

差分化された 2 次元戸田格子の解の行列式表現と

パッケージ C o e f f 2

広 大 工 広 田 良 吾 (Ryogo Hirota)

加 古 富 志 雄 (Fujio Kako)

数式処理を実際の研究に使ってみると、式の爆発が起こって計算不能になる場合が多い。これは数式処理による式の表現が不必要にすべての項を展開していることによる。この困難をさけるために、新しいパッケージ coeff2 が作成された。

coeff2 は関数 f を指定したカーネル例えば x, y のみで展開し、その係数をひとまとめにする働きをする。

$\text{coeff2}(f, x, y)$; によって (f : 多変数の複雑な多項式)

$$f = (c_0 + c_1 x + c_2 y + c_3 x^2 + \dots) / c_{10}$$

のように f の表現が簡約化される。

f を原式の形に表現するときは、 $\text{eval2}(f)$; を使う。

このパッケージ coeff2 を使って、差分化された 2 次元戸田格子の解のカソラチ行列式による表現が確かめられた。

差分化された 2 次元戸田格子の解を $\tau(\ell, m, n)$ とする。2 次形式で表した 2 次元戸田方程式は

$$\begin{aligned} & [\Delta_{+1} \Delta_{-m} \tau(\ell, m, n)] \tau(\ell, m, n) \\ & - [\Delta_{+1} \tau(\ell, m, n)] [\Delta_{-m} \tau(\ell, m, n)] \end{aligned}$$

$$\begin{aligned}
& + \tau(\ell, m, n) \tau(\ell+1, m-1, n) \\
& - \tau(\ell+1, m, n-1) \tau(\ell, m-1, n+1) = 0 \quad (1)
\end{aligned}$$

である。ここで Δ_{+1} , Δ_{-m} はそれぞれ前進および、後退差分演算子を表す。

$\tau(\ell, m, n)$ を $N \times N$ のカソラチ行列式で表現する。

$$\tau(\ell, m, n) = \begin{vmatrix} f_1(\ell, m, n) & f_1(\ell, m, n+1) & \cdots & f_1(\ell, m, n+N-1) \\ f_2(\ell, m, n) & f_2(\ell, m, n+1) & \cdots & f_2(\ell, m, n+N-1) \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ f_N(\ell, m, n) & f_N(\ell, m, n+1) & \cdots & f_N(\ell, m, n+N-1) \end{vmatrix}$$

もし $f_i(\ell, m, n)$ ($i=1, 2, \dots, N$) が次の差分方程式

$$\Delta_{+1} f_i(\ell, m, n) = f_i(\ell, m, n+1),$$

$$\Delta_{-m} f_i(\ell, m, n) = -f_i(\ell, m, n-1)$$

を満たせば $\tau(\ell, m, n)$ は (1) 式の解になることが予想される。この予想を確かめるために $N=1, 2, 3, 4$ のときに計算を実行すると (1) 式は $N=4$ のときに爆発して計算が不可能になる。coeff2 を使って式の爆発を押さえ、 $N=4$ まで予想は確かめられた。

式 (1) を計算する Reduce Program を図 1 に示す。

coeff2 のソース Program を図 2 に示す。

REDUCE

SLISP/M200H AT HIROSHIMA UNIV. (86/11/10 11:53:15) : 925680 BYTE

REDUCE 3.2, 15-FEB-86 ...

1: OPERATOR F,H;

2: TODA:=(A-B)*H(L,M,N)*H(L+1,M-1,N)-A*H(L,M-1,N)*H(L+1,M,N)

2: +B*H(L+1,M,N-1)*H(L,M-1,N+1)\$

3: FORALL L,M,N,S LET

3: F(L+1,M,N,S)=F(L,M,N,S)+(1/A)*F(L,M,N+1,S);

3: F(L,M-1,N,S)=F(L,M,N,S)+B*F(L,M,N-1,S);

4: PROCEDURE TAU(NN); BEGIN SCALAR A; MATRIX MM(NN,NN);

4: FOR I:=1:NN DO FOR J:=1:NN DO MM(I,J):=F(L,M,N+J-1,I);

4: A:=DET(MM)\$ CLEAR MM; RETURN A END;

TAU

5: FORALL X,Y,Z,NN LET H(X,Y,Z,NN)=SUB(L=X,M=Y,N=Z,TAU(NN));

6: LOAD"FAPLIB:CDEFF2";

7: FORALL X,Y,Z LET H(X,Y,Z)=H(X,Y,Z,1);

8: TODA;

. 0

9: FORALL X,Y,Z LET H(X,Y,Z)=H(X,Y,Z,2);

10: TODA;

1

```

0
11: FORALL X,Y,Z LET H(X,Y,Z)=H(X,Y,Z,3);

12: TODA;

0
13: FORALL X,Y,Z LET H(X,Y,Z)=H(X,Y,Z,4);

14: TODA;

*** GC-STORAGE EXHAUSTED
* ((B . 2) . -1)
15: FORALL X,Y,Z LET H(X,Y,Z)=CDEFF2(H(X,Y,Z,4),A,B);

16: CDEFF(TODA,A,00)$

*** 005 004 003 002 001 000 ARE NON ZERO
17: EVAL2(005); EVAL2(004); EVAL2(003); EVAL2(002); EVAL2(001); EVAL2

0

0

0

0

0

0

0

23: BYE;

*** END OF RUN
READY

```

图 1 统 3

lisp.src:coeff2

1

Wed Dec 10 16:22:45 1986

```
#####;
%      COEFF2
%      Replace coefficients of expression by dummy
%
%      BY F.KAKO
%      August 7, 1986
%
% for example:
%
%      f := coeff2((x+y)**2,x,y);
%
%      
$$F := X^2 * c1 + X*Y*c2 + Y^2 * c3$$

%
%      eval2 f;
%
%      
$$X^2 + 2*X*Y + Y^2$$

%
%#####;
%;
```

```
SYMBOLIC PROCEDURE SIMP!-COEFF2 U;
  BEGIN SCALAR X,V,KLST;
    IF NULL U THEN RETURN NIL ./ 1;
    V := SIMP!* CAR U;
    IF NULL CDR U THEN RETURN V;
    KLST := FOREACH Y IN CDR U COLLECT !*A2K Y;
    X := SETKORDER KLST;
    V := COEFF2!-F(REORDER NUMR V, KLST) ./
        COEFF2!-F(REORDER DENR V, KLST);
    SETKORDER X;
    RETURN REORDER NUMR V ./ REORDER DENR V
  END;
```

```
SYMBOLIC PROCEDURE COEFF2!-F(U,KLST);
  IF DOMAINP U THEN U
  ELSE IF NOT(MVAR U MEMQ KLST) THEN
    BEGIN SCALAR X;
      X := MKID2();
      PUT(X,'MYVALUE,U);
      RETURN LIST((X .** 1) .1)
    END
  ELSE ADDF(MULTF(COEFF2!-F(LC U, KLST), !*P2F LPOW U),
    COEFF2!-F(RED U,KLST));
```

四 2

lisp.src:coeff2

2

```
PUT('COEFF2,'SIMPFN,'SIMP!-COEFF2);
```

```
SYMBOLIC PROCEDURE EVAL2 U;  
  MK!*SQ EVAL2!-SQ SIMP!* U;
```

```
FLAG('EVAL2),'OPFN);  
FLAG('EVAL2),'NOVAL);
```

```
GLOBAL '(SYM COUNT);  
LISP SYM COUNT := 0;
```

```
SYMBOLIC PROCEDURE MKID2();  
  <<SYM COUNT := SYM COUNT + 1;  
  COMPRESS NCONC(EXPLODE '!c, EXPLODE SYM COUNT)>>;
```

```
SYMBOLIC PROCEDURE EVAL2!-SQ U;  
  CANCEL(EVAL2!-F NUMR U ./ EVAL2!-F DENR U);
```

```
SYMBOLIC PROCEDURE EVAL2!-F U;  
  IF DOMAINP U THEN U  
  ELSE ADDF(MULTF(EVAL2!-P LPOW U, EVAL2!-F LC U),  
    EVAL2!-F RED U);
```

```
SYMBOLIC PROCEDURE EVAL2!-P U;  
  BEGIN SCALAR V,W,N;  
    N := CDR U;          %INTEGER POWER;  
    V := CAR U;          %MAIN VARIABLE;  
    IF ATOM V THEN  
      IF (W := GET(V,'MYVALUE)) THEN  
        RETURN EXPTF(REORDER W,N);  
    RETURN !*P2F U  
  END;
```

```
END;
```