数理解析研究所講究録
第486巻 1983年 195-211

15. 話題提供
―数式処理システムをめぐって―

東京大学大型計算機センター 金田 康正
(Yasumasa Kanada)

1. はじめに

最近、英国ケンブリッジ大学(以後ケンブリッジ大学と略す。)で作成された、積分パッケージ及び因数分解パッケージを使用する機会があった。その際の使用経験にもとずいた話題提供である。まず始めに、

1) 大学学期末試験問題による積分パッケージの機能評価

について述べ、次いで

2) 二次の線型微分方程式の変形理論への因数分解パッケージの利用と、パッケージの実用性について

について述べる。

今回使用したソフトウェアは、以下に示す性格を持ったものである。使いがって手を向上させた後、東京大学大型計算機センターを通じて公開されるので、大いに利用いただきたい。

a) IBM 360/370用のBCPL(一部改良を行なってある)と少

しのアセンブラーで記述してあるLISP処理系によって駆動される[1] REDUCE 2数式処理システム[2]。(現在、A.C. Hearnが配布している版と比較してみると少し古い。しかし特殊な事を行なわないならば、使用上何の問題もない。)従って、他機種への書き替えは困難である。

b)積分パッケージで使用しているアルゴリズムの説明は文献3に示してある。

c)因数分解パッケージは、A.C. Normanが作成したものであるが、アルゴリズムについての解説は今の所公にされていないようである。

d)積分パッケージと因数分解パッケージとを同時に使用するのは不可能である。

e) 1980年10月に、今回使用したソフトウェア全体を入手した。

f)両パッケージとも、500Kバイト程度のメモリーでも動作するが、ガーベージコレクション、I/O等のオーバーヘッドが多くなり、実用的見地からはあまりすすめられない。

2. 積分パッケージの機能評価

計算機による積分プログラムとして有名なSAINT[4]は、MIT

のJ. Slagleによって1960年頃に作成された。このプログラムはその当時に作られたばかりのLISP言語で書かれ、IBM 7090計算機の上で動作していた。プログラムの占める大きさは、3kwの作業用領域を含め10kwであった。その実力は、当時のMITにおける54題のfreshman calculus examinationを含む全86題の問題のうち84題を解いたことからおしはかれよう。平均解答時間は、IBM 7090を用いて2.4分だった。一番時間がかからなかったのは、

$$\int \frac{1}{x} dx$$

の計算で0.03分、一番時間がかかったのは、

$$\int \frac{\sec^2 t}{1-\sec^2 t - 3\tan t} dt$$

の18分であった。

Slagleは、発見的方法(試行錯誤)による解法でプログラムしたが、その後1960年代後半から1960年代前半にかけ、積分アルゴリズムは大きく進歩した。この新しいアルゴリズムによる解法をインプリメントしたものが今回評価したパッケージである。評価試験用の問題として採用したのは、ある工科系私立大学の教養部で単位認定の為に用いられていた試験問題78題である。(尚これを全部解くには、人手－現役の学生－で約1時間かかる。)

試験結果から言うと、75題が何らかの解を出した。解が出

なかった3題は、未定義関数(DESPARSE)を呼んでいた為である。平均計算時間は、HITAC M-200Hで126ms. 一番時間がかかったのは、

$$\int \frac{1}{2}\left(\cos\frac{9}{8}x + \cos\frac{3}{8}x\right)dx$$

の474ms. 一番時間がかからなかったのは、

$$\int \frac{1}{9+x^2}\,dx$$

の26msであった。付録1にいくつかの計算例を示しておくので参考にされたい。

予想されていたことであるが、パッケージの答と人手による答は全てが合致していなかった。人手の答が違っていたのもあれば、入力ミスで別の計算をしていたのもあった。人手でも機械でも結果は正しいが、表現が違っているものを以下にいくつか選び出すことで、積分を行なう場合の問題を明らかにしてみよう。

◎
- 問題　$\int (\cos x)(\sin 2x)\,dx$
- 人手解　$-\frac{2}{3}\cos^3 x + C$
- 機械解　$\frac{1}{3}(-2\cdot\cos 2x\cdot\cos x - \sin 2x\cdot\sin x) + C$

人間は、問題を $-2\int \cos^2 x\,(\cos x)'\,dx$ と見なして計算したが、プログラムは三角関数の積を和の形に変形し、その後部分積分している。

◎
- 問題　$\int \frac{1}{\sin x}\,dx$
- 人手解　$-\frac{1}{2}\log\left|\frac{1+\cos x}{1-\cos x}\right|+c$
- 機械解　$\log\left(\tan\frac{x}{2}\right)+c$

人間は、$\int \frac{\sin x}{\sin^2 x}\,dx = -\int \frac{(\cos x)'}{1-\cos^2 x}\,dx$

$$= -\int\left(\frac{1}{1-\cos x}+\frac{1}{1+\cos x}\right)\frac{1}{2}(\cos x)'\,dx$$

と変形していった。両方とも正しいが、どちらが望ましいかは、結果をどのように使うかに依存する。（機械解は絶対値をとっていない事に注意。）

◎
- 問題　$\int \frac{dx}{\sqrt{x^2+4}}$
- 人手解　$\log\left|x+\sqrt{x^2+4}\right|+c$
- 機械解　$\frac{1}{2}\log\{-\log(\sqrt{x^2+4}-x)+\log(\sqrt{x^2+4}+x)\}+c$

人間は公式を用いて結果を出している。機械解をもう少し変形すれば人手解と同じになるが，どこまで簡約化すれば良いのか、判断を自動的に行なうのは困難。

◎
- 問題　$\int (1-e^x)^4 e^x\,dx$
- 人手解　$-\frac{1}{5}(1-e^x)^5+c$
- 機械解　$\frac{1}{5}\{e^x(e^{4x}-5e^{3x}+10e^{2x}-10e^x+5)\}+c$

人間は展開せずに直接に解を出した。機械は展開後部分積分を行なっているが，こうなると、因数分解ルーチンがあっても $\frac{1}{5}(1-e^x)^5$ の形にもどすのは困難である。

3. 二次の線型微分方程式の変形理論への因数分解パッケージの利用

今回問題になった変形理論は文献5, 6にあるので参考にされたい。結論から言えばある理論の証明を行なう為に、2変数x, yの2つの有理関数

$$\begin{cases} u = \varphi(x, y) \\ v = \psi(x, y) \end{cases}$$

に対し、Jacobi行列式が恒等的に1であること、すなわち

$$\begin{vmatrix} \varphi_x & \varphi_y \\ \psi_x & \psi_y \end{vmatrix} \equiv 1$$

を示したいのである。ここでφ, ψは5個のパラメータt, a, b, c, dを含み、φは次のように書かれる：

$$\varphi(x, y) = \frac{U(x, y)}{V(x, y)} 。$$

ただし

$$\begin{aligned} U(x, y) = &-dA^2((a+b+c)x^2-(a+b+c+d)(t+1)x+(b+c+d)t) \\ &+AB((a+b+c-2d)x-((a+b)t+a+c)) \\ &-AC(\tfrac{1}{2}(2x-t-1)y-(a+b+c+2d)) \\ &+B^2, \end{aligned}$$

$$\begin{aligned} V(x, y) = &A(dA((a+b+c+3d)x-((c+d)t+(b+d))) \\ &-(a+b+c+2d)B \end{aligned}$$

$$+\frac{1}{2}C\ y),$$

A, B, C は

$$A=(x^3-(t+1)x^2+tx)y-(a+b+c)x^2+((a+b)t+a+c)x-at.$$

$$B=\frac{1}{4}(x^3-(t+1)x^2+tx)(3x^2-2(t+1)x+t)y^2$$
$$-\frac{1}{2}(5(a+b+c)x^4-4((2a+2b+c)t+2a+b+2c)x^3$$
$$+3((a+b)t^2+(4a+2b+2c)t+a+c)x^2$$
$$-2t((2a+b)t+2a+c)x+at^2)y$$
$$+((a+b+c)x^2-((a+b)t+a+c)x+at)\times$$
$$(2(a+b+c)x-((a+b)t+a+c))$$
$$+(a+b+c+d)d(x^3-(t+1)x^2+tx).$$

$$C=\frac{1}{2}(x^3-(t+1)x^2+tx)^2y^2$$
$$-\frac{3}{2}(x^3-(t+1)x^2+tx)((a+b+c)x^2-((a+b)t+a+c)x+at)y$$
$$+((a+b+c)x^2-((a+b)t+a+c)x+at)^2.$$

$\psi(x,y)$ は、

$$\psi(x,y)=-\frac{dA(u-x)^2+B(u-x)+C}{A\,u(u-1)(u-t)}.$$

ここで

$$u=\varphi(x,y).$$

説明すべきことは簡単である。7変数の式を微分したり、積をとったりするわけで、メモリーが問題になりそうであることが予想された。まずはためしにと簡単なプログラムで試してみたところ(これが大事。こうでないか、ああじゃない

かという心配はやってみてからすればよい。まさに「案ずるより産むが易し」である。)やはりパンクしてしまった。「メモリーはケチケチ精神」のモットーで、①なるべく式を展開しないように、②二度以上の繰り返しの計算をさける為にまず変数に置きかえ整理した後、変数に値を代入する、③式のくくりだしが可能ならば、できるだけくくりだしを行なう、といった手法を用い、メモリーパンクをおさえながらやっと、

$$\mathrm{VAL}=\left(\begin{vmatrix}4x & 4y\\ 4x & 4y\end{vmatrix}-1\right)*\left(\begin{vmatrix}4x & 4y\\ 4x & 4y\end{vmatrix}\text{の分母}\right)$$

の式が展開できるようになった。あとはこれが恒等的にゼロであることを示せばよいのである。—しかしながら常にうまくいくとは限らない。

① a, b, c, d, t, x, y,に適当な数値をいれて、VALを数値計算し、それが0であることを示す方法。(n次多項式が$n+1$点で0の値をもてば、その多項式は恒等的に0であるということを利用する。)⟶もし0にならなければ、恒等式が成立しないことを示すので、理論がおかしいことになる。いくつかの場合で0になれば、恒等式が成立していそうだということを保障してくれる。今の場合、変数の次数が高く、$a, b, \cdots$に$2, 3, \cdots$といった値を代入しても多倍長整数用にメモリーをとられてしまう。また計算に

時間がかかってしまうので、この方法の使用はあきらめなければならなかった。

②①における数値の多倍長化をさけるために数値を適当な modulus で基本整数におさえ込んでやる方法がある。この場合、VALの次数が高く、計算時間がかかりそうなのでこれも採用しなかった。(これは場合によっては有効な方法となる。)

さてそこで登場するのが因数分解パッケージである。ためしに因数分解をやってみたところ、VALは、

$$U(x,y),\ V(x,y)^2,\ V(x,y)-U(x,y)$$

$$T*V(x,y)-U(x,y),\ A(x,y)^2$$

のそれぞれで割り切れることが判明した。(本当に因子かどうかは、割り切れるかどうかやってみれば良い。)この時の商をVAL1としたところ、VAL1はA(x,y)に関して6次の式になり、

VAL1のA(x,y)の	0	次の項は B(x,y)の	5	次で	割り切れる
〃	1	〃	4	〃	
〃	2	〃	3	〃	
〃	3	〃	2	〃	
〃	4	〃	1	〃	

ことが、因数分解の結果判明した。しかしここまできても、項の数は激減しているものの、正攻法ではいっきに展開できない。(いかに大きな多項式か！) VAL1をいろいろ会話的

にいじってみたところ、付録2の後半部分に示す性質があることが判明した。以上をプログラムし、バッチジョブで一括処理したところ、めでたくVAL1が0になった。HLISP－REDUCEでもチェックし、正しく0になった。これで、めでたしめでたし。証明が完了！計算機による計算は、木村先生の未完の手計算を完全にフォローしていることが、その後判明し、その面からも理論は正しいことが証明された。

4. おわりに

以上の経験により次のことが言えよう。

◎得られた式の表現上の問題があるので、バカチョン式に積分パッケージを使うことには、問題がある。

◎因数分解は、式変形の強力な助っ人である。

◎少し手をぬくと、メモリーに関する制限で展開はうまくゆかなくなる。

◎会話で見通しを立て、長時間実行はバッチで行なう方法は効果的である。

◎会話処理において、REDUCEから抜けだすことなく、割り込みをかけられることは、会話において是非とも必要な機能である。

◎現状では、処理速度、利用可能メモリーの点で、最先端

の研究に利用するには，数式処理の力不足を感じる。

文献

1) J. P. Fitch and A. C. Norman : "Implementing LISP in a High-level Language", Soft. - Practice and Exp. Vol. 7 (1977) 713～725

2) A. C. Hearn : "REDUCE 2 User's Manual", UCP－19, Computational Physics Group, University of Utah, Salt Lake City, Utah (1973 March)

3) J. H. Davenport : "On the Integration of Algebraic Functions". Lecture Notes in Computer Science. 102. Springer－Verlag. 1981

4) J. R. Slagle : "A heuristic program that solves symbolic integration problem in freshman calculus", J. ACM 10, 4 (Oct. 1963) pp. 507～520

5) T. Kimura : "On the Isomonodromic Deformation for Linear Ordinary Differential Equation of the Second Order" Proc. Japan Academy, 57 Ser. A, 6 (1981) pp. 285～290

6) T. Kimura : "On the Isomonodromic Deformation for Linear Ordinary Differential Equations of the Second

Order. II" Proc. Japan Academy, 58 Ser. A, 7(1982) pp. 294~297

付録1. 積分パッケージの使用例

```
INT(COS(T)*SIN(2*T),T);
POTENTIAL CANCELLATION DETECTED
POTENTIAL CANCELLATION DETECTED
POTENTIAL CANCELLATION DETECTED
(TIME TAKEN 265 MILLISECONDS)

( - 2*COS(2*T)*COS(T) - SIN(2*T)*SIN(T) + 2)/3

Time increment = 0.28+0.51 seconds. Total = 2.21+4.65
```

← 意味不明

ユーザーサービス時間、システム時間(GBC, I/O, プログラムロード, etc.)

総 〃 総 〃

```
INT(((SIN X**3)*(COS X**2)),X);
POTENTIAL CANCELLATION DETECTED
(TIME TAKEN 335 MILLISECONDS)

                   4                      2
(3*COS(X)*SIN(X)    - COS(X)*SIN(X)    - 2*COS(X) - 2)/15

Time increment = 0.35+0.11 seconds. Total = 6.03+6.08

INT((COS X / SIN X**2),X);
POTENTIAL CANCELLATION DETECTED
(TIME TAKEN 123 MILLISECONDS)

( - 1)/SIN(X)

Time increment = 0.13+0.11 seconds. Total = 6.17+6.19

INT((1/(4-X**2)**(1/2)),X);
(TIME TAKEN 176 MILLISECONDS)

                   2
(i*(LOG(SQRT( - X   + 4) - i*X) - LOG(lastsqrt + i*X)))/2

Time increment = 0.12+0.20 seconds. Total = 6.87+6.61

INT((1/(X**2-4)**(1/2)),X);
(TIME TAKEN 190 MILLISECONDS)

                 2
( - LOG(SQRT(X    - 4) - X) + LOG(lastsqrt + X))/2

Time increment = 0.20+0.15 seconds. Total = 8.22+8.10
```

```
INT((1/(2*X**2-4)**(1/2)),X);
THE FOLLOWING QUADRATIC DOES NOT SEEM TO FACTOR

   2
( A )  - 2

(TIME TAKEN 355 MILLISECONDS)

                          2
(SQRT(2)*( - LOG(SQRT(X    - 2) - X) + LOG(lastsqrt + X)))/4

Time increment = 0.37+0.48 seconds. Total = 10.35+10.52

INT((1/(X**2-9)),X);
(TIME TAKEN 32 MILLISECONDS)

(LOG(X - 3) - LOG(X + 3))/6

Time increment = 0.04+0.12 seconds. Total = 0.98+1.29

INT((1/((-3)-2*X**2)),X);
(TIME TAKEN 45 MILLISECONDS)

( - ATAN((SQRT(2)*SQRT(3)*X)/3)*SQRT(2)*SQRT(3))/6

Time increment = 0.06+0.12 seconds. Total = 1.40+2.29

INT((1/(9+(2-X)**2)),X);
(TIME TAKEN 32 MILLISECONDS)

ATAN((X - 2)/3)/3

Time increment = 0.04+0.12 seconds. Total = 1.79+2.77

INT((1/(6*X-9*X**2-4)),X);
(TIME TAKEN 42 MILLISECONDS)

( - ATAN((3*SQRT(3)*X - lastsqrt)/3)*lastsqrt)/9

Time increment = 0.05+0.12 seconds. Total = 1.94+3.01

INT(X**3*LOG(X),X);
POTENTIAL CANCELLATION DETECTED
(TIME TAKEN 36 MILLISECONDS)

  4
(X *(4*LOG(X) - 1))/16

Time increment = 0.05+0.11 seconds. Total = 0.18+0.78

INT((LOG X)**2,X);
POTENTIAL CANCELLATION DETECTED
(TIME TAKEN 37 MILLISECONDS)

          2
X*(LOG(X)   - 2*LOG(X) + 2)

Time increment = 0.05+0.11 seconds. Total = 0.33+1.12

INT(X**3*E**X,X);
POTENTIAL CANCELLATION DETECTED
(TIME TAKEN 40 MILLISECONDS)

 X   3      2
E *(X  - 3*X  + 6*X - 6)

Time increment = 0.05+0.10 seconds. Total = 0.57+1.67
```

```
INT(E**X/(1-E**(2*X)),X);
POTENTIAL CANCELLATION DETECTED
(TIME TAKEN 66 MILLISECONDS)

            X             X
( - LOG(E    - 1) + LOG(E   + 1))/2

Time increment = 0.08+0.12 seconds. Total = 0.92+2.23

INT(1/((X-3)*(X+5)),X);
(TIME TAKEN 34 MILLISECONDS)

(LOG(X - 3) - LOG(X + 5))/8

Time increment = 0.05+0.13 seconds. Total = 1.18+2.64

INT(1/((1+2*X)*(3+4*X)),X);
(TIME TAKEN 45 MILLISECONDS)

(LOG(8*X + 4) - LOG(8*X + 6))/2

Time increment = 0.06+0.11 seconds. Total = 1.50+3.29

INT(1/((4-X**2)*(4+X**2))(X);
(TIME TAKEN 65 MILLISECONDS)

(2*ATAN(X/2) - LOG(X - 2) + LOG(X + 2))/32

Time increment = 0.08+0.18 seconds. Total = 1.70+3.72

INT((1/((1-X)*(1+X**2))),X);
(TIME TAKEN 56 MILLISECONDS)

                       2
(2*ATAN(X) + LOG(X   + 1) - 2*LOG(X - 1))/4

Time increment = 0.07+0.17 seconds. Total = 1.93+4.14
```

付録2.

理論式の証明に使用したREDUCEプログラムリストの例

```
OPERATOR PHAI,PSAI,U,V,AA,BB,CC,AC,B1,B2$
OFF EXP$
PHAI(X,Y):=U(X,Y)/V(X,Y)$
CC(X,Y):=AA(X,Y)*AC(X,Y)$
PSAI(X,Y) := -(D*AA(X,Y)*(PHAI(X,Y)-X)**2+BB(X,Y)*(PHAI(X,Y)-X)+CC(X,Y))
                /(AA(X,Y)*PHAI(X,Y)*(PHAI(X,Y)-1)*(PHAI(X,Y)-T))$
VAL := DF(PHAI(X,Y),X)*DF(PSAI(X,Y),Y)-DF(PHAI(X,Y),Y)*DF(PSAI(X,Y),X)$
VAL := NUM(VAL) - (VALDEN:=DEN(VAL))$
VAL := SUB(DF(U(X,Y),X)=WDFUX,DF(U(X,Y),Y)=WDFUY,DF(V(X,Y),X)=WDFVX,
           DF(V(X,Y),Y)=WDFVY,
           AA(X,Y)=WAA,BB(X,Y)=WBB,AC(X,Y)=WAC,
           U(X,Y)=WU,V(X,Y)=WV,
           DF(AA(X,Y),X)=WDFAAX,DF(AA(X,Y),Y)=WDFAAY,
           DF(BB(X,Y),X)=WDFBBX,DF(BB(X,Y),Y)=WDFBBY,
           DF(AC(X,Y),X)=WDFACX,DF(AC(X,Y),Y)=WDFACY,VAL)$
```

```
VALDEN := SUB(DF(U(X,Y),X)=WDFUX,DF(U(X,Y),Y)=WDFUY,DF(V(X,Y),X)=WDFVX,
           DF(V(X,Y),Y)=WDFVY,
           AA(X,Y)=WAA,BB(X,Y)=WBB,AC(X,Y)=WAC,
           U(X,Y)=WU,V(X,Y)=WV,
           DF(AA(X,Y),X)=WDFAAX,DF(AA(X,Y),Y)=WDFAAY,
           DF(BB(X,Y),X)=WDFBBX,DF(BB(X,Y),Y)=WDFBBY,
           DF(AC(X,Y),X)=WDFACX,DF(AC(X,Y),Y)=WDFACY,VALDEN)$
ON  EXP$
U(X,Y):=-D*AA(X,Y)**2*((A+B+C)*X**2-(A+B+C+D)*(T+1)*X+(B+C+D)*T)
        +  AA(X,Y)*BB(X,Y)* ((A+B+C-2*D)*X-((A+B)*T+A+C))
        -  AA(X,Y)*CC(X,Y)* ((1/2)*(2*X-(T+1))*Y-(A+B+C+2*D))
        + BB(X,Y)**2$
V(X,Y):=AA(X,Y)*(D*AA(X,Y)*((A+B+C+3*D)*X-((C+D)*T+B+D))
           -(A+B+C+2*D)*BB(X,Y)
           +(1/2)*CC(X,Y)*Y)$
ON EXP;
DFUX := DF(U(X,Y),X)$
DFUX:= SUB(DF(AA(X,Y),X)=WDFAAX,DF(AA(X,Y),Y)=WDFAAY,
           DF(BB(X,Y),X)=WDFBBX,DF(BB(X,Y),Y)=WDFBBY,
           DF(AC(X,Y),X)=WDFACX,DF(AC(X,Y),Y)=WDFACY,
           AA(X,Y)=WAA,BB(X,Y)=WBB,AC(X,Y)=WAC,DFUX)$
DFUY := DF(U(X,Y),Y)$
DFUY:= SUB(DF(AA(X,Y),X)=WDFAAX,DF(AA(X,Y),Y)=WDFAAY,
           DF(BB(X,Y),X)=WDFBBX,DF(BB(X,Y),Y)=WDFBBY,
           DF(AC(X,Y),X)=WDFACX,DF(AC(X,Y),Y)=WDFACY,
           AA(X,Y)=WAA,BB(X,Y)=WBB,AC(X,Y)=WAC,DFUY)$
DFVX := DF(V(X,Y),X)$
DFVX:= SUB(DF(AA(X,Y),X)=WDFAAX,DF(AA(X,Y),Y)=WDFAAY,
           DF(BB(X,Y),X)=WDFBBX,DF(BB(X,Y),Y)=WDFBBY,
           DF(AC(X,Y),X)=WDFACX,DF(AC(X,Y),Y)=WDFACY,
           AA(X,Y)=WAA,BB(X,Y)=WBB,AC(X,Y)=WAC,DFVX)$
DFVY := DF(V(X,Y),Y)$
DFVY:= SUB(DF(AA(X,Y),X)=WDFAAX,DF(AA(X,Y),Y)=WDFAAY,
           DF(BB(X,Y),X)=WDFBBX,DF(BB(X,Y),Y)=WDFBBY,
           DF(AC(X,Y),X)=WDFACX,DF(AC(X,Y),Y)=WDFACY,
           AA(X,Y)=WAA,BB(X,Y)=WBB,AC(X,Y)=WAC,DFVY)$
U(X,Y)  := SUB(AA(X,Y)=WAA,BB(X,Y)=WBB,AC(X,Y)=WAC,U(X,Y))$
V(X,Y)  := SUB(AA(X,Y)=WAA,BB(X,Y)=WBB,AC(X,Y)=WAC,V(X,Y))$
AA(X,Y):=B1(X)*Y - B2(X)$
BB(X,Y):=(1/4)*B1(X)*(3*X**2 - 2*(T+1)*X + T)*Y**2
  -(1/2)*(5*(A+B+C)*X**4 - 4*((2*A+2*B+C)*T+2*A+B+2*C)*X**3
  +3*((A+B)*T**2+(4*A+2*B+2*C)*T+A+C)*X**2-2*T*((2*A+B)*T+2*A+C)*X+A*T**2)*Y
  +       B2(X)*(2*(A+B+C)*X - ((A+B)*T+A+C))
  +       (A+B+C+D)*D*B1(X)$
AC(X,Y):=(B1(X)*Y-2*B2(X))/2$
B1(X)  := X**3 - (T+1)*X**2 + T*X$
B2(X)  := (A+B+C)*X**2 - ((A+B)*T+A+C)*X +A*T$
ON EXP;
B1(X)  := B1(X)$
B2(X)  := B2(X)$
AA(X,Y)  := AA(X,Y)$
BB(X,Y)  := BB(X,Y)$
AC(X,Y)  := AC(X,Y)$
DFUX  := DFUX$
DFUY  := DFUY$
DFVX  := DFVX$
DFVY  := DFVY$
DFAAX  := DF(AA(X,Y),X)$
DFAAY  := DF(AA(X,Y),Y)$
DFBBX  := DF(BB(X,Y),X)$
DFBBY  := DF(BB(X,Y),Y)$
DFACX  := DF(AC(X,Y),X)$
DFACY  := DF(AC(X,Y),Y)$
%OFF EXP;
ON EXP;
U(X,Y)  := U(X,Y)$
V(X,Y)  := V(X,Y)$
```

```
VAL := VAL/(WU*WV**2)/(WV-WU)/(T*WV-WU);

LET WV*X = WU - WUX,
    WV*WDFUX = WU*WDFVX + WUWDFUX,
    WV*WDFUY = WU*WDFVY + WUWDFUY,
    WAA*WDFCCX = WCC*WDFAAX + WAAWDFCCX,
    WAA*WDFCCY = WCC*WDFAAY + WAAWDFCCY,
    WU**3 = WUV*WU**2 + WV*WU**2,
    WU**2*WV = WUV*WU*WV + WU*WV**2;
VAL1:=VAL;
CLEAR WV*X,WV*WDFUX,WV*WDFUY,WAA*WDFCCX,WAA*WDFCCY,WU**3,WU**2*WV;
CLEAR VAL;
ON EXP;
WU := U(X,Y);
WV := V(X,Y);
WDFUX := DFUX;
WDFVX := DFVX;
WDFUY := DFUY;
WDFVY := DFVY;
WUX := WU - WV*X;
WUWDFUX := WV*WDFUX - WU*WDFVX;
WUWDFUY := WV*WDFUY - WU*WDFVY;
WUV := WU - WV;
WDFAAY := 2*WDFACY;
ON GCD;
WUX := WUX$
WUWDFUX := WUWDFUX$
WUWDFUY := WUWDFUY$
WUV := WUV$
DFBBY := DFBBY$
OFF GCD;
VAL1 := VAL1/WAA**2$
ARRAY WORK(6);
K := COEFF(VAL1, WAA, WORK);
VAL1 := 0$
ON GCD;
WORK(0):=WORK(0)$
WORK(1):=WORK(1)$
WORK(2):=WORK(2)$
WORK(3):=WORK(3)$
WORK(4):=WORK(4)$
WORK(5):=WORK(5)$
WORK(6):=WORK(6)$
OFF GCD;
ARRAY WORK1(6),WORK2(10),WORK3(10),WORK4(10),WORK5(10),WORK6(10),WORK7(10),
      WORK8(10);

ALGEBRAIC PROCEDURE EVALUATE(ARG);
   BEGIN SCALAR S0,S1,S2,S3,S4,S5,S6,K0,K1,K2,K3,K4,K5,K6;
      K0 := COEFF(ARG, WBB, WORK1);
      S0 := 0;
      FOR J0 := K0 STEP -1 UNTIL 0 DO BEGIN;
      K1 := COEFF(WORK1(J0), WAC, WORK2);
      S1 := 0;
      FOR J1 := K1 STEP -1 UNTIL 0 DO BEGIN;
         S2 := 0;
         K2 := COEFF(WORK2(J1), WDFAAX, WORK3);
         WORK2(J1) := 0;
         FOR J2 := K2 STEP -1 UNTIL 0 DO BEGIN;
            S3 := 0;
            K3 := COEFF(WORK3(J2), WDFBBX, WORK4);
            WORK3(J2) := 0;
            FOR J3 := K3 STEP -1 UNTIL 0 DO BEGIN;
               S4 := 0;
               K4 := COEFF(WORK4(J3), WDFACX, WORK5);
               WORK4(J3) := 0;
               FOR J4 := K4 STEP -1 UNTIL 0 DO BEGIN;
```

```
                    S5 := 0;
                    K5 := COEFF(WORK5(J4), WDFBBY, WORK6);
                    WORK5(J4) := 0;
                    FOR J5 := K5 STEP -1 UNTIL 0 DO BEGIN;
                       S6 := 0;
                       K6 := COEFF(WORK6(J5), WDFACY, WORK7);
                       WORK6(J5) := 0;
                       WRITE                        "J1 = ",J1,",J2 = ",J2,
                              ",J3 = ",J3,",J4 = ",J4,",J5 = ",J5,",K6 = ",K6;
                       FOR J6 := K6 STEP -1 UNTIL 0 DO BEGIN;
                          S6 := S6 * DFACY + WORK7(J6);
                          WORK7(J6) := 0;
                          END;
                       S5 := S5 * DFBBY + S6;
                       END;
                    S4 := S4 * DFACX + S5;
                    END;
                 S3 := S3 * DFBBX + S4;
                 END;
              S2 := S2 * DFAAX + S3;
              END;
           S1 := S1 * AC(X,Y) + S2;
           END;
           S0 := S0 * BB(X,Y) + S1;
           END;
        RETURN S0
     END;
WORK(0):=EVALUATE(WORK(0)/WBB**5)/AA(X,Y)$
WORK(1):=(EVALUATE(WORK(1)/WBB**4)+WORK(0)*BB(X,Y))/AA(X,Y)$
WORK(0):=0$
WORK(2):=(EVALUATE(WORK(2)/WBB**3)+WORK(1)*BB(X,Y))/AA(X,Y)$
WORK(1):=0$
WORK(3):=(EVALUATE(WORK(3)/WBB**2)+WORK(2)*BB(X,Y))/AA(X,Y)$
WORK(2):=0$
WORK(4):=(EVALUATE(WORK(4)/WBB)+WORK(3)*BB(X,Y))/AA(X,Y)$
WORK(3):=0$
WORK(5):=(EVALUATE(WORK(5))+WORK(4)*BB(X,Y))/AA(X,Y)$
WORK(4):=0$
WORK(6):=EVALUATE(WORK(6))+WORK(5);
WORK(5):=0$
END;
```

一番最後の結果
めでたく 0 になっている。

```
WORK(6):=EVALUATE(WORK(6))+WORK(5);

J1 = 3,J2 = 0,J3 = 0,J4 = 1,J5 = 0,K6 = 0

J1 = 3,J2 = 0,J3 = 0,J4 = 0,J5 = 0,K6 = 1

J1 = 2,J2 = 0,J3 = 0,J4 = 1,J5 = 0,K6 = 0

J1 = 2,J2 = 0,J3 = 0,J4 = 0,J5 = 0,K6 = 1

J1 = 1,J2 = 0,J3 = 0,J4 = 1,J5 = 0,K6 = 0

J1 = 1,J2 = 0,J3 = 0,J4 = 0,J5 = 0,K6 = 1

J1 = 0,J2 = 0,J3 = 0,J4 = 0,J5 = 0,K6 = 1

WORK(6) := 0
```