プログラムのページ

77-04 簡単な数式処理言語 FORMAL による部分分数展開プログラム

植田 健治*, 永田 守男*

1. はじめに

数式処理の研究は古くからおこなわれ, MIT では MACSYMA[1] のような大がかりなシステムも作られている. 国内でもいままでに実験的なシステムが作成され, FORMAC[2] が使われたりしてきた. 最近, 通研で AL[3] が開発され, ユタ大学の REDUCE[4] が東大で使えるようになった. その他開発中のものもいくつかあるように聞いている.

われわれは, メリーランド大学の簡単な数式処理言語 FORMAL[5] を入手したので, 慶応義塾大学の UNIVAC 1106 上に, 有理関数の不定積分アルゴリズム等で重要な部分分数展開のプログラムを作成した. この言語の紹介を兼ねてプログラムと実行結果を示す.

2. 数式処理言語 FORMAL

FORMAL は FORTRAN V に機能を付加した形の簡単な数式処理言語で, メリーランド大学の UNIVAC 1108 にインプリメントされている.

2.1 FORMAL の言語構成

FORTRAN V の文の中に引用符' でくくられた数式処理のための文 (F-文と呼ぶ) を挿入することができる. (プログラム例参照). FORMAL の処理系は F-文を読み込むと, これを処理するサブルーチンを呼び出すための CALL 文に置き換える. すべての文を FORTRAN の文にして FORTRAN コンパイラに渡す. このように FORMAL の処理系はプリプロセッサとサブルーチン群からできている.

2.2 数式の内部表現

FORMAL が処理の対象とする数式は, 計算機内部で演算子前置の n 進木構造で表わされる. f を演算子, $x_1, \cdots, x_n$ をオペランドとした式の例を図-1 に示す.

算術演算子として, + (加算), * (乗算), ** (べき乗) の 3 種類を使い, $-e$ は $(-1)*e$, e_1-e_2 は $e_1+(-1)*e_2$, e_1/e_2 は $e_1*e_2**(-1)$ で表現する. ここで e, e_1, e_2 は任意の数式を示す.

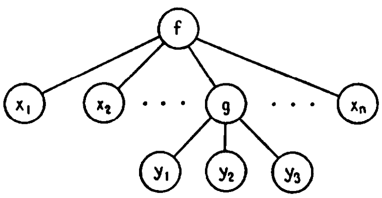

図-1 数式の内部表現

数式を表わす木構造のすべての部分木に対して主演算子 (lead operator) という概念 (図-1 の木全体では f) があり, その種類により整数値が対応する. 交換則や結合則の成立する + と * の下のオペランドには一定の規則で順序がつけられている.

2.3 数式処理機能

FORMAL では, 式の展開, 係数の抜き出し, 置換, 分母と分子の抜き出し, 微分, 部分式の抜き出し等の機能が用意されている. あとで示すプログラムに関係するものを表-1 にまとめておく.

表-1 部分分数展開プログラムで使う数式処理機能

1. FORMAL ステートメント	
'V=E'	V に E を代入する
'READ V'	数式をひとつ読み込み V に代入する
'PRINT V'	V に代入されている式を 'V=...' の形で印刷する
'OPTION EXPAND (N)'	分配律及び二項展開の自動的な適用の範囲を定める N=0: 何も適用しない N=1: 分配律のみの適用 N>1: 分配律及び N 以下の指数部をもつ式への二項展開の適用
2. FORMAL 関数とその値	
EXPAND [E]	E を展開した式
ARG [E, N]	E の主オペレータに対する N 番目の引数
NARG [E]	E の主オペレータに対する引数の数
NUM [E]	E の分子
DENOM [E]	E の分母
BASE [E]	E の底
EXPON [E]	E の指数部分
COEFF [E, X]	E における X の係数
CODEM [E]	E を通分した式
3. FORTRAN の値をとる関数とその値	
LDOP ('E')	E の主オペレータに対応する整数値 (プログラム参照)
NARG ('E')	E の主オペレータに対する引数の数
IVALUE ('E')	整数 E
IDENT ('E1, E2')	E1 と E2 が等しいとき .TRUE., そうでないとき .FALSE.
4. その他	
#I (...)	FORMAL ステートメント内での整数型の FORTRAN 式の使用を示す

* 慶応義塾大学工学部

3. 部分分数展開プログラム

既約な1変数整数係数多項式を分母と分子に持つ式を，未定係数法で有理数係数の部分分数に展開するプログラムを説明する．

3.1 部分分数展開

$p_i(x)$, $q(x)$ を，x を変数とする整数係数多項式としたとき次の形の式が与えられたとする．

$$\frac{q(x)}{p_1(x)\cdot p_2(x)\cdots p_n(x)} \tag{3.1}$$

この式と同等な次のような式をみつける．

$$\frac{r_1(x)}{p_1(x)}+\frac{r_2(x)}{p_2(x)}+\cdots+\frac{r_n(x)}{p_n(x)} \tag{3.2}$$

```
C       AN INTEGER CORRESPONDING TO
C            THE LEAD OPERATOR OF THE F-EXPRESSION :
C  CONSTANTS (ZERO 1/INTEGER 2/RATIONAL 3/REAL 4),
C  VARIABLE 6-7, +  18, *  19, **  20, DIFF. OPERATOR  23,
C  EXP 32,LOG 33,TNH 34,SIN 35,COS 36,ATN 37,FAC 38,BIN 39,STEP 40
      LOGICAL IDENT
      DIMENSION ID(15),T(20),IORDER(20)
      DIMENSION IT(20,2),IQ(12),IPN(12),IQN(12),C(20.21)
C <<<<<<<<<<<<<<<<<<<<<<<<  1  >>>>>>>>>>>>>>>>>>>>>>>
     'OPTION EXPAND(0)'
    1 PRINT '(IX,72''H'')'
     'READ E'
      IF(IDENT('E,0')) STOP
      PRINT '('' THE ORIGINAL EXPRESSION : '')'
     'PRINT E'
     'NE = NUM[E]'
     'DE = DENOM[E]'
C <<<<<<<<<<<<<<<<<<<<<<<<  2  >>>>>>>>>>>>>>>>>>>>>>>
      NA = NARG('DE')
      IDS = 0
      DO 250 J = 1,NA
        'EJ = ARG[DE,#I(J)]'
         IF(.NOT. IDENT('EJ,X')) GO TO 210
         ID(J) = 1
         GO TO 240
  210    LOEJ = LDOP('EJ')
         IF(LOEJ .EQ. 18) GO TO 220
         IF(LOEJ .NE. 20) GO TO 999
        'IB = BASE[EJ]'
        'IE = EXPON[EJ]'
         IF(LDOP('IE') .NE. 1) GO TO 999
         IF(IDENT('IB,X'))     GO TO 230
         IEF = IVALUE('IE')
        'OPTION EXPAND(#I('IEF'))'
        'EJ = EXPAND[EJ]'
  220   'LASTEJ = ARG[EJ,NARG[EJ]]'
         IF(LDOP('LASTEJ') .EQ. 19) 'LASTEJ = ARG[LASTEJ,2]'
        'IE = EXPON[LASTEJ]'
         IF(LDOP('IE') .NE. 1)GO TO 999
  230    ID(J) = IVALUE('IE')
  240   'A(#I(J)) = EJ'
  250    IDS = IDS + ID(J)
     'OPTION EXPAND(1)'
      DO 260 J = 1,NA
        'B(#I(J)) = 1'
         DO 260 L = 1,NA
            IF(L .EQ. J) GO TO 260
           'B(#I(J)) = B(#I(J)) * A(#I(L))'
  260 CONTINUE
     'OPTION EXPAND(0)'
      I = 0
      IDSD = IDS - 1
      DO 285 J = 1,NA
         I = I + 1
         C(1,I) = IVALUE('COEFF[B(#I(J)),X]')
         DO 280 L = 2,IDSD
  280       C(L,I) = IVALUE('COEFF[B(#I(J)),X**#I(L)]')
     5   + (1 - 2*MOD(L,2)) * IVALUE('COEFF[B(#I(J)),(-X)**#I(L)]')
         C(IDS,I) = IVALUE('ARG[B(#I(J)),1]')
         IF(ID(J) .EQ. 1) GO TO 285
         IDJ1 = ID(J) - 1
         DO 284 K = 1,IDJ1
            I = I + 1
            DO 282 L = 2,IDSD
  282          C(L,I) = C(L-1,I-1)
            C(1,I) = C(IDS,I-1)
  284       C(IDS,I) = 0
  285 CONTINUE
      IDSI = IDS + 1
      DO 290 J = 1,IDSD
  290    C(J,IDSI) = IVALUE('COEFF[NE,X ** #I(J)]')
      C(IDS,IDSI) = 0
      IF(LDOP('NE') .NE. 19) C(IDS,IDSI) = IVALUE('ARG[NE,1]')
C <<<<<<<<<<<<<<<<<<<<<<<<  3  >>>>>>>>>>>>>>>>>>>>>>>
      DO 310 J = 1,IDS
  310    IORDER(J) = J
      DO 390 L = 1,IDS
         L1 = L + 1
         MAXJ = L
         MAXK = L
         XAM = ABS(C(L,L))
         DO 320 J = L,IDS
            DO 320 K = L,IDS
               IF(ABS(C(MAXJ,MAXK)) .GE. ABS(C(J,K))) GO TO 320
               MAXJ = J
               MAXK = K
               XAM = ABS(C(J,K))
  320    CONTINUE
         IF(MAXJ .EQ. L) GO TO 340
         DO 330 K = L,IDSI
            XAM = C(L,K)
            C(L,K) = C(MAXJ,K)
  330       C(MAXJ,K) = XAM
  340    IF(MAXK .EQ. L) GO TO 360
         DO 350 J = 1,IDS
            XAM = C(J,L)
            C(J,L) = C(J,MAXK)
  350       C(J,MAXK) = XAM
         IO = IORDER(L)
         IORDER(L) = IORDER(MAXK)
         IORDER(MAXK) = IO
  360    DO 370 K = L1,IDSI
  370       C(L,K) = C(L,K) / C(L,L)
         DO 390 J = 1,IDS
            IF(J .EQ. L) GO TO 390
            DO 380 K = L1,IDSI
  380          C(J,K) = C(J,K) - C(J,L) * C(L,K)
  390 CONTINUE
      DO 395 K = 1,IDS
         J = IORDER(K)
  395    T(J) = C(K,IDSI)
C <<<<<<<<<<<<<<<<<<<<<<<<  4  >>>>>>>>>>>>>>>>>>>>>>>
      E = 1.E-5
      Z = 10.E7
      DO 420 J = 1,IDS
         S = ABS(T(J))
         ISIGN = 1
         IF(T(J) .LT. 0.) ISIGN = -1
         IT(J,1) = 0
         IT(J,2) = 1
         IF(S .LT. E) GO TO 420
         DO 411 K = 1,10
            ND = K - 1
            X = S * FLOAT(10 ** ND)
            IF(Z .LT. X) GO TO 412
  411    CONTINUE
  412    IY = 10 ** ND
         IX = X
         DO 413 K = 2,12
            KD = K
            IQ(K)  = IX / IY
            IR = MOD(IX,IY)
            IF(IR .EQ. 0) GO TO 414
            IX = IY
  413       IY = IR
  414    IPN(1) = 1
         IPN(2) = IQ(2)
         IQN(1) = 0
         IQN(2) = 1
         ND = 2
         IF(ABS(FLOAT(IQ(2)) - S) .LT. E) GO TO 416
         DO 415 K = 3,KD
            ND = K
            IPN(K) = IPN(K-1) * IQ(K) + IPN(K-2)
            IQN(K) = IQN(K-1) * IQ(K) + IQN(K-2)
            IF(ABS(FLOAT(IPN(K))/FLOAT(IQN(K)) - S) .LE. E) GO TO 416
  415    CONTINUE
  416    IT(J,1) = ISIGN * IPN(ND)
         IT(J,2) = IQN(ND)
  420 CONTINUE
C <<<<<<<<<<<<<<<<<<<<<<<<  5  >>>>>>>>>>>>>>>>>>>>>>>
      I = 0
     'E = 0'
      DO 540 J = 1,NA
        'F = 0'
         DO 530 K = 1,ID(J)
            I = I + 1
  530      'F = F + (#I(IT(I,1))/#I(IT(I,2))) * X**#I(K - 1)'
  540   'E = E + CODEM[F] / ARG[DE,#I(J)]'
      PRINT '('' THE PARTIAL FRACTION DECOMPOSITION OF E :'')'
     'PRINT E'
      GO TO 1
C
  999 PRINT '('' ***** ILLEGAL EXPRESSION !! *****'')'
      GO TO 1
      END
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
THE ORIGINAL EXPRESSION :
E =
1 / (  (1 - X) * (2 - X) * (3 + 2 * X + X ** 2) )
THE PARTIAL FRACTION DECOMPOSITION OF E :
E =
(3/22 + 5/66 * X) / (3 + 2 * X + X ** 2) + 1/6 / (1 - X) - 1/11 / (2 - X)
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
THE ORIGINAL EXPRESSION :
E =
(1 + X ** 3) / (X * (- 1 + X) ** 3)
THE PARTIAL FRACTION DECOMPOSITION OF E :
E =
1 / ( - 1 + X) ** 3 * (3 - 3 * X + 2 * X ** 2) - 1 / X
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
```

図-2 部分分数展開プログラムと例題

ただし，$\deg(q(x)) < \sum_{i=1}^{n} \deg(p_i(x))$，$\deg(r_i(x)) < \deg(p_i(x))$ $(1 \leqq i \leqq n)$ とする．$\deg(p(x))$ は $p(x)$ の x に関する最高次数を示す．

(3.2)式を通分して(3.1)式と等しいとおくと次の式が得られ，x の各次数の係数に関する連立方程式を解いて $r_i(x)$ を決定する．

$$q(x) = \sum_{i=1}^{n} \left(r_i(x) \prod_{\substack{j=1 \\ j \neq i}}^{n} p_j(x) \right) \tag{3.3}$$

3.2 プログラム

ふたつの例題を付したプログラムを**図-2**(前頁参照)に示す．このプログラムは次の5つの部分から成る．

(1)数式を読み込み，分母と分子に分離し，(2)連立方程式を立て，(3)初等的な掃き出し法[6]で解き，浮動小数点数の解を求め，(4)この解を連分数展開し[7]有理数の形にし，(5)部分分数展開した数式を生成して出力する．

4. おわりに

FORMAL は，ここに示したような簡単な数式処理を手軽におこなうためには便利な道具で*，数値計算の前処理にも活用できる．

しかし，出力が FORTRAN の数式の形なので人間にとって見にくい，プリプロセッサ方式のため実行時間が長くなる，ユーザが新しい機能を作成するときに再帰的な手続が書けないなどの欠点がある．

最後に，要請に応じて FORMAL の処理系を送ってくださったメリーランド大学の Prof. Mesztenyi，慶応義塾大学の UNIVAC 1106 への移植に協力いただいた原田賢一講師と中田淳子さんに感謝する．

* FORMAL には会話型の版もある．

参考文献

1) MATHLAB Group: MACSYMA User's Manual, Project MAC, MIT, Cambridge, Mass. (1973)
2) IBM Corp.: PL/I-FORMAC Symbolic Mathematics Interpreter, Contributed Program Library, 360D-03.3.004 (1969)
3) 池原，岡田：数式処理言語 AL の特徴とその意義，第 17 回プログラミング・シンポジウム報告集，pp. 179〜190 (1976)
4) Hearn, A. C.: REDUCE 2 User's Manual, Univ. of Utah (1973)
5) Mesztenyi, C. K.: FORMAL User's Manual, CN-1.1, Univ. of Maryland (1971)
6) 浦 昭二：FORTRAN 入門 (改訂版)，培風館，東京 (1972)
7) 和田秀男：連分数と近似値，数学セミナー，Vol. 15, No. 4, pp. 54〜58 (1976)

(昭和 52 年 4 月 5 日受付)
(昭和 52 年 7 月 6 日再受付)