

日本で使用できる数式処理システム

名大 プラズマ研究所 金田 康正

1. はじめに

「数式を記号のまま計算機で処理する事が可能なシステム」という意味での数式処理システムの研究は、欧米及び日本で精力的に行なわれており、その成果としての数式処理システムは、世界的に見ても、汎用なものから特殊分野向けのものまで数十は下らないと思われる。しかし、「数十人以上の利用者がいる」、という意味で、研究成果が役に立っているシステムは十指^{*}に満たないであろう。本稿では、本当に使いものになるシステムのうち、現在日本で使用可能なものを紹介すると共に、それらの簡単な説明を行ないたい。この解説が今後の研究のお役に立てば幸である。

* ALTRAN (ベル研), ASHMEDAI (米アルゴンヌ国立研究所), CAMAL (英ケンブリッジ大), FORMAC (IBM), MACSYMA (MIT), MATHLAB (MIT), μ -MATH⁷⁹ (ハワイ大), SAC-1 (米ウィスコンシン大), SCHOONSCHIP (CERN), 等。

2. 実用の道具としての数式処理

問題その1.

9次の判別式の項の数は26059であることを確かめよ。

問題その2.

$p^2 + q^2 = 1$ の束縛条件の下で

$$\varepsilon = \frac{p^2 x^4 + q^2 y^4 - 2ipqxy(x^2 - y^2) - 1}{2px(x^2 - 1) - 2iqy(1 - y^2)}$$

$\bar{\varepsilon}$ = 上記 ε の complex conjugate

とした時にこの ε は、

$$\begin{aligned} & (\varepsilon \bar{\varepsilon} - 1) \left\{ (x^2 - 1) \frac{\partial^2 \varepsilon}{\partial x^2} + (1 - y^2) \frac{\partial^2 \varepsilon}{\partial y^2} + 2 \left(x \frac{\partial \varepsilon}{\partial x} - y \frac{\partial \varepsilon}{\partial y} \right) \right\} \\ &= 2 \bar{\varepsilon} \left\{ (x^2 - 1) \left(\frac{\partial \varepsilon}{\partial x} \right)^2 + (1 - y^2) \left(\frac{\partial \varepsilon}{\partial y} \right)^2 \right\} \end{aligned}$$

の式を満足する事を確かめよ。

といった問題が出された時、どうすれば良いであろうか？
 まづは、自分がやるにしろ、他人にお願いするにしろ、手と紙を用いて計算してみることが頭に浮かぶことであろう。上記の問題については、実際にやってみると、時間量、計算量からして、あまり現実的な方策ではない事にすぐ気が付くはずである。しかも本気で問題を解くのであれば、2人以上の

* 実際の論文審査では、しばしばお目にかかる問題である。

人間による独立計算で、Cross check を行なわなければならない。カウスのような計算の天才、計算を楽（み）にしている人は別にして、連日連夜、計算に悩まされている人にとりては、そのような計算の楽（み）のものではあらずがよい。それはまた、何人かの人間を、何日か、その種の計算に従事させる事になり、結果としてある意味での人間のムダ使い、頭脳の無意味な消費をおこなうしかねない。

このような状況に陥、た時に思い出していただきたいのが、実用の道具としての数式処理である。「数値計算を行なう機械としての電子計算機」というのは、電子計算機が発見されたいきさつからして当然の事であるが、工学技術の進歩、研究成果の実現されたものとしての数式処理アルゴリズムの開発の結果、今では計算機を用いて、数式の、

- ① 降べき、昇べき順への並び換え、
- ② 共通項のくくり出し、
- ③ $x + 5x = 6x$, $x * 0 = 0$, 等の簡約化、
- ④ $\frac{d \sin x}{dx} = \cos x$, 等の微分、
- ⑤ 変数への数値あるいは別の数式の代入、
- ⑥ 多項式、有理関数の展開、
- ⑦ 多項式同志の GCD の計算、
- ⑧ 行列及び行列式の計算、

⑨ $\sum_{i=1}^n i = \frac{n(n+1)}{2}$, $\sum_{i=1}^{\infty} \frac{1}{i^2} = \zeta(2) = \frac{\pi^2}{6}$ 等の(無限)級数の求和

⑩ $\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$ 等の極限值計算

⑪ 多倍長整数、高精度浮動小数点演算機能を用いた多項式の数値計算解の導出、

⑫ 多変数多項式の因数分解、

⑬ 定積分、不定積分、

⑭ S_i , C_i , E_i , Erfc , ... 等の特殊関数の取り扱い、

⑮ 微分方程式の解、

⑯ 積分方程式の解、

等の計算を端末の前に座、で行なえるような状況になりつつあり、*上記のような数式の処理を機械的にくり返し行なう必要がある場合、人間がこつこつとやるのはどうかと思われる時代になりつつある。まさしく、

"When you are in Rome, do as Romans do"
をもいって、

"When you are in trouble, do as symbolic formula-manipulation specialists do"

があたりまえなのである。現在の所、また、「数式処理を行なう事

*①~⑯は、日本国内で使える数式処理で行なえる。

④~⑯までの高度な処理は、また実用になり、といえると言いかねないが、うまくゆけば、使うことで大いに助かる程度までにはなるといえる。

によって、この計算が可能にな、た」という主旨の論文は、毎年十編程度しか出版されてい、ない。数値計算においては、計算機を使うのがあたりまえで、わざわざ「数値計算を行なう事によって、この計算が可能にな、た」と書くような論文は、今ではまれである。い、がれ数式処理もそのような状況になるであろう(そうなるようにしなければなら、ないと思、っている)。こうな、るころでは遅いのである。今の内に道具としての数式処理を使いこ、なし、将来に備え、論文を書きまくろうではないか！ 今がその一番のチャンスなのである！

3. 数式処理システムを使用する上での注意点

数式処理は一般には、メモリーとCPU時間を多大に消費するとみなさねばならない。だ、いたいの目安としては、四則演算のストロークがmsオーダー、一つの項を記憶するのに数語のメモリーというのが妥当な線であろう。数式処理はこれ以外に、数値計算にはない特殊性がある。例えば、

- ①最終結果は簡単でも、中間結果が、べらぼうな大きさになる事が普通である (intermediate swell)。(前の問題2の場合、(左辺-右辺)を計算した時に、簡約する前の項の数は1000項以上であり、最終的には簡約化する事で0になる。) 実際の処理においては、CPU時間よりも、メモリーの上限

で処理が不可能になる事が多い。従って、中間結果が大きくならないような工夫を意識的に行なう必要が、問題が複雑あるいは高次になればなるほどでてくる。例えば、行列演算について考えてみると、数値計算では、何の問題もないが、行列要素が記号になると、とたんに、その次数が高次であればあるほど、この中間結果の爆発が問題になってくる。従って、実際の行列(式)の数式処理では、「行列要素をなるべく小さな式にするなどの工夫を行なう、行列演算を行なう後に、元の式にもどす(代入する)」といった手順で処理する必要性が出てくる。

②これは①と相入れない側面であるが、CPU時間の面になって考えてみると、一度計算した値は、なるべく記憶しておき、同じ計算をくり返しては行なわない、という方針が大事である事も多い(いずれにしても *ms* オークターの計算機で処理しているのであるから)。数値計算における時のように、*sin*, *cos*, *exp* の値を何度もくり返し計算するプログラムは、数式処理においては非常に危険である。

③処理シスと人間とのインターフェースがそれほど良くない。これは、ユーザー数が少ないという点、システムを作る人と、使う人が同一人ではないといった点に帰因すると思われる。しかし、いずれ、人間とのインターフェースの

問題は改良されてゆくと思われる。それまでの間は、使う方が、受け身になって、使わしていただくという感覚で使わないと、システムを使いこなせない。

とい、た点である。アルゴリズム、計算方法の少しの違いで一桁ぐらいの計算時間の違い、あるいは、できる計算もできなくなってしまうことがある。しかし、これらの点は今の所、しかなないと割り切っていてしまい、問題点を頭の片すみにおしやり、計算アルゴリズムに注意を払って数式処理システムにぶち当たれば、現在のポケット電卓程度の手軽さで、数式の処理を計算機に行なわせる事は可能である。

いふれにしろ、「習うより慣れろ」の精神が大切なのである。「数式処理を行なわない計算なんて」と思うようにすれば、あなたはもう大丈夫、研究のトップを走る事ができる事受け合いです。

4. 研究者の日本で使用可能な数式処理システム一覽

- ① ASHMEDAI …… 名大アラ研、理研で使用可。QED (量子電気力学) 計算用システム。一般の数式処理用にも使えなくはないが、数式処理を行なわせる言語が、FORTRAN, 又はALGOL のようでないで、かなりとっつきが悪い。システムはFORTRANで書かれているので、ポータビリティ

は高い。バッチ処理専用。

- ② CAMAL …… 名大プラ研。英 Cambridge 大学で開発された FORMAC と同程度の機能を有するシステム。システムは、BCPL で書かれており、ASHMEDAI ほどではないが、ポータビリティは高い部類に属する。処理スピードは REDUCE の 3~10 倍。天体力学に出てくる計算に向いているが、汎用にも十分使用できる。人間とのインターフェースは良い。使える変数の文字数は一文字という制限があるが、そのかわり、データを記憶するのに必要なメモリーは少なく、すむような構造をしているので、メモリーの少ない所での使用に向く。メモリーネックの計算では、これを使うのが良いだろう。バッチ処理専用。

- ③ MACSYMA …… 東芝 Multics システム。米国では、MIT, J. Moses の方針により、ARPA を通って使用できる。KDD ICAS サーバスを利用すれば、使えなくはないが、かなり不便だし、お金もかかりそう。(東芝にたのめば使える可能性があるが、version がかなり古いので、バグが多い。) MIT で開発中の LISP machine が \$150,000 で 1980 年 12 月に出荷される予定であるが、その上で MACSYMA 及び、その他 MIT で動いているソフトウェアが動くというこたなので、それを購入するのも一案である。最近カリ

フォルニア大学バークレー校の R. J. Fateman により、
 VAX 11/780 で動く Unix 上に MACSYMA の VAX 版とも
 言うべき VAXSYMA が開発されたので、何とかして入手す
 れば日本でも動く可能性はある。オリジナルの MACSYMA
 は DEC-10 の上で動く巨大なソフトウェアであり、10 数年
 にわたる研究成果の集積である。(パターンマッチングに
 よる積分で有名な SIN, 人間にとって見やすい数式の二
 次元出力を行なう事で有名な MATHLAB, その他 Project
 MAC の成果が取り込まれている。尚 MATHLAB は DECUS
 のライブラリーの中に含まれているので、DEC-10, -20 を所
 有しているユーザーは利用できるはずである。数解研 DEC-
 2020 の上では使える。) システムは LISP の一種 MACLISP
 で書かれており、ソースが入手できれば、他の IBM 系の機
 械で動かす事は原理的には可能である。ただし、MACSYMA
 を動かすには大量のメモリーを必要とするので、小さなセ
 ンターでは動かせ得ないであろう。MACSYMA を用いれ
 ば、定積分、不定積分、因数分解、グラフ出力、微分、微
 分方程式を解くこと、Taylor 展開、limit 計算、etc. を
 実行できる、とのことである。(ただしそれらが実用にな
 るかどうかは別)。TSS での使用に向く。

④ μ -MATHTM 理研。Intel 8080 マイクロコン上で動く

く教育用システム。簡単な積分、行列演算、非線形方程式の解等が求められるので、使用目的に限れば、卓上型数式処理システムとして十分に利用できる。ソフトウェアは約10万円。ハードウェアとして64kbのメモリー、ディスク装置を持つマイクロコン(約100万円)を必要とする。1分間に、 $290!$, $(1+x)^{20}$, $(x_1+x_2+\dots+x_9)^2$, $\sin(x_1+x_2+\dots+x_5)$ の各展開が可能である。LISPとほぼ一対一に対応する μ -LISPで表われており、その気になれば大型機の上で動かす事も可能。簡単な処理ならば、大型機と十分に立ちうちできる。システムはインタープリター方式で動作するが、最近、コンパイラーが装備されたとのことであり、さらに高速化されたとみるのが良いだろう。TSSでの使用。いつでも、どこでも好きな時に使え、しかも安いので、パフォーマンス/コストは抜群に良い。

- ⑤ PL/I-FORMAC IBMデータセンター、名大プラ研。
1965年、J. Sammet 氏らにより、天体力学に於ける、 f, g 級数の計算に使用されたので有名。おともとがバッチ指向の処理系なので、会話処理には向かない。しかし、簡単な会話処理を行なうプログラムを書く事により、ある程度の会話処理を行なわせる事は可能。数式処理に必要な基

本機能の集合体のようなシステムで、パターンマッチング、積分、*limit* 計算などの高度な仕事を行わせるには、ユーザーが各自プログラムする必要があり、その点、使いがては良くない。IBM PL/I コンパイラについて良く知っておれば、他の機械への移植は、困難であるが可能。

- ⑥ REDUCE …… 北大、東北大、東大、名大、京大、九大各大型計算機センター、原研、高エネルギー研、理研、筑波大電子情報系、東京工大総合情報処理センター、東大情報科学科、分子研、プラ研、数解研、広島大情報処理センター。これは汎用数式処理システムであるが、QED (量子電気力学) に出てくる行列計算が能率よく行なえるよう設計されているので、高エネルギー物理学の計算にも使える。もとユタ大学にいた A. C. Hearn が作成。かなり積極的に distribution を行なっており、UNIVAC, IBM, DEC, CDC の各計算機の上で動作している。システムは LISP に一皮、皮をかぶせた SLISP 言語で書かれており、システムのポータビリティは高い。最近の版では、不定積分が実行できる。いまだ、因数分解、多倍長浮動小数点パッケージ等を含んだシステムが公開になるものと思われる。
- TSS、バッチ両方で使用できる。現在日本で動いている大部分の REDUCE は、HLISP-REDUCE と呼ばれているもの

ので、Hearn が公開している版と比べて、倍程度スピードは速い。しかし、不定積分を行なう事はできない。又、TSSでの利用が、とても良いように作られており、使いかたでは悪くはない。特にプラ研のシステムでは、LISPのレベルから、ファイルをeditする事ができるので、REDUCEの環境から抜け出す事なく、仕事ができるようになり、この事で、仕事の能率が格段に高い。

5. 終わりに.

4の一番最後に紹介したREDUCEは、全世界で使用できるシステムになり、この使い方を知らないと損はないように思われます。ただ一番の問題点は、マニュアル、使用例が少ない点であり、これは、利用者の協力を得て、早急に何とかしたいと考えっておりますので、今しばらくお待ちいただきたいと思っております。その他のシステムについてのお問い合わせは、プラ研までお願いいたします。いずれにしろ、数式処理システムの多くは、FORTRANを知っていれば使えますから、まず、数式の検算などに使ってみて、効果を確認しながら、順次、困難な問題に進まれるのが効率的だと思われます。

6. 文献

文献については、

Y. Kanada "HLISP and Supplementary HLISP-REDUCE manual" からまづ引きされた。

付 録

4.で紹介した各システムの出力例を付録として付けおく。各システムの特徴がおわかりいただければ幸いである。簡単な説明を表き込んでおいたので参考にされた。MACSYMAのきれいな出力例は入手できなかった、含めなかった。μ-MATH⁹⁹の出力は理研、情報科学研究室の相馬氏にお世話になった。

付録 1. ASHMEDA / 出力例 (Legendre 多項式の計算)

```

6.560 91 20000 . C - - - - -
6.560 91 20000 . C - - - - - LEGENDRE POLYNOMIALS, AN EXAMPLE.
6.560 91 20000 . C - - - - -
6.560 91 20000 . C - - - - -
6.600 91 20000 . ← 入力イメージを意味する
6.600 91 20000 . C AS AN OBTUSE EXAMPLE, CONSIDER THE LEGENDRE POLYNOMIALS.
6.600 91 20000 . C (THE TROUBLE WITH THE EXAMPLE IS THAT WHILE THESE PROGRAMS
6.600 91 20000 . C ARE FAST FOR DEALING SIMPLY WITH LARGE EXPRESSIONS,
6.600 91 20000 . C DEALING WITH MANY EXPRESSIONS IS QUITE AWKWARD.)
6.600 91 20000 . C
6.600 91 20000 . C THESE POLYNOMIALS ARE GIVEN BY THE RECURSION RELATION
6.630 91 20000 . C  $P(L) = Z * (2 - 1/L) * P(L-1) - (1 - 1/L) * P(L-2)$  WITH  $P(-1) = 0$ ,  $P(0) = 1$ .
6.630 91 20000 . C WHICH WE IMPLEMENT AS FOLLOWS.
6.630 91 20000 . C
6.630 91 20000 . C IN THIS EXAMPLE, WE KEEP IDENTITIES FOR THE PREVIOUS 2 POLYNOMIALS AND
6.640 91 20000 . C THE PARAMETER (IL) NEEDED IN THE RECURSION RELATION. ON EACH ITERA
6.640 91 20000 . C SLIDE  $P(L-1)$  TO  $P(L-2)$ ,  $P(L)$  TO  $P(L-1)$  AND CHANGE THE SUBSTITUTION
6.640 91 20000 . C FOR 'IL'.
6.640 91 20000 . C
6.640 91 20000 . C INITIALIZE THE MANY EXPRESSIONS NEEDED.
6.640 91 20000 . INPUT,10 B * INPUT,11 C * INPUT,12 IL * INPUT,7 1 * INPUT,4 0
6.640 91 20000 2 INPUT ( 743217452) 10
6.640 97 20000 2 INPUT ( 743217452) 11 ← システムからの出力
CR SYM 1 004D K( 11)
6.640 103 20000 2 INPUT ( 743217452) 12
CR SYM 1 004F L( 12)
CR SYM 2 001A IL ( 738138144)
6.640 109 20000 2 INPUT ( 743217452) 7
CR SYM 1 004B G( 7)
6.640 114 20000 2 INPUT ( 743217452) 4
CR SYM 1 0049 D( 4)
6.640 119 20000 . SET/MODE(1)=1
6.640 119 20000 4 SET (1516991312) 1061560932 1 1
6.640 119 20000 . INPUT,2 0 * INPUT,6 Z*(2-IL)*B-(1-IL)*C * INPUT,3 1
6.640 119 20000 2 INPUT ( 743217452) 2
CR SYM 1 0047 B( 2)
6.680 124 20000 2 INPUT ( 743217452) 6
CR SYM 1 006D F( 6)
CR SYM 2 001B Z (2060114368)
6.680 177 20000 2 INPUT ( 743217452) 3
CR SYM 1 0069 C( 3)
6.680 182 20000 .
6.680 182 20000 . C '7' = P(L)
6.680 182 20000 . C '10' AND '4' ARE LHS AND RHS FOR  $P(L-1) = \dots$ 
6.680 182 20000 . C '11' AND '5' ARE LHS AND RHS FOR  $P(L-2) = \dots$ 
6.680 182 20000 . C '2' AND '3' ARE LHS AND RHS FOR  $NN*IL = 1$ 
6.680 182 20000 . C '12' IS THE INCREMENT FOR THE LHS OF  $NN*IL = 1$ .
6.680 182 20000 . C ('NN' STANDS FOR AN INTEGER. ON SUCCESSIVE PASSES, IT WILL BE 1,2,
6.680 182 20000 . C THEN, CHANGE THE LHS OF 'NN*IL=1'
6.680 182 20000 .
6.680 182 20000 . OPEN,2 PCOLCT,8 OPEN,12 COLCT,8 MOVE,2,8
6.680 182 20000 2 OPEN (1222184272) 2
6.680 182 19993 2 PCOLCT(1274858638) 8
CR SYM 1 0067 H( 8)
6.680 182 20000 2 OPEN (1222184272) 12
6.680 182 19993 2 COLCT ( 269646100) 8
FLIP AUT H. 17 WORDS, 1 TREES, 1 CALLS, 1 TERMS, 0 ZERGES, 0 TEST
6.720 191 20000 3 MOVE (1062548628) 2 8
6.720 186 20000 .
6.730 186 20000 . C NEXT, SLIDE DOWN  $P(L-1)$  TO  $P(L-2)$  AND  $P(L)$  TO  $P(L-1)$ 
6.730 186 20000 .
6.730 186 20000 . MOVE,5,4,4,7
6.730 186 20000 5 MOVE (1062548628) 5 4 4 7
CR SYM 1 0065 E( 5)
6.730 186 20000 .
6.730 186 20000 . C SET UP SUBSTITUTION FOR  $P(L-1)$  AND  $P(L-2)$  AND  $NN*IL =$ .
6.730 186 20000 .
6.730 186 20000 . MAKEID,1,2,3,10,4,11,5
6.730 186 20000 8 MAKEID(1032753478) 1 2 3 10 4 11 5
CR SYM 1 0063 A( 1)
6.730 206 20000 .

```

プログラムのうちは、著者は理解
してないので略す。

```

6.760 206 20000 . C GENERATE P(L) AND PRINT IT.
6.760 206 20000 .
6.760 206 20000 . OPEN,6 SUBST,1 COLCT,7 LOOK,7
6.760 206 20000 2 OPEN (1222184272) 6
6.760 206 19993 2 SUBST (1549393988) 1
6.760 206 19988 2 COLCT ( 269646100) 7
FLIP ADT G. 17 WORDS, 1 TREES, 2 CALLS, 1 TERMS, 0 ZEROES, 1 TEST
6.760 215 20000 2 LOOK ( 982938020) 7
+ 3*Z*1/2
6.760 215 20000 . C THIS CAN NOW BE REPEATED.
6.760 215 20000 .
6.760 215 20000 . OPEN,2 PCOLCT,8 OPEN,12 COLCT,8 MOVE,2=8,5=4,4=7,1=7
6.760 215 20000 2 OPEN (1222184272) 2
6.760 215 19993 2 PCOLCT(1274858638) 8
6.760 215 19983 2 OPEN (1222184272) 12
6.760 215 19976 2 COLCT ( 269646100) 8
FLIP ADT H. 17 WORDS, 1 TREES, 2 CALLS, 1 TERMS, 0 ZEROES, 1 TEST
6.760 224 20000 9 MOVE (1062548628) 2
6.760 190 20000 . MAKEID(1)2=3*10=4*11=5 OPEN,6 SUBST,1 COLCT,7 LOOK,7
6.760 190 20000 8 MAKEID(1032753478) 1 2 3 10 4 11 5
6.760 214 20000 2 OPEN (1222184272) 6
6.810 214 19993 2 SUBST (1549393988) 1
6.810 214 19988 2 COLCT ( 269646100) 7
FLIP ADT G. 23 WORDS, 2 TREES, 4 CALLS, 2 TERMS, 0 ZEROES, 1 TEST
6.810 229 20000 2 LOOK ( 982938020) 7
+ 3*Z*1/2 - 1/2
6.810 229 20000 . OPEN,2 PCOLCT,8 OPEN,12 COLCT,8 MOVE,2=8,5=4,4=7,1=7
6.810 229 20000 2 OPEN (1222184272) 2
6.810 229 19993 2 PCOLCT(1274858638) 8
6.820 229 19983 2 OPEN (1222184272) 12
6.820 229 19976 2 COLCT ( 269646100) 8
FLIP ADT H. 17 WORDS, 1 TREES, 2 CALLS, 1 TERMS, 0 ZEROES, 1 TEST
6.820 238 20000 9 MOVE (1062548628) 2
6.820 198 20000 . MAKEID(1)2=3*10=4*11=5 OPEN,6 SUBST,1 COLCT,7 LOOK,7
6.820 198 20000 8 MAKEID(1032753478) 1 2 3 10 4 11 5
6.820 226 20000 2 OPEN (1222184272) 6
6.820 226 19993 2 SUBST (1549393988) 1
6.840 226 19988 2 COLCT ( 269646100) 7
FLIP ADT G. 25 WORDS, 2 TREES, 6 CALLS, 2 TERMS, 0 ZEROES, 4 TEST
6.840 243 20000 2 LOOK ( 982938020) 7
- 3*Z*1/2 + 5*Z*Z*1/2
6.840 243 20000 . OPEN,2 PCOLCT,8 OPEN,12 COLCT,8 MOVE,2=8,5=4,4=7,1=7
6.840 243 20000 2 OPEN (1222184272) 2
6.840 243 19993 2 PCOLCT(1274858638) 8
6.840 243 19983 2 OPEN (1222184272) 12
6.840 243 19976 2 COLCT ( 269646100) 8
FLIP ADT H. 17 WORDS, 1 TREES, 2 CALLS, 1 TERMS, 0 ZEROES, 1 TEST
6.860 252 20000 9 MOVE (1062548628) 2
6.860 204 20000 . MAKEID(1)2=3*10=4*11=5 OPEN,6 SUBST,1 COLCT,7 LOOK,7
6.860 204 20000 8 MAKEID(1032753478) 1 2 3 10 4 11 5
6.860 238 20000 2 OPEN (1222184272) 6
6.860 238 19993 2 SUBST (1549393988) 1
6.860 238 19988 2 COLCT ( 269646100) 7
FLIP ADT G. 32 WORDS, 3 TREES, 8 CALLS, 3 TERMS, 0 ZEROES, 4 TEST
6.860 262 20000 2 LOOK ( 982938020) 7
+ 35*Z*Z*Z*1/8 - 15*Z*Z*1/4 + 3*1/8
6.860 262 20000 . OPEN,2 PCOLCT,8 OPEN,12 COLCT,8 MOVE,2=8,5=4,4=7,1=7
6.860 262 20000 2 OPEN (1222184272) 2
6.860 262 19993 2 PCOLCT(1274858638) 8
6.860 262 19983 2 OPEN (1222184272) 12
6.860 262 19976 2 COLCT ( 269646100) 8
FLIP ADT H. 17 WORDS, 1 TREES, 2 CALLS, 1 TERMS, 0 ZEROES, 1 TEST
6.860 271 20000 9 MOVE (1062548628) 2
6.860 213 20000 . MAKEID(1)2=3*10=4*11=5 OPEN,6 SUBST,1 COLCT,7 LOOK,7
6.860 213 20000 8 MAKEID(1032753478) 1 2 3 10 4 11 5
6.860 252 20000 2 OPEN (1222184272) 6
6.910 252 19993 2 SUBST (1549393988) 1
6.910 252 19988 2 COLCT ( 269646100) 7
FLIP ADT G. 35 WORDS, 3 TREES, 10 CALLS, 3 TERMS, 0 ZEROES, 7 TEST
6.910 279 20000 2 LOOK ( 982938020) 7
+ 15*Z*1/8 - 35*Z*Z*1/4 + 63*Z*Z*Z*1/8
6.910 279 20000 .
6.910 279 20000 . RTRNR,1,12
6.910 279 20000 3 RTRNR (1468944340) 1 12
CR SYM 1 0061 1( 9)

```

付録 2. CAMAL 出力例 (f.g. 係数 a 計算)

```

1  F[19]; G[19]
2
3      F[0] = 1; G[0] = 0; U = -3ab; V = c-2bb; W = -b(a+2c)
4      FOR N = 1:1:18
5          F[N] = UdF[N-1]/da + VdF[N-1]/db + WdF[N-1]/dc - aG[N-1]
6          G[N] = UdG[N-1]/da + VdG[N-1]/db + WdG[N-1]/dc + F[N-1]
7      PRINT[F[N]] ; PRINT[G[N]] ; PRINT[TIME]
8      REPEAT
9
10 STOP
11 PM(0)
12 END

```

↑
プログラーリスト

$$UdF[N-1]/da = U * \frac{dF[N-1]}{da}$$

出力の一部
↓

CAMAL -- 6 Feb 1975

F[1] = 0

G[1] = 1

TIME=0.00 SECS

F[2] = - a

G[2] = 0

TIME=0.00 SECS

F[3] = 3 b a

G[3] = - a

TIME=0.00 SECS

F[4] = 3 a c + a^2 - 15 b^2 a

G[4] = 6 b a

TIME=0.01 SECS

F[5] = -(45 b a c + 15 b a^2 - 105 b^3 a)

G[5] = 9 a c + a^2 - 45 b^2 a

TIME=0.01 SECS

F[6] = -(45 a c^2 + 24 a^2 c + a^3 - 630 b^2 a c - 210 b^2 a^2 + 945 b^4 a)

G[6] = -(180 b a c + 30 b a^2 - 420 b^3 a)

G[18] までの計算時間 0.9 秒

付録 4. PL/I - FORMAC 出力例

(Reversed Power Series の計算)

プログラマリスト

↓

INPUT TO KFA FORMAC PREPROCESSOR - VERSION 3 - GMD, DECEMBER

```

1      SERIES:  PROCEDURE OPTIONS(MAIN);
2      FORMAC_OPTIONS;
3      OPTSET(EXPND);
4      OPTSET(LINELENGTH=60);
5      OPTSET(INT);
6      /* THIS PROGRAM WILL GENERATE UP TO NMAX-1 COEFFICIENTS
7      OF A REVERSED POWER SERIES.  THE VALUE OF NMAX IS
8      ENTERED BELOW.      */
9      NMAX = 8;
10     DO N=2 TO NMAX;
11         LET(N="N");
12         LET(A(N) = 0);
13         DO I = 2 TO N-1;
14             LET (MUM = N);
15             DO K = 2 TO I;
16                 LET(K="K");
17                 LET(MUM=MUM*(N+K-1));
18             END;
19             LET(I="I");
20             LET(TEMP=MUM/FAC(I));
21             M=N-I-1;
22             LET(C(0) = B(1)**I);
23             LET(M="M");
24             DO J = 1 TO M;
25                 LET(J="J");
26                 LET(C(J)=0);
27                 DO K = 1 TO J;
28                     LET(K="K");
29                     LET(C(J)=C(J)+(K*I-J+K)*B(K+1)*C(J-K) );
30                 END;
31                 LET( C(J) = (1/(J*B(1))) * C(J) );
32             END;
33             LET (A(N) = A(N) + TEMP * C(M) );
34         END;
35         LET (A(N-1) = A(N)/N + B(N-1) );
36         PRINT_OUT(A(N-1) );
37         PUT SKIP(2);
38     END;
39     END SERIES;

```

FACOM OSIV/F4 PL/I V05L10

SERIES: PROCEDURE OPTIONS(MAIN);

** SOURCE STATEMENT LISTING **

PL/Iに変換されたソースプログラム

```

STMT LVL NST
1      0      SERIES: PROCEDURE OPTIONS(MAIN);

2      1      0  /* DECLARES FOR RUN-TIME ROUTINES. */
                DECLARE
                DENFMC1 ENTRY(CHAR(*)),
                DENFMC2 ENTRY(CHAR(*)),
                DENFMC3 ENTRY(FIXED BIN(31),FIXED BIN(31)),
                DENFMC4 ENTRY(CHAR(*)),
                DENFMC7 ENTRY(CHAR(*)),
                DENFMC8 ENTRY(CHAR(*)),
                DENFMC9 ENTRY(CHAR(*)) RETURNS(BIN FLOAT(53)),
                DENFMCA ENTRY(CHAR(*)) RETURNS(BIN FIXED(31)),
                DENFMCB ENTRY(CHAR(*)) RETURNS(BIN FIXED(31)),
                DENFMCC ENTRY(CHAR(*)) RETURNS(BIN FIXED(31)),
                DENFMCD ENTRY(CHAR(*),BIN FLOAT(53)),
                DENFMCE ENTRY(CHAR(*),BIN FIXED(31)),
                DENFMCF ENTRY(CHAR(*),CHAR(*) RETURNS(BIT(1))),
                DENFMCG ENTRY(CHAR(*),CHAR(*) RETURNS(BIT(1))),
                DENFMCH ENTRY,
                DENFMCL ENTRY(CHAR(*)) RETURNS(POINTER),
                DENFMCM ENTRY(CHAR(*),ENTRY);

3      1      0  CALL DENFMC3(03,0);
4      1      0  CALL DENFMC3(00,60);
5      1      0  CALL DENFMC3(05,0);
                /* THIS PROGRAM WILL GENERATE UP TO NMAX-1 COEFFICIENTS
                OF A REVERSED POWER SERIES. THE VALUE OF NMAX IS
                ENTERD BELOW. */
6      1      0      NMAX = 8;
7      1      0      DO N=2 TO NMAX;
8      1      1  CALL DENFMC1('N='||N||');
9      1      1  CALL DENFMC1('A(N)=0');
10     1      1      DO I = 2 TO N-1;
11     1      2  CALL DENFMC1('MUM=N');
12     1      2      DO K = 2 TO I;
13     1      3  CALL DENFMC1('K='||K||');
14     1      3  CALL DENFMC1('MUM=MUM*(N+K-1)');
15     1      3      END;
16     1      2  CALL DENFMC1('I='||I||');
17     1      2  CALL DENFMC1('TEMP=MUM/FAC(I)');
18     1      2      M=N-I-1;
19     1      2  CALL DENFMC1('C(0)=B(1)**I');
20     1      2  CALL DENFMC1('M='||M||');
21     1      2      DO J = 1 TO M;
22     1      3  CALL DENFMC1('J='||J||');
23     1      3  CALL DENFMC1('C(J)=0');
24     1      3      DO K = 1 TO J;
25     1      4  CALL DENFMC1('K='||K||');
26     1      4  CALL DENFMC1('C(J)=C(J)+(K*I-J+K)*B(K+1)*C(J-K)');
27     1      4      END;
28     1      3  CALL DENFMC1('C(J)=(1/(J*B(1)))*C(J)');
29     1      3      END;
30     1      2  CALL DENFMC1('A(N)=A(N)+TEMP*C(M)');
31     1      2      END;

```

FACOM OSIV/F4 PL/I V05L10

SERIES: PROCEDURE OPTIONS(MAIN);

```

STMT LVL NST
32     1      1  CALL DENFMC1('A(N-1)=A(N)/N+B(N-1)');
33     1      1  CALL DENFMC2('A(N-1)');
34     1      1      PUT SKIP(2);
35     1      1      END;
36     1      0  END SERIES;

```

44

$$A(1) = B(1)$$

出力結果

$$A(2) = B(2) + 2 B(1)^2$$

$$A(3) = 5 B(2) B(1) + B(3) + 5 B(1)^3$$

$$A(4) = B(4) + 3 B(2)^2 + 21 B(2) B(1)^2 + 6 B(3) B(1) + 14 B(1)^4$$

$$A(5) = 7 B(4) B(1) + B(5) + 28 B(2)^2 B(1) + 7 B(2) B(3) + 84 B(2) B(1)^3 + 28 B(3) B(1)^2 + 42 B(1)^5$$

$$A(6) = 8 B(4) B(2) + 36 B(4) B(1)^2 + 8 B(5) B(1) + B(6) + 12 B(2)^3 + 180 B(2)^2 B(1)^2 + 72 B(2) B(3) B(1) + 330 B(2) B(1)^4 + 4 B(3)^2 + 120 B(3) B(1)^3 + 132 B(1)^6$$

$$A(7) = B(7) + 90 B(4) B(2) B(1) + 9 B(4) B(3) + 165 B(4) B(1)^3 + 9 B(5) B(2) + 45 B(5) B(1)^2 + 9 B(6) B(1) + 165 B(2)^3 B(1) + 45 B(2)^2 B(3) + 990 B(2)^2 B(1)^3 + 495 B(2) B(3) B(1)^2 + 1287 B(2) B(1)^5 + 45 B(3)^2 B(1)^2 + 495 B(3) B(1)^4 + 429 B(1)^7$$

付録5. REDUCE 出力例 (次頁にf.g級数計算の例あり)

```
@execm("e @save@.text non")
```

```
E
```

```
f /g2/
```

```
l *
```

```
scalar f1,f2,g1,g2;
```

```
v
```

```
f g2:=f1+deps*df(g1,eps)+dmu*df(g1,mu)+dsig*df(g1,sig)$
```

```
down write "g(",i,") := ",g2;
```

```
down f1:=f2$
```

```
down g1:=g2
```

```
c /g2/g2;:/
```

```
g1:=g2;
```

```
down
```

```
end;
```

```
b
```

```
end;
```

```
c /;:/; /
```

```
end;
```

```
i
```

```
INPUT
```

```
E
```

```
end s
```

```
KEQ52460I SAVED IN DATA SET 'C004.@SAVE@.TEXT' \
```

```
= 0
```

```
@begin(78)
```

```
(STEP/TOTAL) TIME = (2/2) MS
```

```
>in in; ← 修正したファイルを読み込む。
```

```
(STEP/TOTAL) TIME = (18/20) MS
```

```
ON TEST$
```

```
(STEP/TOTAL) TIME = (7/27) MS
```

```
FOR I:=2 STEP 2 UNTIL 50 SUM I**2;
```

```
22100
```

```
(STEP/TOTAL) TIME = (57/84) MS
```

```
W:=FOR I:=1:10 PRODUCT I;
```

```
W := 3628800
```

```
(STEP/TOTAL) TIME = (24/109) MS
```

```
ARRAY A(10);
```

```
*** A ALREADY DEFINED AS ARRAY
```

```
(STEP/TOTAL) TIME = (9/119) MS
```

```
A(0):=1$
```

```
(STEP/TOTAL) TIME = (8/127) MS
```

```
FOR I:=1:10 DO A(I):=I*A(I-1);
```

← LISP から ファイル を 修正

以下.一部出力例略

SCALAR F1,F2,G1,G2;

(STEP/TOTAL) TIME = (11/275) MS

DEPS:=-SIG*(MU+2*EPS)\$

f.g. 級数の計算例

(STEP/TOTAL) TIME = (16/292) MS

DMU:=-3*MU*SIG\$

(STEP/TOTAL) TIME = (13/305) MS

DSIG:=EPS-2*SIG**2\$

(STEP/TOTAL) TIME = (14/319) MS

F1:=1\$

(STEP/TOTAL) TIME = (6/325) MS

G1:=0\$

(STEP/TOTAL) TIME = (6/331) MS

FOR I:=1:8 DO BEGIN F2:=-MU*G1+DEPS*DF(F1,EPS)+DMU*DF(F1,MU)+DSIG*DF(F1,SIG)\$

WRITE F(I,) := ,F2;

G2:=F1+DEPS*DF(G1,EPS)+DMU*DF(G1,MU)+DSIG*DF(G1,SIG)\$

WRITE G(I,) := ,G2;

F1:=F2\$

G1:=G2;

END;

F(1) := 0

G(1) := 1

F(2) := - MU

G(2) := 0

F(3) := 3*MU*SIG

G(3) := - MU

F(4) := MU*(3*EPS + MU - 15*SIG²)

G(4) := 6*MU*SIG

F(5) := 15*MU*SIG*(- 3*EPS - MU + 7*SIG²)

G(5) := MU*(9*EPS + MU - 45*SIG²)

F(6) := MU*(- 45*EPS² - 24*EPS*MU + 630*EPS*SIG² - MU² + 210*MU*SIG² - 945*SIG⁴)

G(6) := 30*MU*SIG*(- 6*EPS - MU + 14*SIG²)

以下出力例 略