

## 解説



### ユーザの立場からみた数式処理システム†

仁 木 直 人†

数式処理システムを導入しようとするとき避けては通れない、各システムの経済的側面をも加えた評価を試みる。また、ユーザからみた使いやすい数式処理システムはどのようなものか考察を行う。

#### 1. はじめに

各分野の研究または業務を遂行する上で、数式処理システムの使用が大きな手助けとなることは多い。しかし、実際にシステムを使用してみると、不便・不満な点がまだまだ多いことも事実である。また、これから使用してみたいと思う人に対して、数式処理システムに関する情報があまりにも少ないように感じる（情報誌<sup>1)</sup>の存在は貴重である）。このような認識から、ユーザとして関心の深い使用上の諸問題について述べることに、潜在的ユーザ層の持っているであろう疑問に答え、またシステム開発者への提言を行いたいと思う。

なお、表題の『ユーザの立場から見た数式処理システム』に関する論評は、他の稿においてもいくつかなされている。本稿では、それらとオーバーラップしない形で、筆者の考えを述べてみたい。

ただし、はじめに断っておきたいことがある。ここでは、その目的上、各システムに関して各方面から評価を行っている。これらの評価は、筆者の必ずしも豊富とはいえない知識と使用経験に基づいたものであり、当然異論があるであろう（特にその開発者・販売代理店から）。また、システムの改良などにより、評価がまったく逆転する事態も起りかねない。

#### 2. 数式処理システムの総合的評価

現在、金銭的問題を問わなければ、日本でだれでもが手にいれることのできる数式処理システムは、

MACSYMA<sup>2)</sup>, REDUCE<sup>3)</sup>, SMP<sup>4)</sup>, muMATH<sup>5)</sup>の4種である。また、IBMのSCRATCHPAD<sup>6)</sup>も一般への公開の兆しが見えてきており、国産の汎用システムであるGAL<sup>7)</sup>も公開準備中と聞く。その他、『つて』を辿れば入手可能なシステムや特定のサイトで使用可能なシステムも数多く存在する（たとえば、CAMAL<sup>8)</sup>, MAPLEなど）が、ここでは上記の4種以外は考慮しない。

REDUCEは、500ドルの手数料のみで、RAND Corp.のA. C. Hearnから配布を受けられる（機種によって異なることがある）。他の3種は日本にも代理店があり、相応の値段で買入れることができる。それぞれ

MACSYMA ¥3,000,000~5,600,000(500,000)

SMP ¥3,400,000(700,000)

muMATH ¥70,000~100,000

程度である。括弧の中は大学・研究機関が非営利的な目的で使用する場合の価格である。ビジネス用ソフトウェアやCAD用ソフトウェアなどと比べて、その機能や応用の広さを考えても、決して高価なものではない。しかし、4種の内、どれが『高い』か『安い』かは価格だけでは判断できない。

##### 2.1 MACSYMA

現存する数式処理システムの中で、MACSYMAは自他ともに認める最も強力なシステムである。最も優秀な人材の100年人の労力で開発したというだけのことにはある。筆者もぜひ手元で使いたいと思うシステムであるが、実現していない。それはなぜか？

『高い』からである。システムすなわちソフトウェアの値段が高いといっているのではない。筆者の勤務先では、括弧内の大幅な割引価格で導入できるのであるから、そう手の届かないものではない。MACSYMAが稼働できる計算機の問題である。

MACSYMAが走る主な計算機にDECのVAX-11シリーズおよびSymbolics 3600がある。VAX-11はいわゆるスーパーミニコンであり、高性能な機種は1

† A Survey of Computer Algebra Systems from User's Standpoint by Naoto NIKI (The Institute of Statistical Mathematics).

†† 統計数理研究所

億円を大きく超える。ところが、そのような上位機種を使っても、ふたりが MACSYMA (その VAX 版である Vaxima) を使いだすと他のユーザのジョブに多大な影響を与えるため、結局は先に始めている Vaxima 使用者が終るまで待たなければならないと聞いている。かといってなんとも VAX-11 を導入するわけにはなかなかいかない。ただ、近日中に DEC 社の計算機としては大幅にコストパフォーマンスが向上した MicroVAX-2 が市場に出るので、これを個人用に使うことにより『高嶺の花』を手に入れるユーザが出てくるであろう。

もうひとつの Symbolics 3600 に MACSYMA を載せた構成は評判が良いようである。問題は Symbolics 3600 が LISP マシンであることに尽きる。人工知能や知識工学が盛んになってきたとはいえ、まだまだ LISP マシンの需要は汎用計算機のそれと比べてはるかに小さいといわざるを得ない。他にめばしい用途がない場合、MACSYMA だけのために 2000 万円近いまたはそれ以上する計算機を購入するのは難しいことである。また、数式処理はそこだけでクローズしたのではなく、得られた結果を用いて大規模な数値計算を行い、そのまた結果を用いて作図を行い、その結果いかんではまた数式処理に戻り…などという具合に実際の処理が進むのであるから、LISP マシン単独では全体の処理を完了できない場合が多いと思われる。残りの処理を他の汎用計算機に委ねることになれば、数式を計算機間で受け渡す手間の面とともに、経済的観点からも不利といわざるを得ない。

メインフレーム (=IBM) 版の MACSYMA がまず第一に望まれるものである (うわさは数年前からあるが、進展をみない)。また、最近発達が著しいワークステーション (特に CPU に MC 68020 を用いた UNIX マシン) で MACSYMA が走れば、ユーザ数が一気に拡大することは疑いないであろう (『計画中』となって久しい)。

## 2.2 REDUCE

REDUCE は世界的に最も広く使われている数式処理システムである。サイト数は 1000 を遙かに越え、非公式なクローンをいれと 10000 に近いとさえいわれている。メインフレームからパソコンまで『LISP の載る計算機であれば REDUCE が走る』というような景観を呈している。その第一の理由はソース・プログラムの公開にある。

たとえば IBM 版であれば、現在のところ、誰でも

500 ドルの手数料だけで、REDUCE のソース・プログラムとそのマニュアル類、さらには Standard LISP<sup>9)</sup> のアセンブラ・ソース・プログラムまでが入った磁気テープを手に入れることができる (機種によっては入手先や条件が異なる)。REDUCE のソースは、RLISP という parser が可能にした、LISP とは思えないような PL/I 的な記述法で書かれている。Standard LISP がもともと多くの LISP システムの最大公約数的な性格を意図したものである上、言語上・機能上の相違を RLISP に押し込むことが可能であるから、各種の LISP 上に REDUCE システムが移植されたのは、むしろ当然である。

したがって、多くのユーザはそのための計算機を新たに購入することなく、ほとんどタダ同然 (少し前までは 500 ドルですらなく 200 ドルであった) で、本格的な数式処理の恩恵にあずかってきた。しかも、メインフレームを使用する場合には、その強力な OS の制御の元にある他のジョブに迷惑がかかることも少なく、すいているときには高速処理を実現できた。

主な問題点は、MACSYMA に比べて機能が足りないことと、『売りもの』ではないせいかシステム中のバグに注意が必要な点である (これらの話題は他稿にゆずる)。この両者とも、ソースが公開されていることから、自分で新機能の組み込みやバグの修正ができるという救いがあるが、他人の作成した大規模なシステムの変更は必ずしも簡単ではない。Hearn に連絡すれば相談にのってくれるが、相手の忙しいこともあり、緊急を要しないような話題では半年以上待たねばならないであろう。REDUCE ユーザ間の情報交換の組織化が当面急がれよう (その試みとして参考文献 10) がある)。

小さな問題かもしれないが、経済的な問題として、REDUCE が『売りもの』でないことがある。筆者のような国立研究機関や国公立大学などに所属する者は、『手数料を公費で外国に前払い』するのは困難なことが多い。そこでついポケットからとなってしまうが、バージョン・アップの度の出費で参る。現在のそのような公開主義を変えずに『売ってくれたら』と心底思う。

なお、CP/M-68K を OS とするパソコン用のシステムは、(株)ビー・ユー・ジー (札幌)、Metacomco 社 (英) などから販売されていることを付け加えておく。価格は 7.5~12.8 万円と聞いている (他に LISP を購入する予算が必要)。

### 2.3 SMP

数式処理システムを記述する言語としては FORTRAN と LISP が多いが、他にも BCPL, PROLOG, APL2 など使われている (たとえば, 参考文献 8), 11), 12)). SMP は C で書かれていることを『売りもの』としている。

C は UNIX マシンの主要言語であるから、稼動する計算機の範囲は広い。また、数値計算やグラフィック表示が容易に効率よく行えることも想像に難くない。しかし、せっかく C でシステムが記述されているにもかかわらず、そのソース・プログラムが公開されているわけでもないし、ユーザによる特殊機能の組み込みも難しいようである。もっとも供給元では『MACSYMA や REDUCE では LISP の知識が必要になることがあるが、SMP では C およびシステムに関する知識を要しない』と、逆に特長のひとつとしている。この辺のニュアンスの違いは使用経験のある程度積まないと判断ができない。

販売を開始したのが 1984 年と遅く、その前から特に注目されていたわけでもないのに、ユーザ・サイドからは SMP に関して未知な部分が多い。いわば『どの馬の骨ともつかない』システムである。日本国内で販売活動が始まったのはつい最近であり、筆者のところでも、Apollo 用システムの入ったフロッピー・ディスク 5 枚が届いたばかりで、まだ install していない (それでも研究機関では日本で最初のグループに入るらしい)。バグがかなりあることは半分覚悟しており、なにか問題が生じたときの相談相手は当然ごく限られているであろう。『高い買物』をしたかどうかは、とにかく使ってみなくては分からない。

最近 MACSYMA と SMP の説明会があいついで開催された。特に SMP の会場は、これまで数式処理システムを使ったことのない人たちが多数集まり、入場を断るほどの盛況で、数式処理への関心の高さがうかがい知れた。対照的に、MACSYMA の会場はなんらかの数式処理システムの使用経験者が多くを占めた (したがって多人数ではない) ようである。興味あるのは、MACSYMA 側のエイジェントが SMP をごく低く評価したことである (現存する数式処理システム中、第 1 位は MACSYMA、第 2 位は REDUCE、…とした上で、SMP は辛うじて第 100 位だそうである) 逆に SMP 側は、MACSYMA の欠点を克服するためにこのシステムを開発したことを強調し、明言こそしなかったが、すでに MACSYMA の域を越えた、も

しくは、近い将来越えるであろうことを印象づけようとしていた。

以上のような現状で、まったく評価を下しようがない。ただ、ひとついえることは、完成されたシステムとして MACSYMA が新たな発展の努力をあまり行っていないのに対し、他のシステム (特に SMP) ではその意欲が旺盛な点である。いつまでも MACSYMA が横綱然としていられるか、これまた分からないところである。

### 2.4 muMATH

単純な構成ながら、muMATH は『なかなか良くできたシステム』である。また、LISP の一種である muSIMP で書かれたソース・プログラム・モジュールの集まりとして提供されるので、中身を調べたり、改造したり、なにかと便利である。新しい機能の追加も比較的容易であり、数式処理システムの教育・実験には最適であろう。

しかし、なんといっても搭載される計算機が CP/M または MSDOS を OS としたマイクロ・コンピュータであり、muMATH および muSIMP も高速性を目指した構成をしていないため、数式処理システムとしての本格的な実用性を期待するには無理がある。一方、『数式計算のできる電卓』と考えると大変便利なものであり、しばしば必要となる『手でも可能だが煩わしい』式の計算には大きな手助けとなる。経済的事情からマイクロ・コンピュータ以外は使いにくいユーザにとっては、『お買い得』なソフトウェアのひとつと思われる。

最近、冗談半分本気半分で『muSIMP をメインフレームに移植しよう』という議論がある。処理速度さえ補えば、muMATH は魅力あるシステムである。

## 3. ユーザからみて使いやすいシステムとは

より良い数式処理システムという概念に関しては、システム開発者からの見方とユーザからの要望とは少しずれているかもしれない。ここでは、いくつかの話題を取りあげて、その微妙な違いを考えたい。

### 3.1 式の展開に関して

式の展開に関して 2 通りの基本方針がある。ひとつは、できるだけ展開や通分などを自動的に行い、結果を単一の有理式とする方針で、REDUCE がこれを採用している。もうひとつは、展開や通分などはユーザが指示したときのみに行い、不要な展開などを避けようとする方針で、MACSYMA および SMP はこちら

である。

どちらの方針も一長一短があり、簡単に軍配を挙げかねる。細かい議論は別として、要は『展開や通分をしなければ、結果が簡単な式になったときに分からない』という主張と『展開や通分は簡単であるが、因数分解などの逆演算に要する手間は比較にならない程大変である』という主張の違いである。

どちらの主張を重要とみるかは、好みの問題や応用分野により違いがあろう。しかし、実際に使う身としてどちらの方針が案かといえば、前者である。

前者の方針下では、入力した式はドンドン展開されるということだけに注意していればよい（メモリ使用量や計算量の増大も考慮に入れる必要がある）。展開されたくない部分は適当なシンボルまたはオペレータ表示で代用しておき、必要に応じて、局所的に代入を行えば済んでしまう。また、有理式の分母は、多くの場合、元となった有理式の分母の因数多項式の積（典型的には最小公倍多項式）であろうから、改めて因数分解する必要は少ない。

や余談になるが、結果を常に簡単化（長さは無視する）して眺めていると、偶然、新しい知見を得ることがある。たとえば、多項式の各項の特徴から、全体がある関数の展開形になっていることを発見するかもしれない。処理の隅々まで管理を行うことは、効率が良いかもしれないが、『機械任せ』の部分からの『思わぬ副産物』は期待薄である（期待するほうが図々しいが）。

後者の方針は、数式処理システムを『紙と鉛筆』の代用物として使おうとするユーザにとっては自然なものであるかもしれない。また、計算機にかかる負担は一般に軽いと思われる。しかし、その趣旨を忠実に守ろうとすれば、1ステップごとに、いまある式をどう加工すれば後の処理に最も都合が良いかを考え、式の特定部分の指定を伴う適当な関数呼び出しを行うことになり、さぞかし疲れることであろう。またあらかじめプログラムを組んでおき、その指示通りに実行させるバッチ的処理（会話処理に入る前に既知の処理を行わせることがよくある）は難しいと思われる。

開発中の国産システム GAL では、両者の方針の中間を狙い、『完全に』ではなく『適度に展開する』モードを標準とするという。その趣旨はユーザとして大いに歓迎するが、使うほうの意図とうまく合うかどうか興味津々である。

### 3.2 擬変数について

式の展開が思い通りにならないことも含め、ミスの修正、特殊機能の実現、入力キータッチの節減などのために、PL/I の『擬変数 (pseudo-variable)』または Common LISP<sup>13)</sup> の『汎変数 (generalized variable)』などに相当する概念が使えると大変便利と思えるが、実現は困難なのであろうか？

たとえば、REDUCE で

```
NUM (A):=SUB (BC=B+C, NUM(A))+1;
```

のように書いて、分子の部分だけを変更するなどの操作ができれば、その恩恵は非常に大きいと思われる。この形式ではあるパラメータに対する関数またはオペレータ値あるいは計算式の定義と区別できない(?) ならば、

```
SETF(NUM(A), SUB(BC=B+C, NUM(A))+1);
```

のような関数形式でも我慢はできる。REDUCE では単に (RPLACA A (AEVAL...)) を生成すれば良いし、他のシステムの場合もそう難しいことではないように思えるが、システム開発者の意見を聞いてみたい。

### 3.3 Pretty Print について

結果を見やすい形でユーザに呈示することは、数式処理システムに課せられた必要条件である。『美しい出力 (Pretty Print)』の能力を持つことはほぼ常識になりつつあるが、まだまだ問題点が多い。

ひとつは、ビットマップ・ディスプレイあるいはグラフィック・ディスプレイを備えた計算機が多数を占める情勢になりつつあるにもかかわらず、その表示能力を生かした Pretty Printer を持つシステムが少ないことである。もちろん、計算機により異なるドライバを用意しなければならないなど難しい面があることは理解できるが、少なくとも『売りもの』のシステムは『本当の Pretty Printer』を備えるべきであろう。

特に、ビットマップ・ディスプレイを持つワークステーションで動くシステムでは、人間が見やすいだけでなく、画面上で修正した式が再入力できるような工夫が欲しいように思う。画面編集機能を持つ端末（実際はマイクロ・コンピュータであるが）を用い、入力と同一の形式 (REDUCE では OFF NAT 指定) で出力したものを編集・再入力するだけでずいぶん便利なおものである。そこに『見やすさ』と『修正の容易さ』『修正ミスの少なさ』が加わればいいことがない。

数式処理で求めた結果を発表する場合、別の問題が生じる。最近では学術雑誌でも（そのまま写真製版できる）camera-ready 原稿を要求するものが増えてきて

いる。多くの人にとっては面倒なこの事態も、数式処理ユーザにとっては歓迎すべき動向といえる。そのような雑誌では、少々長い式でも載せてくれることが多いからである。

問題は、その雑誌に合わせた形式で、かつ、ギリシア文字や添字つき変数などが含まれる表現を、きれいな字体（場合によっては数種のフォントの）で印刷を行う必要が生じることである。すなわち、システム外の『Pretty Printer』（たとえば TROFF<sup>14)</sup>）に数式の清書を委ねることになるのであるが、そのための考慮はほとんどされていないのが現状である。

筆者は、ON LIST, OFF NAT 指定の下で行った REDUCE 出力を元に、エディタや簡単な変換プログラムを用いて（典型的な手順はマクロ化してある）清書プログラムの数式表現形式に変換しているが、かなりの手間を要している（もちろん出力結果を見ながら打ち直すよりは遙かに楽である）。式の一部を囲うためのカッコと関数またはオペレータのパラメータを表すカッコを区別して出力してくれるだけでも、大きく手間が省けるはずである。また、行を変えるときはできるだけ演算子の前でを行うとか、ベキの部分をはなんらかの記号で囲うとか、簡単な改良で変換作業は大幅に容易になるはずである。

### 3.4 数値計算について

数値計算を効率よく行うため、FORTRAN あるいは C のプログラムとして数式処理の結果を出力することに関しては、他稿に議論があり、ここでは述べないが、ひとつだけ疑問がある。『LISP 上に構築されたシステム (MACSYMA, REDUCE など) が、なぜ LISP で数値計算するためのプログラムを出力しないのか』ということである。最近の LISP は数値計算能力が高く、多倍長整数や有理数を（場合によっては多倍精度浮動小数点数をも）扱える利点は捨てがたい。内部表現を少し加工するだけで済むようなものであるので、システム開発者が組み込む気になるかどうかの問題であろう。

## 4. おわりに

以上、あまり取り上げられない話題を拾い集めて議論したため、とりとめのめないことになってしまったのは勘弁願いたい。

数式処理の発展はユーザ層の厚みを増すことであると信じて、筆者はなん回かシンポジウムの世話役を買ってでたが、回を重ねるにつれ、これまで数式処理と

あまり縁のなかった分野の方の出席や問い合わせも目立つようになってきた。本稿は、その目的の一助になればとの思いで書いたものであるし、また、システム開発に携わる方々に対する声援のつもりでもある。

なお、本稿では多くの商標を使用した。ここに一覧の形で呈示させていただく。

|                         |                         |
|-------------------------|-------------------------|
| Apollo                  | Appolo Computer, Inc.   |
| CP/M, CP/M-68K          | Digital Research, Inc.  |
| IBM                     | I B M                   |
| MACSYMA, Symbolics 3600 | Symbolics, Inc.         |
| MC 68020                | Motorola, Inc.          |
| MSDOS                   | Microsoft Corp.         |
| muMATH, muSIMP          | The Soft Warehouse      |
| SMP                     | Inference Corp.         |
| UNIX                    | Bell Laboratories       |
| VAX-11, MicroVAX II     | Digital Equipment Corp. |

## 参 考 文 献

- 1) 「数式処理通信」第1号 (Vol. 1, No. 1, 1983) ~ 第11号 (Vol. 3, No. 3, 1985).
- 2) MATHLAB Group: MACSYMA Reference Manual, MIT, Mass. (1977).  
MATHLAB Group: A Brief Overview of MACSYMA, MIT Computer Sci. Lab., Mass. (1983).
- 3) Hearn, A. C. (ed.): REDUCE User's Manual, Ver. 3.2, Rand Corp., Santa Monica (1985).
- 4) Wolfram, S.: SMP Reference Manual, Ver. 1, Inference Corp., Los Angeles (1983).
- 5) The Soft Warehouse: The muMATH/muSIMP-80 Symbolic Mathematics System Reference Manual for the CP/M Version, Microsoft Corp., Bellevue, Wash. (1981).
- 6) Jenks, J. D.: A Primer: 11 Keys to New SCRATCHPAD, EUROSAM 84 (Lecture Notes in Computer Science 174), Springer, Berlin, pp. 123-147 (1984).
- 7) 佐々木建昭, 元吉文男: 国産数式処理システム GAL の開発, 数式処理通信, Vol. 3, No. 2, pp. 16-21 (1985).
- 8) Fitch, J. P.: CAMAL User's Manual, Univ. of Cambridge (1975).
- 9) Marti, J. B., Hearn, A. C., Griss, M. L. and Griss, C.: Standard LISP Report, Univ. of Utah, Salt Lake City (1978).
- 10) 文部省科学研究費「数式処理の学際的应用への総合的研究」研究班 (編集): REDUCE プログラミング資料第1集, 第2集, 第3集 (予定), 東京

- 大学大型計算機センタ研究開発部 (1984, 1985, 1986).
- 11) 永田守男 : 数式処理における Prolog, 情報処理, Vol. 25, No. 12, pp. 1319-1328 (1984).
- 12) Yasui, Kiyotaka : On Representation and Manipulation of Mathematical Formulas Using General Array Data Structure -- IFORP -- APL84, ACM (APL Journal, Vol. 2, No. 2, pp. 17-22 に転載されている) (1984).
- 13) Steele, G. L., Jr., Fahlman, S. E., Gabriel, P., Moon, D. A. and Weinreb, D. L. : Common Lisp, Digital Equipment Corp., Bedford, Mass. (1984).
- 14) Tuthill, B. : Typesetting on the Unix system, BYTE, Vol. 8, No. 10, pp. 253-262 (1983).

(昭和 61 年 3 月 3 日受付)

---