

パーソナルコンピュータ用数式処理システムの拡張(2)

大久保 明伸*

Extension of Symbolic Mathematical Operation System Using Personal Computer

Akinobu Ohkubo

Abstract

This report discusses an extension of symbolic manipulation system and an proposition of the language education using this system.

This system has faculties of symbolic manipulation, graphic functions, LISP interpreter, Prolog interpreter and support functions.

1. まえがき

本校における情報処理教育の主要な科目は、FORTRAN, BASIC, 機械語などの言語教育が中心である。しかし、ソフトウェア科学の発達により、電子計算機システムの利用分野は、事務処理や技術計算への利用ばかりでなく、非数値処理の分野に利用されるようになってきた。データベース, CAD/CAM, CAI, ロボット工学, 人工知能機械など、近頃急速に実用化されはじめた分野が多い。これに伴い、ソフトウェア技術者の不足が問題になっている。本校の卒業生の就職先での仕事も、この分野で実務に就くものが多い。ますます、この傾向は急速に進むものと考えられる。このような観点にたつて、電子計算機システムの非数値处理的な利用方法の提案やそれにもとづくプログラミング教育法の提案などを行ってきた。本報告は、muMATHを基礎にし、種々の拡張をおこなったUMATHのまとめと、情報処理教育への利用の提案をおこなう。

2. UMATHの概要

muMATHはmuSIMPで記述された数式処理システムである。muSIMPはLISP系の言語であるが、この入

力プログラム(プリコンパイラ)はFORTRANやPASCALの関数副プログラムに似た形式での入力ができる。入力プログラムにより翻訳されたプログラムやデータはS式になる。このS式がシステムにより評価される。また、出力プログラムはS式を数式表現で表示する。詳しくは文献(2)-(12)を参照してほしい。muMATHの機能をまとめてみると、

- (1) 数式の四則計算と多項式の整理
- (2) 数式の微分と不定積分と定積分
- (3) 極限操作
- (4) 無限桁の整数計算
- (5) 代数方程式の解法
- (6) 行列, 行列式の処理
- (7) その他

である。

問題点は因数分解の機能がないことである。しかし、この点を除けば、ある程度の実用にたえるものである。このシステムを使用しリッカチの方程式を解析した例もある。UMATHはこのmuMATHの機能の上に、高専の工学教育への利用を目的とし、構成したプログラム群のことである。

2-1 UMATH開発の目的

工学教育では、数値計算のために昔から数表や計算尺が広く使用されてきた。それらは現在、電卓にとってか

*宇部工業高等専門学校電気工学科

わった。しかし、工学教育では数式の計算も重要である。数式の計算は学生の手計算で行うのが普通である。そのため演習などで現れる数式は手計算可能な範囲しか出題できないのが現実である。例えば、微分方程式の解法の1つに級数解法があるが、実際に種々の方程式の解を手計算で求めるのは不可能に近い。複雑な関数のテーラ展開も、正確に計算するのは困難である。行列式も5行5列程度になると、大半は計算不可能になる。そのため、これらの計算はほとんどFORTRANやBASICによる数値計算的な演習問題にまわされることになる。解法を学ぶ場合は手計算も必要とおもわれるが、解法を利用する場合は数値計算の道具として電卓があるように、数式処理用電卓が必要である。実際、大型計算やミニコンピュータにおける数式処理システムでは主としてこのような目的に使用されている。さらに、現在パーソナルコンピュータの普及と汎用性の高いOSの普及で、パーソナルな数式処理システムが実現可能な状況にある。このシステムを具体的に使用する効果的な教材の開発が必要である。単なる電卓としても、十分意義のあるシステムとおもわれる。

2-2 UMATHの機能

UMATHは大きく別けて次の3つの機能をmuMATHの上に構成したものである。記述言語はmuSIMPである。図1にこの構成を示す。処理の内容は高専の工学、数学の教科書程度の数式を前提にしている。もちろん、以下の機能は相互に利用可能である。

- (1) 電卓的な機能
 - 微分方程式の級数解法
 - ラプラス変換
 - 形式的な微分
 - 分数なしの行列式の展開
 - 数式の清書
 - BASICステートメントの発生
 - その他
- (2) グラフィック機能
 - カーブプロットへの図形出力
 - グラフィックディスプレイへの図形出力
 - その他
- (3) プログラム入門者のための補助機能
 - 純LISPインタプリタ
 - ミニPROLOGインタプリタ

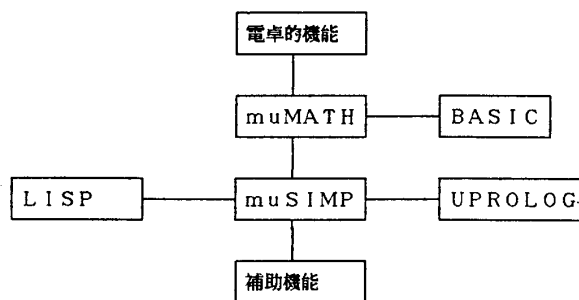


図1 UMATHの構成

プログラミング入門書

その他

このシステムの使用例を図2、図3、図4にしめす。詳しい使用方法是文献を参照してほしい。また、グラフィックスの機能に関しては、付録にまとめておく。

3. 数式処理システムの教育への導入

現在、本校の情報処理関係の教育は、大きく別けて、汎用ミニコンピュータによるFORTRAN教育とパーソナルコンピュータによるBASIC教育である。また、一部の学科ではパーソナルコンピュータによる機械語教育がおこなわれている。FORTRANやBASICを基礎にしたプログラミング教育では、例題や演習問題は、言語の性質からどうしても数値計算的なものにならざるをえない。また、学生が情報処理以外の科目で計算機を利用する場合でも数値計算的な利用にかぎられる。また、いずれの場合も、数値計算的なプログラミングから得られるプログラムの概念や計算機に対する概念は、ごく限られたものになる危険性がある。

一方、現在の計算機の使用分野は広範囲にわたっている。特に、人工知能的な分野に使用するため種々の新しい言語が開発されている。人工知能に対する考え方は、別にして、この分野で用いる言語やアルゴリズムの発想は数値計算的なプログラミングでは、なかなか得られないものである。数値計算に対して、ここでは非数値計算とか、記号処理とよぶことにする。FORTRANやBASICから記号処理の原理へ学生を導くためには、PASCALが適当である。この議論は文献を参照してほしい。手続き型の言語であるPASCALは、FORTRANやBASICを習得した学生には、比較的容易に習得可能な言語とおもわれる。しかし、数式処理システムを使うことで、より具体的な記号処理を体験したり、プログラム技法を習得でき

```

? LAP(1+T^2+7*SIN(T));
@: 2/S^3 + 1/S + 1/(1/7+S^2/7)

? _LAP(U(T));
@: 1 / S

? LAP(D(T));
@: 1

? LAP(SIN(T+#PI/4));
@: 1/(2^(1/2)*S+2^(1/2)/S) + 1/(2^(1/2)*S^2+2^(1/2))

? X;
@: (B*M/C+A*M)*(P/O+Q/O) / ((G+H/(I+K))*(F*L+L*E))

? X&
@: (* (^ (+ G (* H (^ (+ I K) -1))) -1) (+ (* B (^ C -1) M) (* A M))
(^ (+ (* F L) (* L E)) -1) (+ (* (^ O -1) P) (* (^ O -1) Q)))

? PRETTY(X)$
      B * M          P      Q
  (----- + A * M) * (--- + ---)
      C              O      O
-----
      H
  (G + -----) * (F * L + L * E)
      I + K
    
```

図2 UMATHの電卓的な実行例

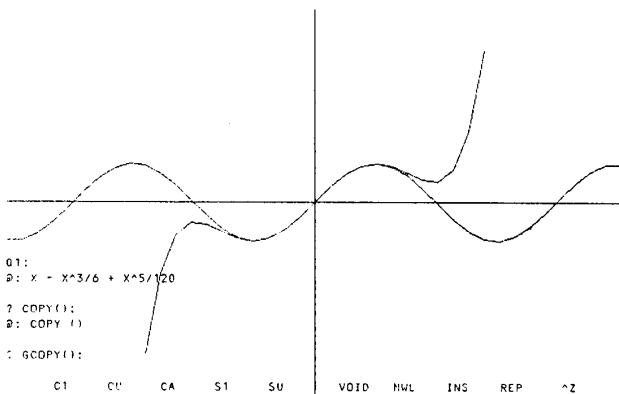


図3 図形表示の例

```

(ASSERT (LIKES TOM MARY))
(ASSERT (LIKES MARY CAT))
(ASSERT (LIKES *X *Y))
(LIKES TOM MARY) *S
(LIKES TOM CAT) *S
(LIKES CAT APPLE) *S
(AND ((EVAL (INT (SIN X) X) *ANS)
(EVAL (DIF *ANS X) *TEST)
(EQ *ANS *TEST))) *S
    
```

図4 PROLOGによる実行例

る利点がある。

以下、卒業研究生と一部の学生に対し数式処理システムの講義や演習を行った経験から得られた結果を報告する。

3-1 数式処理システムの電卓的利用

まず、数式処理を使用した経験がない学生に対しては簡単な多項式の整理や展開を実施してみせる。さらに、手計算では不可能な例を実行させて見せる。特に、教科書の例題にある微分や積分を処理させると、このシステムに興味をもつ学生が多い。少しおおきめな行列式の展開や微分方程式の解などを示すと、さらに興味をもつ。このような電卓的な機能を利用して、応用数学の例題や

電気回路などの例題を構成すれば、より現実的な問題を提示する事が可能である。このような使い方限定すれば、2時間から3時間程度の講義(主として、数式の入力方法とプリント方法)で十分である。学生はこのような利用を通して、数式の処理もまたプログラム可能であることが体験できる。ハイブリッド処理関数で得られた数式をグラフ化したり、数値計算のプログラムへ数式を渡し、それを実行したりして、より高度の利用方法も体験できる。単なる記号処理用の言語では、これらの機能(関数)はすべて、教官が製作しなければならないが、muMATH+UMATHを使用すれば、負担はかなり軽減できる。このように、数式処理電卓を利用することで、新しい発想に元づく教育方法が開発可能である。問題点は、数値電卓の普及により分数計算ができない学生が現れているとい

う説があるように、数式の計算が自由にできない学生が現れる心配がある。手計算は計算の方法を習得する一方法である。従って、どの程度の学生にこのシステムを使用するかは問題である。しかし、次に述べる数式処理のプログラムを製作させることで、このような問題点のほとんどは克服される。

3-2 数式処理のプログラミング教育

数式処理のプログラムは muSIMP で記述する。muSIMP は LISP 系の言語で、FORTRAN や BASIC を先に学習したものには、理解が困難なようである。本質的なステートメントは10個程度であるが、プログラムの取り扱うデータの構造とプログラムにあらわれる再帰関数は初めての学生が多い。特に、LISP の EVAL 関数や APPLY 関数の理解は、本格的なプログラムを作るのには、極めて重要である。このため、どうしても、LISP の原理を理解する必要がある。そのために、muSIMP の上で動作する、ごく原理的な LISP 処理系を使用する。また、人工知能的な処理系といわれる PROLOG も関数としてくみこまれているので、より高度な処理も可能である。

数式処理のプログラムを製作することで、次のような利点がある。

- (1) 記号処理の具体的な応用を体験できる。
- (2) 数式の処理アルゴリズムを深く理解できる。
- (3) 手続き型言語では得られないプログラム技法を習得できる。
- (4) FORTRAN 流の考え方で、製作していたプログラムを別の発想で作ることができる。

などである。(1)は、LISP でもよいという意見もあるが、muSIMP の文法の親しみやすさや、基本的に数式処理能力を有するシステムを使用すべきである。逆に、muSIMP で LISP の原理は理解できる。(2)は、先に述べた電卓的な使用より、さらに深く処理(計算)の原理を理解できる。例えば、微分方程式の級数解法などは、実際に人手で計算することは困難であろう。また、解析解の方法も、プログラムできることに気がつくことになろう。数値的な解法とあわせて、より深い微分方程式の理解が得られるはずである。

(3)では、プログラムに対する概念の拡張が期待できる。データの属性や構造をより深く理解することになる。また、再帰プログラムは学生の理解が得にくい技法であるが、数式処理のプログラムを通して、自然に習得できる。

(4)では、卒業研究などで処理する問題のなかには、図形や記号列の処理をしなければならないもののがかなりある。これらの問題を処理するためには、一度は記号処理を体験させておいたほうがよい。学生は FORTRAN や BASIC でアルゴリズムやデータの構造を考える癖がついている。このため、自由な発想をさまたげているように思われる。

4. むすび

工学教育に使用する目的で、既存の数式処理システム muMATH の上に、UMATH を開発してきた。これを具体的な教科で使用するためには、各担当者が専用の機能と教材を作らなければならない。また、どの学年から使用するかは、慎重に検討する必要がある。

また、プログラム教育用の言語として、何を教えるかは、現在再検討の時期にきている。現実的な側面から考えて、当分は FORTRAN の教育は必要であろう。しかし、将来 FORTRAN 以外の言語を教育しなければならない状況にもある。そのなかには、ハードウェアよりの言語である記号言語やC言語のようなものもある。また、人工知能向きの言語である PROLOG, LISP, LOGO 等も実用になりつつある。この方面で仕事を卒業生も増加するものと考えられる。muSIMP/muMATH+UMATH に後者の言語群の概念を包含させることができる。

世の中のニーズにあった学生をつくるために、今後、慎重に検討しなければならない問題である。

UMATH 自身は今後知的な CAI システムへ発展させていく方針である。

5. 参考文献

- (1) muSIMP/muMATH は CP/M または MSDOS で動作する数式システムである。The Soft Warehouse に所有権がある。
- (2) 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
- (3) 大久保：数式処理システムの電卓的な利用法について, 高等専門学校 情報処理教育研究協議会, 58年 8月 3号
- (4) 大久保：パソコンによる数式処理システムの利用, 電気四学会中国支部連合大会講演論文集, 58年
- (5) 大久保：パソコンによる数式処理システムの利用(2),

電気四学会中国支部連合大会講演論文集, 59年

- (6) 大久保: パーソナルコンピュータによる数式処理, 宇部高専研究報告, 59年3月30号
- (7) 大久保: パーソナルコンピュータ用数式処理システムの改善について, 情報処理学会大28回全国大会講演論文集, 59年
- (8) 大久保: パソコンによる数式処理システムの利用(3), 電気四学会中国支部連合大会講演論文集, 60年
- (9) 大久保: パーソナルコンピュータ用数式処理システム (UMATH) の拡張, 宇部高専研究報告, 60年3月31号
- (10) 大久保: パーソナルコンピュータによる数式処理(その1), 数式処理通信 (Vol. 1—No.3) サイエティスト社
- (11) 大久保: パーソナルコンピュータによる数式処理(その2), 数式処理通信 (Vol. 1—No.4) サイエティスト社
- (12) 大久保: muSIMP/muMATH の使い方, 和文解説書 (未発表, 配付可能), 60年

付 録 UMATH の基本グラフィック関数

muSIMP は本来, グラフィック機能はない。数式処理システムにもグラフィック機能をもたせることは, 他の言語と同様に必要である。学生が, muSIMP によるプログラムを製作する場合, 再帰処理の概念を習得しておかなくてはならない。再帰関数の例題は, 図形処理を例題にすれば, 理解しやすい。また, 記号処理と図形処理の関係を示す例題を提示することができる。

数式処理により得られた数式をグラフ化することで, より理解を深めることができる。muSIMP によるグラフィックの方法は次のものが考えられる。

- (1) 出力関数によるカーブプロッタやRS232CのESCシーケンスによる出力装置へのグラフィック
- (2) EXECUTE 関数による機械語プログラムの直接実行
- (3) インタプリンタへの機械語関数のうめこみ。などである。

(1)に元ずいた関数はすでに発表した。(2)は単にグラフを

関数名	引数	対応するBASIC文
GINT	0	L I Oの初期化、必ず実行すること
GSCREEN	4	SCREEN
GVIEW	6	VIEW
GCOLOR1	4	COLOR
GCOLOR2	2	COLOR= (バレット#, カラー#)
GCLS	0	CLS
GPSET	3	PSET/PRESET
GLINE	9	LINE
GCIRCLE	10	CIRCLE
GPAINT1	6	PAINT (色でぬりつぶす)
GPAINT2	10	PAINT (タイルパターンでぬりつぶす)
GGET	7	GET
GPUT1	9	画像のPUT
GPUT2	7	漢字のPUT
GROLL	1	ROLL
GPOINT	4	POINT
GCOPY	6	画面の指定領域のドット状態を指定域に複写

表1 基本グラフィック関数

描く場合は便利だが、データのやりとりが困難である。そこで基本グラフィック関数は(3)の方法でおこなった。

日本電気社の PC9801 の ROM BASIC はグラフィック命令をもっている。MS-DOS の動作中でも、ROM 中のグラフィックルーチンは使用可能である。基本グラフィックルーチンは INT AOH—INT AFH と INT CEH で呼ぶことになっている。MSDOS で使用するには、セグメント F990H の先頭から格納されているベクトルを muSIMP 用のグラフィックルーチンへ移動しておく必要がある。同時に、このルーチンは 1300H の作業エリアを必要とする。

以上の機能を利用する処理プログラムは、MSDOS のアセンブラ (MASM) をで記述した。このプログラムを GRLIO.COM と呼ぶことにする。GRLIO.COM は常駐プログラムである。

muSIMP は 200H 番地から 22DH 番地割り込み形式の

エントリポイントをもっている。GRLIO.COM の各処理ルーチンを、これに割りあてる。

muSIMP の関数として定義する手順は、

(1) GRLIO.COM を実行する。

注) 割り込みベクタの設定と作業番地の初期化を行う。

(2) muSIMP を実行する。

(3) muSIMP によるグラフィック関数の定義プログラム GRSET.MUS を実行する。

注) GRLIO.COM のルーチンと muSIMP のアトムを接続する。

である。以上の手順で表 1 の関数が muSIMP で使用可能である。

(昭和61年10月6日受理)