パーソナルコンピュータによる数式処理

大久保 明伸*

Symbolic Manipulation Using Personal Computer

Akinobu Ohkubo

Abstract

This paper describes a symbolic manipulation system using personal computer.

Program packages of this system are described with muSIMP (under the CP/M) and consist of equation package and matrix package. These packages are suitable for student who wants to solve the problem of technical practice. The students may freely select some comand of these package using muSIMP.

This system is very simple and useful for technical education.

1. まえがき

数式処理とは数式を数式のまま処理することである．パーソナルコンピュータの急激な普及にともない，数値計算用の「パーソナルなツール」として広く利用されている．数年前までの共同利用による計算センターを中心にした計算機の利用に比べると，はるかに便利になった．しかし，利用の中心はあいかわらず，数値計算が中心である．もちろん，大型計算機上には，以前から数式処理システムも使用可能であった．しかし，パーソナルコンピュータのように，いつでも，どこでも自由に使用できる環境にない．現在のパーソナルコンピュータのハードウェアとソフトウェアは，速度の点を除けば，大型計算機の環境に匹敵する．本稿ではパーソナルコンピュータ用の数式処理言語 mu SIMP の上に実験的に作成した数式処理パッケージを示し，パーソナルツールとしての可能性を論じる．ただし mu SIMP は CP/Mの管理下で動作する LISP 系のインタプリタである．

2. パーソナルコンピュータによる数式処理システムの条件

パーソナルコンピュータシステムを数式処理システムとして，利用する場合，パーソナルコンピュータがもつべき条件について，将来の構想も合せて議論する．ここで述べるシステムは個人の研究開発や CAI 的なシステムとして考え，共同利用的なシステムは考えないことにする．

2．1 ハードウェアの構成

本システムの概念的な構成を図1に示す．システムは，2台のパーソナルコンピュータにより構成する．1台はプログラム開発用に，他の1台は実行用に用いる．実際にパーソナルコンピュータ上で数式処理を行っていると，1つの結果を得るのに，数時間かかるような問題に遭遇することはよくあることである．このため1台のシステムでは，プログラムの開発や修正に困難をきたす．CPU 1 と CPU 2 はオンラインで接続されているが，フロッピーディスク装置があれば，オフラインでもかまわない．実行システムは，数式処理で得られた結果を数値計算したり，図に変換したり，他の外部装置を動作させたりすることになるので，そのための入出力装置を必要とする．大型計算機の利用は通信回線を経由するものとする．ハードディスクはプログラムやデータの単なる保存用としてばかりではなく，数式処理データベースの構築と AI 化のためにぜひ必要な装置である．以上のシステムは，価格の急激な低下により，十分実現可能なものと考えられる．

*宇部工業高等専門学校電気工学科

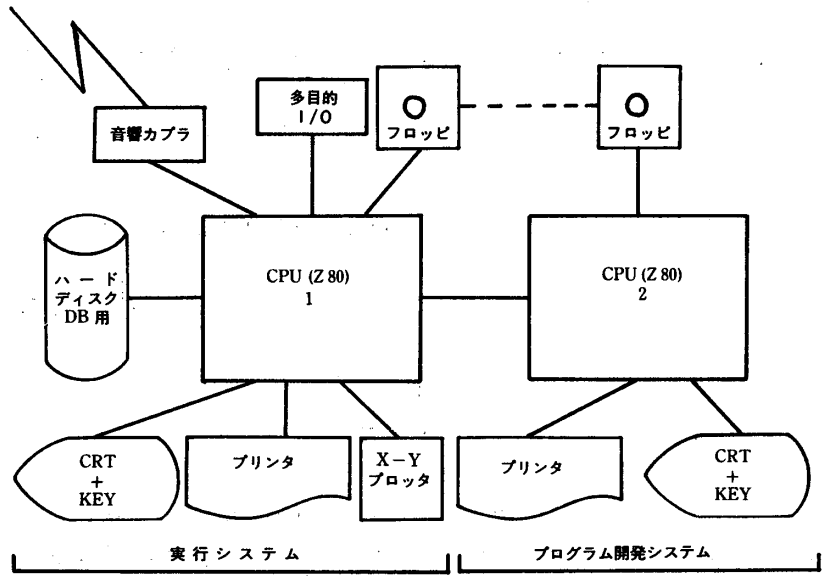

図1 数式処理システム（ハードウエア）

2．2 ソフトウェアの構成

図2に示したように，このシステムを管理するためのOSはCP/Mを想定することにする．CP/Mは現在もっとも多くのソフトウェアとユーザをもつシステムで大型計算機のそれよりも，安価で，使用しやすいシステムである．また，Z80系のCPUであれば，ほとんどのパーソナルコンピュータで利用可能である．データベースやAI化のためのソフト作成用のLISPやPROLOG等が開発され，十分な能力をもっている．また，16 bit用，32 bit用のCP/Mも徐々に開発されている．

図2に本システムのための最小構成を示す．図は，言語と数式処理システムを中心に示してある．

mu SIMPはLISP系の言語で入出力関数が数式向きに整備されている．mu MATHはmu SIMPで記述された数式処理パッケージである．かなり良くできているが，一部のプログラムはある程度熟練をしないと，使用に不便があるので，後述するように，mu MATHの一部を作り直した．UMATHはmu SIMPで現在試験的に開発している数式処理パッケージである．この開発経験を踏えて，本格的な数式処理システムを構築していく予定である．数学の公式集が，具体的な数式の計算に便利なように，この数式処理システムは数式データベースを持つと便利である．データベースの構築も，mu SIMPによるプログラムにより，準自動成生可能である．実際，UMATHの一部のプログラムはmu SIMPにより発生したプログラムを利用している．このことは自己増植可能なデータベースの構築が可能なことを示している．

他言語発生機能は，数式処理システムには必要なものである．行列や微分方程式の処理結果は，一般に複雑である．たとえばこの式を，BASICやFORTRANにより処理したい場合，人手で入力するには，困難なことが多い．大型計算機のREDUCEでも，FORTRANへの変換機能をもつように，このシステムでも変換機能を持つ．CP/Mのテキストファイルは一種類であるので，万能の変換機能をもつプログラムを作成する必要はなく，個別に言語変換をするプログラムを作れば十分であるし，簡単である．PASCALは複雑なデータ構造を直接構成するための機能を有しているため，数式データベースを構築し高度なデータ構造を作るのに便利である．数式処理システムのAI化にはPROLOGを用いて，作成するのが便利である．以上，ソフトウェアの概略について述べたが，これはすべて，CP/Mの柔軟性と豊富なソフトウェアのおかげであることに注目したい．

2．3 数式処理の電卓化[(1)]について

以上述べた数式処理システムは，本格的な開発，研究用としては，満足できるが，数値計算用の道具として，関数電卓があるように，このシステムの一部を小型化して，数式処理電卓も構成可能である．

数式処理は数値計算とちがって，一般に複雑な式の表示が得られる．そのため，電卓化には，ブラウン管ディスプレイの替りに，大型の液晶ディスプレイによる表示

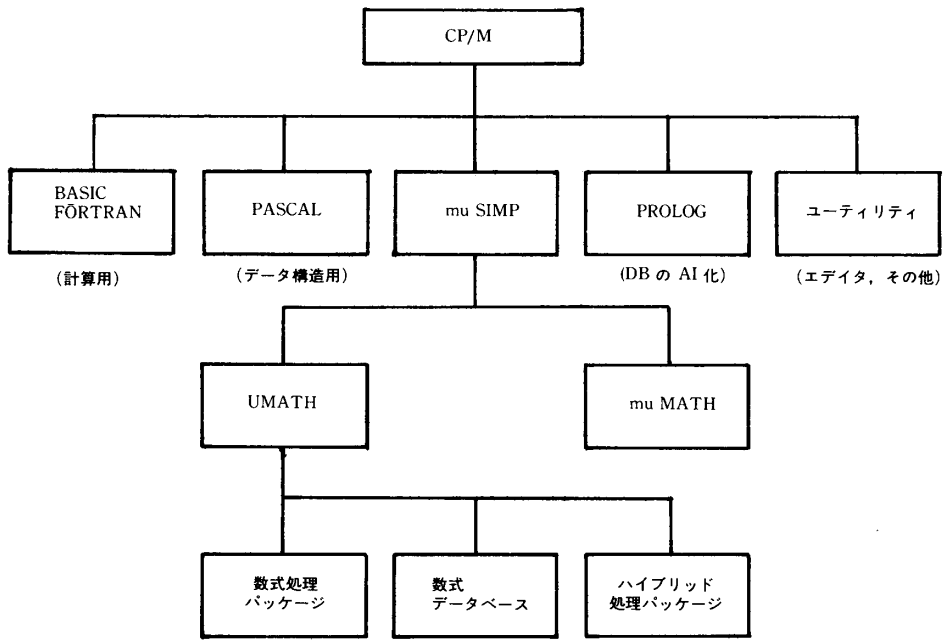

図2　数式処理システムの最小構成（ソフトウエア）

装置が必要である．これにより，小型軽量化と消費電力の問題は解決する．機能面では，どの程度の能力までをROM 化して，どのような分野に利用するかは問題である．いわゆる「ポケット電卓議論」は記憶に新しい．mu SIMP はインタプリンタであるため，機能的には，電卓的な利用が可能である．この点に関する議論は別の機会にゆずることにする．

3．mu SIMP の概要

mu SIMP は A.D.Rich と D.R.Stoutemyer の作成したもので，CP/M 上で動作する．LISP とほぼ同じ記号処理用言語である．従って，処理の対象はS式であり，プログラム自身もS式で格納される．処理系はインタプリタ方式であるため会話形式で利用可能である．必要な機能は，その都度ディスク装置より，ロード可能である．またユーザが作成したプログラムは，システム自身に組み込まれるので，この意味において，自己増植機能をもつしBASIC と同様，ダイレクトモードとプログラムモードをもつ．いずれのモードもシステムの入力プログラムにより入力するが，文法チック機能はあるものの，修正機能は有していないため，大きな入力に対しては CP/M のエディタを用いて作成する．ダイレクトモードの入力形式は

(i)　〈関数名〉(〈引数〉,〈引数〉, ……)；$^{C}_{R}$

(ii)　〈変数名〉：〈S式〉 |〈算術式〉
　　|〈関数名〉(〈引数〉)；$^{C}_{R}$

(i)はプログラム（関数）の実行を指令する．引数があたえられなければ，引数に対して値 FALSE が与えられる．この機能を利用して，大域変数と局所変数とをプログラム作成に利用できる．結果は，システム変数@に与えられる．(ii)の形式は変数への代入である．評価されたS式，算術式，関数の値が代入される．この2つの機能を用い，電卓的に使用できる．

プログラムモードは FUNCTION か SUBROUTINE ステートメント で始まり，関数の実行は，プログラム中で呼びだした場合とダイレクトモードの(i)の形式で指定された場合のみ，実行される．定義された関数はシステム内にS式に変換されて，組み込まれる．プログラムの例を下に示す．次の例は mu SIMP により記述したS式の先頭の ATOM を値とする関数である．LISP で有名な例である．

```
FUNCTION FIRSTATOM(X),
WHEN ATOM(X), X, EXIT,
FIRSTATOM(FIRST(X)),
ENDFUN ;
```

この関数の ATOM, FIRST は LISP の ATOM および CAR と同じもので，システムに組み込まれている．この関数の実行は

X：LIST(A, B, C)；←Xへの代入

FIRSTATOM(X)；←実行

とすればよい．答は @ A である．

もう少し複雑な例として，純 LISPの APPLY を LAPPLY という関数名で定義した例を図-3 に示す．2つの例で分かるように，mu SIMP は LISP とちがい，PASCAL の手続と同様に記述でき，LISPのS式より親しみやすい．これらのプログラムは入力プログラムにより，内部ではS式の形式で記憶され，評価される．

4．mu MATHの概要

mu MATH は記号処理言語 mu SIMP で記述された数式処理プログラムパッケージである．パッケージは複数のファイルに分割されていて，使用する場合には必要なファイルをロードして，mu SIMP に機能を付加することができる．mu MATH の機能の概要を**表1**に示す．以下，代表的な機能を例題により示す．

〔例1〕多項式の四則演算

```
? X+Y+X-2*Y+Z; ←問
@: 2*X - Y + Z ←答

?
@&
@: (+ (* 2 X) (* -1 Y) Z) ←S式

?
EXPD((X+Y)^4); ←展開
@: 4*X*Y^3 + 6*X^2*Y^2 + 4*X^3*Y + X^4 + Y^4
                                    ↑
                                    答

?
(X+Y)^2/(X+Y); ←問
@: X + Y       ←答
```

〔例2〕微分

```
?
DIF(SIN(SIN(SIN(X))),X); ←微分
@: COS(X) * COS(SIN(X)) * COS(SIN(SIN(X))) ←答
```

〔例3〕積分

```
?
INT((1+X^2)^(-1/2),X); ←Xに関する積分
@: LN (2*X+(4+4*X^2)^(1/2)) ←答
```

〔例4〕行列式の値

〔結果1参照〕

以上の例題中に発生する多項式の整理に関しては，7個のパラメータを適当にコントロールしなければならないので，かなりの熟練を必要とす．また配列のランダムアクセス機能には少し問題があり，あまり使用しやすいものではない．これ等 mu MATH の問題点を改良し，将来の本格的な数式処理システムのために，機能の拡張を計った．これに関する精細は次の UMATH の項でのべる．

5． muMATH の拡張 (UMATH)

muMATH は良く作られたプログラムパッケージであるが，一部の機能に不満足な点があり，また技術計算に心要な機能が欠如している．そこで，将来の本格的なパーソナル数式処理システムの開発のための準備として，表2のような機能拡張やプログラム技法の開発を行った．以下，その内で代表的なものについて説明する．

(1) 数式の微分
(2) 数式の積分
(3) 式の展開，整理(7個のパラメータ)
(4) 行列，行列式の処理
(5) 複素数の演算
(6) 600桁の整数計算
(7) その他

表1 mu MATH の機能

(1) デモプログラム 純 LISP シミュレータ，その他
(2) 配列のランダムアクセス
(3) 行列，行列式の処理
(4) 恒等式の処理
(5) 形式微分
(6) 2階線型微分方程式の解
(7) BASIC 文関数発生機能

表2 UMATH の機能

5．1　NEWTON 法による$\sqrt{a}$の高桁計算

このパッケージは NEWTON と REALP より成る．動作例を次に示す．NETON の引数N; Eはそれぞれ$\sqrt{N}$, 誤差10^{-E}に対応する．結果は分数で得られるので，REALP により整数部，小数部のリスト形式により得られるようにした．その他の引数は，局所変数である．(結果2参照)

5．2　純 LISP パッケージ

muSIMP も LISP であるが，この言語のプログラム例として，純 LISP インタプリタを記述した．このパッケージにより LISP 入門教書の各種のプログラム例を実行できる．muSIMP の言語理解のために必要なパッケージである．このパッケージの利用例は別稿にて報告する．

5．3　行列式演算パッケージ

muMATH による行列式計算のアルゴリズムは三角化法による手法を用いているため，数式を要素にする行列式の値に本質的でない分数式が発生する．そのため，muMATH 固有の数式変形パラメータを操作する必要があり，かなりの熟練を必要とする．変形に失則すると，因数分解の機能がないため，本質的でない分数式の消去が困難になる．このため処理時間をある程度犠牲にして，たすきがけ法のアルゴリズムを採用した．たとえば4行4列の行列式の値は次のようなものである．(結果3参照)

このような公式を直接プログラムに埋め込むことにより，この算法による時間の消費を節約した．ただ，この式を直接，手で入力したのでは，誤りが発生しやすいので，この公式を前もってプログラムにより発生させ，その結果を直接プログラムに挿入することにより入力による公式の誤りを防いだ．この手法は後述するハイブリッド処理にも利用した．また，この手法は関数データベース（数式公式集）の作成手法として用いられることを注意したい．この SIMPDET を用いて行列の逆行列の演算プログラム SIMPINV を作成した．SIMPDET の例を結果4に示す。

5．4　2階の線型微分方程式の解法パッケージ

このパッケージは

$$p(x)y''+g(x)y'+r(x)y=0$$

の型の微分方程式を解くための専用パッケージで，構成プログラムは汎用目的には作られていない．2階の線型微分方程式は工学において，頻繁に出現する方程式である．この方程式の解析解に関する，数式処理プログラムは文献(2)に詳しい。ここでは，この方程式の解を級数の形で求めることにした．これは後述するハイブリッド処理のためである．与えられた方程式の解を$y=x^p\sum_{i=0}^{\infty}a_i x^i$と仮定して，pと$a_i$を決定すればよい．解を直接，微分すれば，pの決定方程式とa_iに関する恒等式が得られる．pの決定方程式は高々2次であるので容易に解を決定できる．このpでa_iの恒等式を評価し，直接a_iを決定することもできるし，隣接項の関係としてa_iを求めることができる．係数a_iは，実際にはA（I）の形でプログラム中で処理される．この形式は muSIMP における関数表現である．従って，微分方程式の初期条件は，FUNCTION で設定することができる．

これ等の実行例を示す．

(1)微分方程式1（$y''+xy=0$の15項までの解）
　〔結果5参照〕
(2)微分方程式（$y''+x^2y=0$の解の隣接項の関係）
　〔結果6参照〕

5．5　ハイブリッド処理の一方法

数式処理は，数式処理のみに利用される場合もあるが，処理の結果を数値計算用のプログラムに挿入したいこともある．たとえば，グラフ化であったり，公式の利用であったりする．また，他言語が発生した数式を数式処理システムで処理し，その結果をさらに数値処理することが考えられる．このような処理をハイブリッド処理ということにする．

muSIMP ではこの処理は極めて簡単である．汎用性を考慮しなければ，処理結果をテキストファイルとして，接続すべき言語の形式で出力すればよい．次のプログラムは多項式を BASIC の文関数で出力する，簡単な muSIMP プログラムである．

○BASIC コンバータ
　〔結果7参照〕
○コンバータの出力した BASIC の文関数
　〔結果8参照〕

このファイルは BASIC や CP/M の XSUB コマンドにより利用できる．ただし，再帰的な表現（たとえば，微分方程式のa_iの隣接項の関係式）として得られた結果は，BASIC や FORTRAN では特別なプログラムトリックが必要である．この例は文献(3)を参照してほしい．

6．数式処理システムの教育への利用について

今までの数式処理は主として，大型計算機上でしか利

用できなかった．しかし，本稿で述べたパーソナルコンピュータでの数式処理システムでは，費用と使用面における制約から見て，十分に学生個人が，手軽に利用できるシステムである．まだ，数式処理を教育に利用する提案は非常に少ないようである[4]．いわゆる電卓が，多方面から，その長所と短所が議論されたように，このシステムの教育への利用を，十分に議論する必要がある．

数値的な計算ができない学生に，電卓を与えれば，弊害があるように，数式の具体な計算原則が理解できていない学生に，数式処理システムを与えても，意味のないことである．しかし，理工学の教育におけるかなりの部分は，単なる数式の計算であることが多いのも確かである．ある種の教科では，あまりにも途中の計算式が複雑になりすぎるため，現実的な問題を簡単化しすぎて，本質的な議論ができない場合や，それ自身が省略されてしまうことも良くあるのではないだろうか．以下このような問題点を整理してみると

(1) どのような分野の教育に利用すべきか．
(2) どの程度の学生に利用させるべきか．
(3) どの程度の機能を数式処理システムが持つべきか．

等々である．これ等の問題は今後の本システムの開発と並行して，研究していく予定である．

7．むすび

本稿では，パーソナルコンピュータにおける数式処理の可能性とその試験的なプログラムパッケージの内容の概要について述べた．これまで，大型計算機でしか利用できなかった処理が，ある程度，パーソナルコンピュータで処理可能であることが確められた．使用経験から得られた種々の問題点や構想を提案した．それぞれの各部の詳しいプログラムや議論は続報として発表する予定である．

謝辞

日頃，貴重な討論と色々な助言をいただく，電算機室の関係者に感謝します．

参考文献

1）大久保「パソコンによる数式処理の利用」電気四学会中国支部，58年度予稿集
2）渡辺隼郎「常微分方程式の数式処理」教育出版，1974
3）大久保「BASICによる構造化プログラミング」，宇部高専研究報告，58年3月29号
4）対島「数式処理システム「OECU-MATH」の作成」大阪電通大研究論文集，57年18号

(昭和58年9月13日受理)

```
FUNCTION LAPPLY(FN,ARG,A),
WHEN ATOM(FN),
   BLOCK
    WHEN FN=CAR,FIRST(FIRST(ARG)) EXIT,
    WHEN FN=CDR,REST(FIRST(ARG)) EXIT,
    WHEN FN=CONS,ADJOIN(FIRST(ARG),FIRST(REST(ARG)))
 EXIT,
    WHEN FN=ATOM,ATOM(FIRST(ARG)) EXIT,
    WHEN FN='EQ, FIRST(ARG)=FIRST(REST(ARG)) EXIT,
    LAPPLY(LEVAL(FN,A),ARG,A)
   ,ENDBLOCK EXIT

WHEN FIRST(FN)=LAMBDA,LEVAL(FIRST(REST(REST(FN))),LPAIRLIS(FIRST(REST(FN)),ARG,A
)) EXIT,
WHEN FIRST(FN)=LABEL,LAPPLY(FIRST(REST(REST(FN))),ARG,ADJOIN(ADJOIN(FIRST(REST(F
N)),FIRST(REST(REST(FN)))),A))
EXIT,
ENDFUN&
```

図3　muSIMP によるプログラム例 (LAPPLY)

```
?
DET(#[A,B,C,D,E],
      [F,G,H,'I,J],
      [K,L,M,N,O],        ←行列式の値
      [P,Q,R,S,T],
      [U,V,W,X,Y]@);
@:  (S+(R+(Q-B*P/A)*(-W+C*U/A)/(V-B*U/A)-C*P/A)*(-I+D*F/A+(-X+D*U/A)*(-G+B*F/A)
/(V-B*U/A))/(H+(G-B*F/A)*(-W+C*U/A)/(V-B*U/A)-C*F/A)+(Q-B*P/A)*(-X+D*U/A)/(V-B
*U/A)-D*P/A)  *  (H+(G-B*F/A)*(-W+C*U/A)/(V-B*U/A)-C*F/A)  *  (O+(M+(W-C*U/A)*(-L+
B*K/A)/(V-B*U/A)-K*C/A)*(-J+E*F/A+(-Y+E*U/A)*(-G+B*F/A)/(V-B*U/A))/(H+(G-B*F/A
)*(-W+C*U/A)/(V-B*U/A)-C*F/A)+(Y-E*U/A)*(-L+B*K/A)/(V-B*U/A)-K*E/A+(T+(J+(G-B*
F/A)*(-Y+E*U/A)/(V-B*U/A)-E*F/A)*(-R+C*F/A+(-W+C*U/A)*(-Q+B*P/A)/(V-B*U/A))/(H
+(G-B*F/A)*(-W+C*U/A)/(V-B*U/A)-C*F/A)+(Q-B*P/A)*(-Y+E*U/A)/(V-B*U/A)-E*P/A)*(
-N+K*D/A+(-I+D*F/A+(-X+D*U/A)*(-G+B*F/A)/(V-B*U/A))*(-M+K*C/A+(-L+B*K/A)*(-W+C
*U/A)/(V-B*U/A))/(H+(G-B*F/A)*(-W+C*U/A)/(V-B*U/A)-C*F/A)+(-X+D*U/A)*(-L+B*K/A
)/(V-B*U/A))/(S+(R+(Q-B*P/A)*(-W+C*U/A)/(V-B*U/A)-C*P/A)*(-I+D*F/A+(-X+D*U/A)*
(-G+B*F/A)/(V-B*U/A))/(H+(G-B*F/A)*(-W+C*U/A)/(V-B*U/A)-C*F/A)+(Q-B*P/A)*(-X+D
*U/A)/(V-B*U/A)-D*P/A))  *  (-B*U+A*V)
```

↑答

結 果 I

```
? ANS:NEWTON(3,100); ←√3 の100桁計算

@:
1310817325246392572630296847787815196068239387797622729559530026747778869156184
6750797814092446671349965399120933247163847195080445969830364879257 ∠
7568007355894299805193886450591311948360930095488080836400674988432270698789082
5328299050423513353505200456572868750199442585646054154587503218176

?
REALP(ANS,100); ←分数 ANS の小数倍を100桁もとめる。

@: 1 (
7320508075688772935274463415058723669428052538103806280558069794519330169088000
3708114618675724857 56, 100)
```

㊟上の@は $\sqrt{3}$ の答である。下の@は1は整数倍，( ) の前は100桁の小数倍である．

結 果 2

```
WHEN L=4,Z[1,1]*Z[2,2]*Z[3,3]*Z[4,4]-Z[1,1]*Z[2,2]*Z[3,4]*Z[4,3]-Z[1,1]*Z[2,3]*Z
[3,2]*Z[4,4]+Z[1,1]*Z[2,3]*Z[3,4]*Z[4,2]+Z[1,1]*Z[2,4]*Z[3,2]*
Z[4,3]-Z[1,1]*Z[2,4]*Z[3,3]*Z[4,2]-Z[1,2]*Z[2,1]*Z[3,3]*Z[4,4]+Z[1,2]*Z[2,1]*Z[3
,4]*Z[4,3]+Z[1,2]*Z[2,3]*Z[3,1]*Z[4,4]-Z[1,2]*Z[2,3]*
Z[3,4]*Z[4,1]-Z[1,2]*Z[2,4]*Z[3,1]*Z[4,3]+Z[1,2]*Z[2,4]*Z[3,3]*Z[4,1]+Z[1,3]*Z[2
,1]*Z[3,2]*Z[4,4]-Z[1,3]*Z[2,1]*Z[3,4]*Z[4,2]-Z[1,3]*
Z[2,2]*Z[3,1]*Z[4,4]+Z[1,3]*Z[2,2]*Z[3,4]*Z[4,1]+Z[1,3]*Z[2,4]*Z[3,1]*Z[4,2]-Z[1
,3]*Z[2,4]*Z[3,2]*Z[4,1]-Z[1,4]*Z[2,1]*Z[3,2]*Z[4,3]+
Z[1,4]*Z[2,1]*Z[3,3]*Z[4,2]+Z[1,4]*Z[2,2]*Z[3,1]*Z[4,3]-Z[1,4]*Z[2,2]*Z[3,3]*Z[4
,1]-Z[1,4]*Z[2,3]*Z[3,1]*Z[4,2]+Z[1,4]*Z[2,3]*Z[3,2]*
Z[4,1]
,EXIT,
```

結 果 3

```
? Z;

@: #[A, B, C, D, E],
   [F, G, H, I, J],
   [K, L, M, N, O],   ←行列式Z
   [P, Q, R, S, T],
   [U, V, W, X, Y]@

? SIMPDET(Z); ←Zの値

@: A*(I*R*L*Y-I*R*O*V-I*L*W*T+I*W*O*Q-I*M*Y*Q+I*M*T*V-X*R*J*L+X*R*G*O+X*J*M*Q+
X*L*H*T-X*M*G*T-X*H*O*Q+R*J*N*V-R*N*Y*G+J*L*W*S-J*W*N*Q-J*M*S*V-L*S*Y*H+W*N*G*
T-W*S*G*O+M*S*Y*G+N*Y*H*Q-N*H*T*V+S*H*O*V) + K*(-B*I*R*Y+B*I*W*T+B*X*R*J-B*X*H
*T-B*J*W*S+B*S*Y*H+I*R*E*V-I*W*E*Q+I*Y*C*Q-I*C*T*V-X*R*E*G-X*J*C*Q+X*C*G*T+X*E
*H*Q-R*J*D*V+R*Y*D*G+J*W*D*Q+J*S*C*V+W*S*E*G-W*D*G*T-S*Y*C*G-S*E*H*V-Y*D*H*Q+D
*H*T*V) - (B*X*R*F*O-B*X*M*F*T-B*R*N*Y*F+B*W*N*F*T-B*W*S*F*O+B*M*S*Y*F-X*R*L*E
*F+X*L*C*F*T+X*M*E*F*Q-X*C*F*O*Q+R*L*Y*D*F+R*N*E*F*V-R*D*F*O*V+L*W*S*E*F-L*W*D
*F*T-L*S*Y*C*F-W*N*E*F*Q+W*D*F*O*Q-M*S*E*F*V-M*Y*D*F*Q+M*D*F*T*V+N*Y*C*F*Q-N*C
*F*T*V+S*C*F*O*V) - (B*I*W*O*P-B*I*M*Y*P+B*X*J*M*P-B*X*H*O*P-B*J*W*N*P+B*N*Y*H
*P-I*L*W*E*P+I*L*Y*C*P+I*M*E*P*V-I*C*O*P*V-X*J*L*C*P+X*L*E*H*P-X*M*E*G*P+X*C*G
*O*P+J*L*W*D*P-J*M*D*P*V+J*N*C*P*V-L*Y*D*H*P+W*N*E*G*P-W*D*G*O*P+M*Y*D*G*P-N*Y
*C*G*P-N*E*H*P*V+D*H*O*P*V) + U*(B*I*R*O-B*I*M*T-B*R*J*N+B*J*M*S+B*N*H*T-B*S*H
*O-I*R*L*E+I*L*C*T+I*M*E*Q-I*C*O*Q+R*J*L*D+R*N*E*G-R*D*G*O-J*L*S*C-J*M*D*Q+J*N
*C*Q+L*S*E*H-L*D*H*T-M*S*E*G+M*D*G*T-N*C*G*T-N*E*H*Q+S*C*G*O+D*H*O*Q)
```

結 果 4

```
? SIMPEX(1,0,X,A,X,15);

? EX:@;

@: X*(X^2*(X*(X^2*(X*(X^2*(X*(X^2*(X*A(2)/7076160+A(1)/1710720)-A(2)/45360)-A(
1)/12960)+A(2)/504)+A(1)/180)-A(2)/12)-A(1)/6)+A(2)) + A(1)
```

結 果 5

```
? MAKEEQ(1,0,X^2,A,X,K);

@:TOKUSEI HOTEISIKI
-P+P^2=0
TOKUSEIKON IS FOUNDED
P1=1
P2=0
SOLVE P1

2*X^(-2+P+K)*P*K*A(K)-X^(-2+P+K)*P*A(K)-X^(-2+P+K)*K*A(K)+X^(-2+P+K)*P^2*A(K)+
X^(-2+P+K)*K^2*A(K)+X^(2+P+K)*A(K)



A(-3+S)+(1+S)^2*A(1+S)+(S*A(1+S)+A(1+S))
SOLVE P2
A(-2+S)+(2+S)^2*A(2+S)+(-S*A(2+S)-2*A(2+S))


 (+ (A (+ -3 S)) (* (^ (+ 1 S) 2) (A (+ 1 S))) (+ (* S (A (+ 1 S))) (A (+ 1 S))
)) (A(-2+S)+(2+S)^2*A(2+S)+(-S*A(2+S)-2*A(2+S)), 1, 0)


? EX:@;

@: (+ (A (+ -3 S)) (* (^ (+ 1 S) 2) (A (+ 1 S))) (+ (* S (A (+ 1 S))) (A (+ 1
S)))) (A(-2+S)+(2+S)^2*A(2+S)+(-S*A(2+S)-2*A(2+S)), 1, 0)


? FIRST(EX);

@: A(-3+S) + (1+S)^2*A(1+S) + (S*A(1+S)+A(1+S))


? EXPD(@);

@: 3*S*A(1+S) + S^2*A(1+S) + A(-3+S) + 2*A(1+S)


? SECOND(EX);

@: A(-2+S) + (2+S)^2*A(2+S) + (-S*A(2+S)-2*A(2+S))


? EXPD(@);

@: 3*S*A(2+S) + S^2*A(2+S) + A(-2+S) + 2*A(2+S)
```

結 果 6

```
? FUNCTION OUTBAS(L,NAME),
  WRS(NAME,'BAS),
 LINELENGTH(255),
 NEWLINE(),
 PRINT(10),
 SPACES(2),
 PRINT ("DEF FNF(X)= "),
  PRTMATH(L,0),
 NEWLINE(),
  WRS(),
 ENDFUN$
```

結 果 7

```
? OUTBAS(EX,GA);←式 EX を BASIC の関数に変換

Ok
10  DEF FNF(X)= X*(X^2*(X*(X^2*(X*(X^2*(X*(X^2*(X*A(2)/7076160+A(1)/1710720)-A(2
)/45360)-A(1)/12960)+A(2)/504)+A(1)/180)-A(2)/12)-A(1)/6)+A(2))+A(1)
```

結 果 8