;DO MOP=2 TO II;
    MO=MO+1;IDN=IND+1;
  IPR(IND)=IP(MOP)*IP(L)*IP(K)*IP(J);
LC3:END;END;END;END;
    IF II<=5 THEN GO TO HIYAK;
    DO J=2 TO II;DO K=2 TO II;DO L=2 TO II;DO MOP=2 TO II;
  DO NOP=2 TO II;
  IF NOP<=MOP|MOP<=L|L<=K|K<=J THEN GO TO LC4;
    MO=MO+1;IND=IND+1;
  IPR(IND)=IP(J)*IP(K)*IP(L)*IP(MOP)*IP(NOP);
LC4:END;END;END;END;END;
HIYAK:END COMBI;
 NEWTON:PROC(ISW);
  DCL IPR(50,50);
    DCL (I,J) FIXED BIN;
  ISW=1;
    IRAM(1)=IP(1);INK=MM;
 DO I=1 TO INK;
```

```
      IPR(I,1)=IP(I);END;
      DO J=1 TO M;
        NN=M-J+1;
    DO I=1 TO NN;
      IPR(I,J+1)=IPR(I+1,J)-IPR(I,J);
      END;
       END;
   KING=1;
   DO I=1 TO M;
      KING=KING*I;
    IRAM(I+1)=IPR(1,I+1)/KING;
    END;
      IF IRAM(INK)=0|IRAM(INK)¬=1 THEN GO TO LN1;
      RETURN;
 LN1:ISW=0;
      END NEWTON;
  FFF:PROC;
      DCL (I,K,II) FIXED BIN;
       K=MM/2;II=MM+1;
      LET(IG=0);
      DO I=1 TO M;
      MI=K-I;
      LET(IG=(IG+"IRAM(II-I)")*(X-"MI"));
       END;
     LET(IG=IG+"IRAM(1)");
    END FFF;
      END BUNKAI;
   PRIM:PROC(N,IPRIM);
      DCL N FIXED(15),IPRIM(15) FIXED DEC,
           I FIXED BIN,
   IPR(100) FIXED DEC INITIAL(2,3,5,7,11,13,17,19,23,29,
   31,37,41,43,47,53,59,61,67,71,
   73,79,83,89,97,101,103,107,109,113,
   127,131,137,139,149,151,157,163,167,173,
     179,181,191,193,197,199,211,223,227,229,
    233,239,241,251,257,263,269,271,277,281,
   283,293,307,311,313,317,331,337,347,349,
   353,359,367,373,379,383,389,397,401,409,
    419,421,431,433,439,443,449,457,461,463,
      467,479,487,491,499,503,509,521,523,541) ;
   NN=N;II=1;IPRIM(II)=1;
  DC I=1 TO 100;
  IF IPR(I)**2>NN THEN GO TO PON;
  ELSE DO;
  LA:NM=NN/IPR(I);
   IF NN-NM*IPR(I)=0 THEN DO;
   NN=NM;II=II+1;
  IPRIM(II)=IPR(I);GO TO LA;
   END;END;END;
   PON:II=II+1;IPRIM(II)=NN;
     END PRIM;
       END;LET(ANS=ANS*G1);
    OWARI:PRINT_OUT(ANS);
     GO TO L1;
  OWARI2:END TESHON;
```

目的　与えられたm個の素数群 $p_1,\dots,p_m$ に対し，各組合せの積を計算する．すなわち，$a=p_1^{e_1}\dots,p_m^{e_m}$ と素因数分解されたとき，$(e_1+1)(e_2+1)\dots(e_m+1)$ 個の約数すべてを求める．プロシージャー BUNKAI で使用する．

(f)　プロシージャー：PRIM

計算結果

```
POLINOMIAL TO BE FACTORIZED
         2       3       4       5       6     7
F = 6 X   - 4 X   - 9 X   - 3 X   + 2 X   + X   - 8
-------------------------------------------------------
                 2                   3
ANS = ( 2 X + X   + 2 ) ( - X + X   + 1 ) ( X + 2 ) ( X - 2 )
--------------------------------------------------------------

POLINOMIAL TO BE FACTORIZED
               2     3       4       5     6     7
F = 2 X - 2 X   - X   + 3 X   - 2 X   - X   + X   - 1
-------------------------------------------------------
                3               2     3     4
ANS = ( - X + X   + 1 ) ( X - X   - X   + X   - 1 )
-------------------------------------------------------

POLINOMIAL TO BE FACTORIZED
           2
F = X + X   + 1
---------------
             2
ANS = X + X   + 1
-----------------

POLINOMIAL TO BE FACTORIZED
          2        4       6        8     10
F = 26 X   - 51 X   + 2 X   - 25 X   + X    - 25
-------------------------------------------------
                4                                     4
ANS = ( - X + X   + 1 ) ( X + 5 ) ( X - 5 ) ( X + X   + 1 )
-------------------------------------------------------------
```

目的　整数 a の素因数を求める．プロシージャー SHONE, BUNKAI で使用する．

2. プログラム

使用言語　PL/I-FORMAC

機　種　IBM 360/75

計算時間：

Compile time 0.24 mins, Elaps time 0.82 mins, CPU time 1.26 mins, Core time 2.40 mins.

Core size：200 K バイト

3. 結　果

ここでは，プロシージャー SHPOLI で既約な多項式は3次式までしか記憶してないので，3次の既約式までしか求まらないが，実際は mod 2 のとき3個の4次式，5次式は6個を記憶させしらみつぶしに計算することをさけた．ただし，mod 3 のときは既約な4次式は21個，5次式は63個と多いので，2次記憶装置に記憶し，FORMAC の命令 SAVE, ATOMIZE を使用すればよいと思うが，これは試みていない．

一般に因数分解する場合，因子が高次になると演算回数が爆発的にふえてしまう．ここではプロシージャー SHPOLI, BUNKAI で因子になりうる可能性のある多項式をふるい落しているので，一応高次の場合も有効である．しかし，たとえそれらを使って $f(x)$ が m 次の因子をもつことがわかったときでも，0, ±1, ±2, ... なる $m+1$ 個の値を $f(x)$ に代入したとき，$f(m_i)$ が平均 n 個の約数をもてば，n^{m+1} 回ニュートンの補間式を作り，実際に因子になっているかどうかをみなければならない．こうしたことから因子の次数が6次くらいまでが限界ではないかと思う．

参考文献

1) Van der Waerden: Moderne Algebra Vol. 1, pp. 73～89, Springer-Verlag, 1950.
2) R. J. McEliece: Factorization of Polynomials over Finite Fields, Math. Comp. Vol. 23, pp. 861-867, 1969.
3) 渡辺隼郎：数式処理による常微分方程式の解法のためのプログラミング技法，第9回プログラミングシンポジウム報告集，pp. 43～49, 1968年1月.

(昭和46年1月5日受付)