解 説

数式処理の歴史と将来の展望†

元 吉 文 男†† 佐々木 建 昭†††

1. はじめに

電子計算機で数値を扱い種々の計算ができるようになれば，次には数式そのものを扱おうと考えるのはごく自然である．実際，数式処理が研究され始めたのはかなり古く，1950年代の初期にすでに研究らしきものが記録されている．しかし，人工知能と応用の両分野で本格的に研究が始まったのは1959～60年頃といってよいであろう．爾来，アルゴリズムにシステムに，数式処理は長足の進歩をとげてきた．アルゴリズムでは，Hensel 法による因数分解，Risch の初等関数の不定積分法，Buchberger の Groebner 基底の構成法などは記念碑的であり，単に計算機科学のみならず数学においてもきわめて重要な意味を持つ．システムでは，簡単さとエレガンスを兼ね備えた REDUCE は中型ながら名システムであり，また MACSYMA は非常に幅広い数学知識を集大成した巨大な数学エキスパートシステムといえる．

現在では数式処理は，単にいくつかの応用分野で不可欠な道具としての重要性のみならず，情報科学の確固とした一分野としての地位を得てきた．これは，構成的数学の計算機による実際的実行とその利用を目的としてきた数式処理としては，相応な評価といえる．その評価の上に，数式処理はより広い範囲の数式とより高度な演算を扱うべく発展を続けているのが現状である．その姿は，本特集号の個々のテーマを対象とした稿にみることができる．

数式処理ではアルゴリズム，システム，および応用が三本の柱であるが，本稿では前二者に焦点を当てて過去の歴史をふり返り，将来を展望してみたい．本稿では，数式処理の発展の歴史を便宜上，

草創期 (1953～66)
第一発展期 (1967～75)
第二発展期 (1979～現在)

と分けてみた．それぞれは第2, 3, 4章で詳しく見る．草創期は文字どおり数式処理の始まりで，アルゴリズムもシステムも幼稚なものであったが，数式処理の有用性と可能性が明確に認識された時期である．第一発展期には，効率的な因数分解法や決定的手順に基づく不定積分法などが発見されたことに加えて，REDUCE や MACSYMA などの本格的システムが出現した．第二発展期には数式処理システムの小型化が成功し，数式処理は巷間におおいに普及していった．アルゴリズム開発が高等数学に重点を移したのもこの時期である．

現在，日本では数式処理が大きな注目を集め，システムに興味を持った研究者が増えている．しかし，数年前までは日本で数式処理の研究者と呼べるものはほんの一にぎりであり，それらの人達により細々と研究が続けられてきた．第5章では，世界に占める日本の研究者の成果からみれば紙面配分のバランスを欠くことを承知の上で，これらの人達の仕事と現在進行中の研究にふれてみたい．

最後に，第6章では（筆者等の独断と偏見に基づいた）将来展望を試みる．

なお，数式処理に関しては，1966年より5年に一度 ACM 主催の国際会議が開催されている．1974年より5年に一度 ACM と欧州が共催で国際会議を開いている．1980年以降は，前記の2大会議が開かれない年に SAME（欧州記号代数計算グループ）が国際会議を催している．さらに，MACSYMA グループや大研究所が不定期的に国際会議を開催しているので，これらの会議報告集が大きな情報源になる．数式処理を主対象の一つとする雑誌として Journal of Symbolic Computation も1985年から発刊されている．

2. 数式処理の草創期 (1953～1966)

2.1 人工知能から始まった数式処理

計算機が広く使用されるようになると，非数値的な数学の計算にも計算機を使用する動きが見られるよう

† Formula Manipulation-History and Future View by Fumio MOTOYOSHI (Electrotechnical Laboratory) and Tateaki SASAKI (Institute of Physical and Chemical Research).

†† 電子技術総合研究所
††† 理化学研究所

になった．このような例の中でおそらく最初のものは数式を記号のままで微分するもので，1953 年に Kahrimanian と Nolan が作成したものである[7),8)]（2人の仕事は別々に行われた）．どちらも数式の入出力は3アドレスの機械語のような形で行われる．たとえば Q=SIN P は S00 00P 000 00Q と，$P=X^3$ は E00 00X 003 00P と表される．Kahrimanian のプログラムではこの形を式の内側から順に記述したものを入力とし，Nolan ではこの逆である．どちらの場合でも実際の式との変換は人間が行う必要がある．

彼らの最初の目的は微分可能性を示すだけであったから，入出力は計算機にとって都合のよい上のような形でもよかった．しかし，目的がひろがるにつれ，しだいに数学で使用される通常の表現に近い形が望まれるようになった．Fortran における式の表現法に制限を加えたり，演算記号を変更したりした表現法が考案された．

上に述べたような処理は，知的処理とはいうもののほとんど知的なところがなく，処理手順が定まっており機械的に行うことができる．これに対して積分は，程度の差はあれ人間の知的な活動を必要とするもので，たとえば答が存在しない場合もある（答が存在しないというのは答があらわな形で表現できないという意味である）．不定積分の問題に対して，1961 年に MIT の Slagle は SAINT というプログラムを作成したが[11)]，それは本格的な数式処理として歴史的に重要な仕事である．このプログラムは大学の教養過程の教科書などから選んだ 86 の問題に対して，2つを除いて解くことができた．残りの2つの問題に対しても解けるようにプログラムを拡張することは容易にできるはずである．

この時期に MIT ではほかにも不定積分のプログラムが作成されているが，SAINT ほど強力ではなく，この時代の積分は SAINT で尽くされているといってもよいほどである．なお，SAINT は当時の数式処理としては珍しく LISP で書かれている．

2.2 応用から始まった数式処理

応用分野の計算の中には，退屈で時間はかかるが単純で機械的であり，計算機で実行するのがふさわしいものが多く存在する．特に天文学の分野においては，衛星などの軌道を摂動法を使用して計算するが，この計算自体は単純なものである．しかし，精度を上げると式の大きさが爆発的に増え，一つの式が何十頁にもなることがあり，人手で計算していたのでは時間はかかるし誤りも生じやすく，なんとか早く正確に計算することが求められていた．

このような単純ではあるが計算量が莫大なものは，ほとんどが多項式の計算に帰着させることが可能であり，計算機で実行させるのも比較的容易に実現できる．具体的には1変数の多項式はその係数だけを配列にとって内部表現とすればよい．多変数の場合には1変数の場合を変形して何種類かの方法が考えられるが，ほぼ同じ手法が使用できる．実際にこのような方法で計算機による数式処理が現実の問題に対して適用されていった．

当時の計算では，係数には固定小数点数あるいは浮動小数点数を使用するものが多く，有限の精度しかないために，演算プログラムを使用するユーザが精度のことを考慮する必要があった．入出力についても，データが多項式ということもあって係数を並べただけのものがほとんどである．プログラムもほとんどは FORTRAN や機械語を使用して書かれており，多項式に使用される記憶領域の管理も自分で行うようになっていた．

2.3 初期のシステム

前節までに述べたプログラムは個人用に作成されたものであり，一般のユーザの使用を前提には作られていない．しかし，ユーザの輪が広がるとそれにつれて広い機能が要求され．プログラムは汎用化へと向かった．すなわち数式処理のシステム化である．数式の処理には共通した部分が多くあり，専用に作成されたものであっても，少し手を加えることによって汎用のシステムにすることができる．あるいは，最初から汎用システムとして作成されたものもある．この時期の汎用システムは，多項式・有理式用のシステムと一般数式用のシステムの2つに分けることができる．

多項式・有理式用のシステムは，システムとしてよりも Fortran などのサブルーチンとして用意しておいて，ユーザが自分のプログラムとつなげて使用する形態が多い．この場合にはほとんどが係数を浮動小数点数で表している．代表的なものに ALPAK と呼ばれるパッケージがある[3)]．これに対して，PM と名づけられたシステムは多項式専用ではあるが，システムとして作成されたものである[5)]．PM は係数として任意桁の整数をとる多変数多項式を扱うことが可能で，多項式の値を求める場合には区間算術を使用することもできる．

この時期の一般数式を扱うためのシステムとして

は, FORMAC と Formula ALGOL が有名である. FORMAC は IBM の Samet 女史が 1965 年に開発したシステムで[9], Fortran のサブルーチンとして使用するようになっており, Fortran の数値計算の能力に加えて数式も扱えるようにしたものである. この FORMAC は, 多項式を扱う ALPAC と組み合わせて, ALTRAN という名前の新しいシステムになっている. Formula ALGOL は, 数式用の新しいデータ型を加えるように Algol を拡張して, 言語の中に数式を取り入れたものである.

多項式・有理式の場合には式を整理する規則が簡単で, 変数に順序をつければ一意的な表現にすることが容易である. これに対して, 一般数式の場合には数式の簡単化は容易ではなく, この時期のシステムでは簡単化は深く考慮されてはいない. もちろん 0 を足すとか 1 を掛けるといった定数削除の簡単化は行われている. 簡単化については次章以降で述べることにする.

3. 数式処理の第1発展期 (1967～75)

3.1 アルゴリズムの進展

数式処理の研究はこの時期に多くのアルゴリズムを生み出している. その中で, 多項式 GCD と因数分解の計算法, 初等関数の不定積分アルゴリズムを簡単に紹介しよう.

多項式や有理式の処理をする場合には, 2つの多項式の最大公約式 (GCD) を求める必要が頻繁に生じる. GCD に関しては, ユークリッドの互除法で多項式剰余列を計算する有名な方法が知られている. しかし, 実際に多項式剰余列を計算してみるとすぐに分かることだが, 計算につれて数式が猛烈な勢いで大きくなる. ところが, 剰余列 $(P_1, \cdots, P_{k-1}, P_k, \cdots)$ を詳しく解析すると, P_k には GCD に無関係な因子が含まれ, それが数式巨大化の最大の原因であることが分かった. その因子は $P_1, \cdots, P_{k-1}$ の主係数の組み合わせで表され, 除算により容易に除くことができる. かくして, ユークリッドの互除法による GCD 計算法は実用的なアルゴリズムに改良されたのである. 1966～71 年頃の Collins や Brown らの仕事である.

多項式の因数分解については, 数学では Kronecker の考察した方法が知られており, それで問題は解決したと考えられていた. しかし, Kronecker の方法は可能な因子をしらみつぶしにチェックする方法であり, 有限時間内に計算は終了するというものの, 少しでも複雑な多項式に対しては時間がかかり過ぎてほとんど実用にはならなかった. これに対して, 1967 年 Berlekamp は有限体 (具体的には素数 p のガロア体) における効率的な因数分解法を考察した[13]. Berlekamp の方法を用いて, 1969 年 Zassenhaus は (1) まず多項式を素数 p を法とする空間で因数分解し, (2) 次にその結果を Hensel の補題により $\to p^2 \to p^3 \to p^4 \cdots$ と法を上げて, (3) 最終的に整数上での因子を構成する方法を考案した[22]. この方法は Wang により直ちに多変数多項式に拡張され, 因数分解法は計算量の点で劇的に改善された. 現在では, かなり大きな多変数多項式でも因数分解できるが, もしもアルゴリズムの改善がなければ, 因数分解は計算機にとっても全く不得手の演算であったろう. なお, Hensel の補題を利用して GCD の計算も効率化された.

積分に関しては, 従来は人間の高度な知的活動を必要とするものと考えられており, 人工知能の研究の対象と考えられていた. したがって, その方法も適当な「勘」に基づいて部分積分や置換積分を試み, うまく行けばそれでよし, うまく行かなければ別の方法を試みるかあきらめる, という発見的方法に頼っていた. 実際, 第2章で述べた Slagle のプログラムは発見的方法を組み込んだものである. ところが, 1967 年 Risch は exp と log を任意に組み合わせて得られる数式の集合 (超越的拡大による初等関数の集合) に対しては, 「積分が初等関数の集合に入るかどうかを判定し, 入る場合にはそれを有限回の手順で構成する」アルゴリズムを発見した[21]. さらに Risch は, 平方根関数や立方根関数などの代数関数によって代数拡大された初等関数の集合も含むようにアルゴリズムを拡張した.

数学では「初等関数に対する一般的積分アルゴリズムは存在しないのではないか」といわれていたが, Risch はその誤った考えを打破した. Risch の用いた方法はその後の不定積分アルゴリズムのひな型になるとともに, 現在では微分方程式などの計算に対しても適用されるようになった.

バラ色に輝いていた数式処理のアルゴリズムにも限界があることが認識されたのもこの時期である. それは数式の簡単化に関係しており, 次のようなものである:「あるクラスの式については, 与えられた式が恒等的に 0 に等しいかどうかを判断する手続きが存在しない」[20]. このクラスの例としては, 定数として有理数と π, 変数として x, 関数として sin 関数と絶対値関数を含み, それらに足し算と掛け算を施してできる関数集合があり, わりと身近な式が含まれる.

ある式が恒等的に0に等しいことが判定できないと，2つの式が等しいかどうかの判定もできないことになり，計算がある段階以上に進まなくなったり，ときには誤った結果に導くことになる．実際のシステムでは扱う式のクラスを限定するか，限定せずに誤った結果になるかもしれない危険をおかして処理を行うことになる．また，数式の簡単化を実際に組み込む際には，アルゴリズムで可能なものはすべて組み込むわけではないことに注意しなければならない．実際上のシステムでは計算時間は重要な要素であるから，時間が多くかかる簡単化は行われず，ユーザが途中結果をみて危険な部分を避けるようになっているものが多い．

3.2 システム開発のラッシュ

この時期には，前節で述べたようなアルゴリズムの開発と並行して，数式処理システムの開発が活発に行われた．今日一般に広く使用されている数式処理システムの多くはこの時期に生まれた．代表的なものをいくつかとりあげる．

MIT で開発された MACSYMA は現在使用されているもののなかでは機能が最も豊富なシステムである[19]．MIT では MACSYMA に先立ち，MATHLAB という名のシステムを開発していたが，MACSYMA は MATHLAB をプロトタイプとして大幅に発展させたシステムで，1967年から1983年まで約100年人の労力を投入したといわれている．MACSYMA は対話的に使用することを主目的にしており，ユーザが必要とするであろうほとんどの機能を持たせるようにしてある．数式処理では速度と機能はなかなか両立しないが，MACSYMA では機能に重点をおいているために速度がある程度犠牲になっている．

MACSYMA とよく対比される有名なシステムに REDUCE がある[15]．REDUCE も MACSYMA と同時期に開発が始まり，同時期にユーザに開放されたが，MACSYMA と異なり理工学への応用を第一義に開発された．もとは量子電磁力学の計算をするために作成されたシステムであるが，これに少し手を加えることによって汎用の数式処理システムとなった．応用志向であるため，機能の充実よりも速度と移植性の良さが追求されている．速度は内部表現のベースを多項式の正準表現（実際には多項式の比である有理式）にすることにより達成しているが，そのため一般数式の扱いが少し不自由になっている．移植性の良さは，数ある Lisp の方言の中でも非常に簡単な Standard Lisp でシステムを記述することによって達成している．そのため REDUCE は最近ではパソコン上でも稼働するようになった．MIT の MACSYMA 開発グループが解散（1983年）したのに対して，REDUCE は現在でも拡充を続けており，とくに最近は機能の充実を目指している．

このほかに，天体力学で利用度の高い三角級数の計算を主目的に作られた CAMAL や，IBM の Watson 研究所で同社の社内利用のために開発された SCRATCHPAD[16] といったシステムがこの時期の数式処理システムとして有名である．

3.3 目覚ましい応用例

数式処理の応用には2種類が考えられる．その1つは，電卓を使用して数値計算を行うように，手計算で行ってもなんとかできるが，コンピュータを利用すれば能率も上がるし計算間違いもない，という種類の計算である．この場合には対話的に使用することがほとんどで，式の大きさもそれほどではなく，計算量もたかが知れている．そのかわり，機能的には因数分解や積分などの高度な演算が求められることが多い．このような利用法は，計算機が個人用の TSS 端末で手軽に使用できる環境で始めて可能になるもので，当時としては MIT における MACSYMA だけが可能であり，ほかにはほとんど例がない．

もう1つの利用法は莫大な項数の式を扱う場合である．理工学では頻繁に巨大な式を扱うが，それは多項式の演算の組み合わせと項の置き換えで可能なものが多くあり，高い機能を持たないシステムでも実行できる．あるいは，複雑で大規模な数式の不定積分などは人間が事前に問題を簡単化し，単純なパターンマッチングと項の置き換えだけで答が出るようにしておく．そうしたのち，手計算ではうんざりするような計算を計算機に実行させるのである．したがって，応用の大規模な計算は，そのほとんどが低機能の高速システムにより実行された．

汎用システムとして各所で頻繁に使われた最初のシステムは FORMAC である．FORMAC は天体力学における衛星軌道の計算や，地震波の伝播の計算など，実に種々の計算に使われた．そのなかでも特に Delaunay の計算のチェックが有名である．Delaunay は19世紀の天体力学者で，月の運動を研究し，約20年の歳月を計算に費やしたあと，計算過程と結果を2冊の本にして出版した．FORMAC を使って彼の計算をチェックしたところ，分数の足し算のミスが1つ発見された．そのミスが伝播して，それ以後の答が7箇

所で系統的に誤っていたが，計算ミスは驚くべきことにそれ1つであった．このチェックには6カ月を要したといわれている．

上記の例は単なる計算チェックであり価値は低いが，全く新しい計算を実行して，計算機による数式処理の威力をまざまざと見せつけたのは量子電磁力学の研究者たちである．彼らは電子の磁気能率の高次補正項を計算した．この量は，その値が実験的に非常な高精度（10^{-6}〜10^{-7}）で知られており，当時の物理学の最先端の理論であった量子電磁場の理論の適用限界を示すかも知れないと考えられていた．4次の摂動項は1950年に手で計算され出版された．しかし，その計算にはミスがあることが発見され，1957年に正しい値が得られた．6次の摂動項（奇数次の項は0である）は，文字通り手に負えない代物であると考えられたから，専用の超高速システムを開発して計算が行われた．計算は1973年に終了したが，2つのグループにより独立なシステムを用いて独立に行われた結果がピタリと一致し，また実験値とも誤差の範囲内でピタリと一致した．現在では8次の摂動項が計算され，$\sim 10^{-9}$ にまで精度が向上した実験値と理論値が一致している．

4. 数式処理の第2発展期（1979〜現在）

4.1 高等数学への進出

因数分解が実用的になり，初等関数の不定積分法が発見されたあと，数式処理の研究はそれらのアルゴリズムの改良，より広いクラスの不定積分や微分方程式の求解アルゴリズム，さらには定積分や級数の和の計算法の研究へと向かった．これらの研究の多くは理工学の分野でよく知られた数式と演算を対象としており，情報科学系の研究者に主導されていた．

しかしながら，数式処理のアルゴリズムは多くの数学者を魅了しており，彼ら数学者（あるいは数学志向の研究者）は従来型の研究にはあきたらず，より数学的な演算を数式処理の対象とし始めた．代表的な例は代数的拡大体上の演算（特に因数分解），代数幾何学の諸量の計算アルゴリズム，などである．これらの研究は，代数的拡大体上の演算を除き，互いに関連なく行われていたが，Buchberger の Groebner 基底の計算アルゴリズム[23]が研究者間に認められるにしたがい，コンピュータ環論という一つの大きな流れになってきた．多項式イデアルの基本演算のいくつかに関しては，すでにアルゴリズムの効率化が達成された．また，代数的拡大体上の因数分解に対しては，計算量が多項式時間に収まるアルゴリズムが開発された．

一方，数学の研究に計算機を役立てることは古くから行われており，実験数学という言葉があるくらいである．特に，整数論と群論では計算機の利用が盛んで，群論用にかなりの数のシステムが作られた．なかでも，オーストラリアで開発された CAYLAY という名のシステムは巨大で，Fortran で20万ステップもの規模である（CAYLAY は1975年頃から開発が進められた）．群の中には数式を要素とするものも当然含まれ，また演算の中で数式を処理する場面も多い．したがって，システムの多くは簡単な数式処理機能を含んでいる．

群論用システムと通常の数式処理システムを融合・結合させたシステムはまだ存在しないが，互いに大いに興味をもっている．将来は融合化の道が模索されることは間違いなかろう．

4.2 第2のシステム開発ラッシュ

この時期になって，それまでの数式処理システムに対する反省が行われて，新しいシステムの作成が試みられるようになった．システムに対する不満は，1)パソコンに乗せるには大きすぎる（REDUCE でさえ），2)巨大すぎて移植に不便（MACSYMA），3)計算が遅い（MACSYMA など），4)計算が下手である（REDUCE など），5)数値計算との混合計算ができない，6)システムの拡充（特に高等数学の組み込み）が困難，などである（上記の不満はすべてのシステムにあてはまるわけではもちろんなく，また不満点のなかには相反するものもある）．このような状況が新しいシステムの開発をうながし，μ-MATH（1979年，ハワイ大学）[25]，SMP（1981年，CALTEC）[24]，MAPLE（1983年，カナダ）などが発表され，また国内では GAL の開発が1982年から始まった．

μ-MATH はマイコン上で稼働させることを最大の目標に開発されたが，その割には機能が豊富である．SMP は応用志向を強く追求した巨大システムで，高速性，応用上有用な機能の豊富さ，数値計算との融合性，を兼ね備えている．MACSYMA を下敷にして開発されたため，使用勝手は MACSYMA によく似ている．MAPLE は学生の数学教育用に，小型で移植性に優れたシステムとして開発されたが，バランスがとれたエレガントなシステムである．GAL については次章で触れる．

一方，数式処理で扱う対象が数学化するにつれ，体や環に代表される数学的空間を真面目に扱う必要が生

じてきた．そのためには，単にシステムにこれらの数学的空間を型として導入するだけでは不十分で，たとえばユーザが独自の代数空間を定義する（すなわち新しい型を導入する）ことも可能であることが望ましい．このような目的のために，IBM 研究所では SCRATCHPAD を改良した SCRATCHPAD II を開発中である．SCRATCHPAD II では，ソフトウェア工学で提唱されている抽象データ型の考えを導入して，ユーザによる数学空間の定義に対応しようとしている．数学では，異なる空間で同じ演算（たとえば (GCD) を同じアルゴリズム（たとえばユークリッドの互除法）で計算することが多い．この場合，それぞれの空間で別々に手続きを書くのは労力とメモリの無駄である．SCRATCHPAD II ではデータ型をパラメータ化して，異なる手続きの統合を計っている．

上記のような動きはシステムの現代化と呼ばれているが，それに伴って，REDUCE や MACSYMA などのそれまでの数式処理システムは古典的といわれるようになった．

4.3 数式処理の普及

現在，数式処理は各界・各階層へ急速に普及が進んでいる．すでにシステムを長年使ってきた物理学者や工学者は，数式処理システムが研究の不可欠の道具であることをよく認識している．たとえば一般相対論では，（計量テンソルが与えられたときの）アインシュタイン方程式の解の計算，解の分類，一般座標変換による解の一致性のチェック，がルーチン化されている（これらの計算は膨大な手間を要するが，すでに方程式の厳密解が数十個発見されている）．

このように普及した第1の理由は計算機の進歩である．計算機による数式処理は，扱うデータ量も多いし，処理の内容も複雑になるために，どうしても大規模になってしまう．古典的数式処理の時代には，汎用のシステムを走らすには大型計算機を必要とし，その場合でも計算機をほとんど一人占めする形でしか使用できなかった．そのため，ごく一部の限られた人しか数式処理の恩恵にあずかることができなかった．しかし，ハードウェアの進歩は目覚ましく，現在では大型計算機上では計算時間，メモリのいずれもほとんど不満はなくなっている．さらに Lisp マシンも開発された．Lisp マシンは，もともとは数式処理を実行させることを目的として設計が行われたもので，数式処理をもっと手軽に利用できるようにするためのパーソナルコンピュータである．そのため，いろいろなツールがそろっており，快適な環境で数式処理を行うことができる．

数式処理の普及に関しては，以上に述べた計算機科学全体の進歩に加えて，Hearn と Stoutemyer の名を挙げるべきであろう．Hearn は REDUCE を世界各所に移植し，また世界各地を講演して回り，数式処理の啓蒙に努めた．現在世界中に REDUCE がこれだけ普及したのは Hearn の努力の結果である．Stoutemyer は μ-MATH を開発し，低価格で販売を始めた．μ-MATH はマイコン上で動く世界初のシステムであったので，おりからのマイコンのパワー・アップとマッチして急速な勢いで各界に広がっていった．特に，CAI（コンピュータを利用した教育）に与えた影響は非常に大きい．μ-MATH は改変・拡充が容易なので，ユーザがシステムに手を入れて自分独自のシステムにすることも可能である．ただしマイコンという制約があり，速度・容量の面で大きな障害となっていることは否めない．

なお，システムが普及するにつれ不満点は徐々に改善されてきたが，現在のシステムが依然大きな欠点をかかえていることも事実である．ユーザの不満を減らすべくシステムを改善することは，現在でも重要な仕事である．

5. 日本における数式処理研究

5.1 初期の研究

日本においても数式処理の研究は 1960 年代から行われていた．おそらく日本で最初の数式処理システムは魚木によるものであろう．このほかにも雨宮・森・吉村や石黒らが Lisp で数式処理を行っていた．また渡辺は常微分方程式のためのプログラムを作成した．そのほかにもいくつかのプログラムが作られたが，いずれもシステムと呼ぶほどのものではない．日本で最初の本格的なシステムは，電電公社（現 NTT）横須賀通研が 1975 年に開発した AL である[26]．AL は一般ユーザへの計算受託サービスという形で商用化されたが，受託件数が少なくもっぱら社内用計算に使われた．開発者が配置転換になるとともに保守が続かず，現在は死に体である．

日本で数式処理が本格的にユーザに使われるようになったのは，後藤・金田・寺島らが HLISP という名の Lisp 処理系を開発し，それに REDUCE を乗せて各所に移植したあとである（〜1975）．後藤らはその後も REDUCE の普及に努め，現在では主な大学

の計算センターで REDUCE が使えるようになった．後藤らはさらに Lisp マシン FLATS を開発し，REDUCE の高速化に努めている．

応用面では一応の成果をみたものの，1979 年以前の我が国でのアルゴリズム研究は非常に少なく，後藤・金田のハッシングを利用した多項式の高速乗算法，ほかを数えるのみである．

5.2 現在の研究

1980 年代に入り，京大・数理研で数式処理と数学研究への応用をテーマにした研究会が毎年開催され，また理化学研究所では代数的計算法と題するシンポジウムがほぼ毎年開かれるようになった．これらの集会は数式処理研究に対する関心を急速に高め，各所にシステム開発の気運をもたらすとともに，アルゴリズム研究も若干の活況を呈するようになった．1984 年にはソフトウェア科学会に数式処理研究会が設けられ，年に数回の研究会が開かれるまでになった．

システム関係では，1980 年に佐々木・金田・村尾・渡辺が大規模国産システムを構想し仕事を進めていたが解散，この動きは佐々木により 1982 年から GAL 計画へと引きつがれた．GAL は次章に述べるような数式処理の将来を展望して設計された大型高速システムである．小鹿・三井・渡部は代数方程式の求解のための数値・数式混合システム NAES を REDUCE を利用して開発した．渡辺は彼の以前の微分方程式求解プログラムを改良して，大幅に機能強化した．増田は学生教育用に BASIC でミニシステムを作った．対馬らは μ-MATH を拡変・拡充して，微分記号をオペレータとして扱える数理科学用のシステムを作った．野田らは，C で書いた Prolog 上に，数値と数式の混合演算ができるマイコンシステムを開発中である．安井は APL 2 上に数式処理システムを作成した．そのほかに，ICOT や筆者の一人 (F. M.)，慶大などでもシステム開発計画を発表している．これらの研究で注目されるのは，数式処理に人工知能の手法を導入しようという動きである．対馬や ICOT，筆者の一人 (F. M.) の研究がそれである．

1980 年代の我が国の主なアルゴリズム研究を列記する．

渡辺（隼）：2 階線形常微分方程式の解法；

佐々木・村尾：記号行列式に対する効率的ガウス消去法；

佐々木・古川：多重多項式剰余列の理論；

佐々木：K [x] 上での線形不定方程式の多項式解の公式；

小林・藤瀬・古川：斉次方程式の C 上での一次因子への因数分解アルゴリズム；

小林・古川・佐々木：Groebner 基底の無限級数，および $K[x_1, \cdots, x_n]$ 上の加群への拡張．

6. 数式処理の将来の展望

6.1 数式と数値およびグラフィックスの融合

数値と数式の融合とは，単に数式の変数に値を代入して数値化したり，あるいは数式を FORTRAN 形式で出力したりするだけではない．現在，数値計算は実にさまざまな問題に適用されているが，それは対象を数値化することにより計算が非常に容易になるからである．数式も現実の実体をモデル化したものが多いが，数値に比べて構造を持つだけ計算は容易ではない．数式処理のアルゴリズムはやっかいで時間がかかり，またアルゴリズム化できない問題も多い．そこで，数値計算の手法と数式処理のそれを混用して，より質の高い計算を効率よく実行しようというのが，数値と数式融合の最大の目的である．

このことは簡単にみえるが，数値計算が FORTRAN の，数式処理が主に Lisp の文化圏に属するから，意外にやっかいである．このため，SMP は C 言語で書いてこの融合を実現しており，また MACSYMA でも融合計算が可能である．しかし，両者ともその機能はたいして使われていない．本格的に使うようになるためには，数値計算の研究者達もまきこんで，さらに研究を深める必要がある．

数式処理システムを対話的に使用するユーザにとって，インタフェースの良し悪しは重要である．現在は式を 2 次元的に出力するシステムが多いが，これもほとんどが文字型端末に表示しているものである．そのためベキ指数や添字の上つき，下つきが 1 行分ずれてしまい見苦しいし，積分記号や分数などの表示も不自然になってしまう．現在ではビットマップ・ディスプレイが一般に使用されるようになっており，今後は活字で組むような自然な形で数式を出力することが求められるようになるであろう．

入力についても，現在の 1 次元的に入力する方式ではユーザとのインタフェースがよいとはいえない．出力の場合と同様に 2 次元的な方式が望ましいと思われるが，具体的にどのようにすればよいかはこれからの課題である．一つの解決策として，筆者らは数式を含む文章までも編集できるエディタをフロントエンドに

持つシステムを考えている．もちろん，このエディタは数式を2次元的に表示するスクリーンエディタであり，数式処理部分への入力としてだけでなく，入力した式をファイルにしまうこともできるものである．文章の中へ数式出力を直接埋め込むこともこのエディタを経由させれば可能となる．

6.2 数学公式データベースの開発と利用

現在の数式処理システムでは超越関数の扱いがまだ十分ではない．超越関数をも扱えるシステムにするにはその簡単化が可能でなければならないが，指数関数や三角関数などの初等超越関数とは異なって，簡単化の規則が複雑である．そこでこれらの規則を覚えておき，必要に応じて検索することが考えられる．また演算によって導かれるはずの式でも，計算能力や計算時間のために公式として持っていたほうがよいものが多い．

そこで数学公式データベースというものが考えられる．このデータベースには2通りの利用法がある．1つは通常のデータベースのようにユーザが必要に応じて公式を検索するものである．この場合，利用者が通常見知っていないような公式や，あるいは最新の計算法などもデータとして収納されることになろうが，数式を検索するのにどのようなキーを使用するかなど，未解決な問題がある．もう1つは，数式処理システム自身が式の変形の際に公式を検索しながら簡単化を行ういうなれば公式の自動運用である．これは全く未知の領域で人工知能的な要素が加わったものになるであろう．GAL 計画ではこれら両方の利用法に対処すべく，研究が行われている．

6.3 高等数学の組み込み

第4章に述べたように，数式処理の高等数学への進出はアルゴリズム研究の面ですでに始まっている．しかし，システム面ではやっと代数拡大体上の因数分解，Groebner 基底の計算法，ほかがインプリメントされたにすぎない．イデアルや代数幾何学の演算で現在研究されているものはそのほとんどが多項式計算に帰着されるが，将来は抽象的な対象を扱うようになると予想される．そのときには，システムは新たな検討をせまられよう．今から考えていても遅くはない．

「そんなに数学的なことを研究して，数学以外の応用の役に立つのか？」との疑問が生じようが，われわれは YES と答えたい．数学といえども多くは具体的な問題があって理論が作られ，それが抽象化・一般化されている．したがって高度に見える抽象理論も具体的な問題と密接に結びついている．たとえば，Groebner 基底は古くからの問題である連立代数方程式に直ちに使えるのである．別の例は不定積分のアルゴリズムである．不定積分のアルゴリズム化は50年前ならば高度に数学的であったろう．しかし，Risch の方法は単に初等関数にとどまらず，微分方程式などに対しても拡張されて応用の役にたっているのである．

6.4 システムの総合化と現代化

おそらく今後の数式処理システムは，従来型のシステムが扱った演算に加えて，高等数学や数学公式，さらに集合式や論理式までも扱うようになると考えられる．また，人工知能の手法も取り入れられるであろう．すなわち，システムの総合化である．

だが，そこに至るには大きな難問を解決しなければならない．MACSYMA はもちろんのこと，最新の REDUCE 3 でさえ一人の人間が開発するには手に負えない．しかも，開発には何年も，ときには10年以上もの期日を要する．そのうえにシステムが総合化すればどうなるのであろうか．

大規模システムの構築に対してはソフトウェア工学の成果が使えよう．高等数学の扱いに対しては SCRATCHPAD II などの成果が使えよう．しかし，いずれも保証されたものではなく，SCRATCHPAD II と同じ技術を使ってみたがうまく行かなかったという例もある．数式処理システムの総合化・現代化は今後に課せられた重大な問題である．もしもそれに成功すれば，それは世界的文化資産となるであろう．意欲ある若人が挑戦するに十分な価値のあるテーマであると思う．

参考文献

1) Buchberger, B., Collins, G. E. and Loos, R. ed.: Computer Algebra-Symbolic and Algebraic Computation, Springer Verlag, New York (1982).
2) Buchberger, B. ed.: Journal of Symbolic Computation, Academic Press (1985).
3) Brown, W. S.: The ALPAK System for Non-Numerical Algebra on a Digital Computer-I. Polynomials in Several Variables and Trancated Power Series with Polynomial Coefficient, Bell Sys. Tech. J. 42 (1963).
4) Brown, W. S. and Leagus, D. C.: A Language and System for Symbolic Algebra on a Digital Computer, Comm. ACM 9 (1966).
5) Collins, G.: PM, A System for Polynomial Manipulation, Comm. ACM 9 (1966).

6) Engelman, C.: MATHLAB-a Program for On-Line Machine Assistance in Symbolic Computations, Proc. AFIPS 1965 Fall Joint Comput. Conf., Vol. 27 (1965).
7) Kahrimanian, H. G.: Analytical Differentiation by a Digital Computer, M. A. Thesis. Temple Univ., Philadelphia, Pa. (May 1953).
8) Nolan, J.: Analytical Differentiation on a Digital Computer, M. A. Thesis, Math. Dept, MIT, Cambridge, Mass. (May 1953).
9) Sammet, J. E. and Bond, E.: Introduction to FORMAC, IEEE Trans. Electron. Computers EC-13, 4 (1965).
10) Sammet, J. E.: Survey of Formula Manipulation, Comm. ACM 9 (1966).
11) Slagle, J. R.: A Heuristic Program that Solves Symbolic Integration Problems in Freshman Calculus, J. ACM 10, 4 (1963).
12) Barton, D. and Fitch, J. P.: General Relativity and the Application of Algebraic Manipulation Systems, Comm. ACM 14 (1971).
13) Berlekamp, E. R.: Factoring Polynomials over Finite Fields, Bell System Tech. J. 46 (1967).
14) Brown, W. S.: Rational Exponential Expressions and a Conjecture Concerning π and e, Amer. Math. Monthly 76 (1969).
15) Hearn, C.: REDUCE 2: A System and Language for Alegebraic Manipulation, Proc. SYMSAM (1971).
16) Griesmer, J. H. and Jenks, R. D.: SCRATCHPAD/1, an Interactive Facility for Symbolic Mathematics, Proc. SYMSAM (1971).
17) Jefferys, W. H.: Automated Algebraic Manipulation in Celestial Mechanics, Comm. ACM 14 (1971).
18) Moses, J.: Algebraic Simplification: A Guide for the Perplexed, Comm. ACM 14 (1971).
19) Martin, W. A. and Fateman, R. J.: The MACSYMA System, Proc. SYMSAM (1971).
20) Richardson, D.: Some Unsolvable Problems Involving Elementary Functions of a Real Variable, J. Symb. Logic 33 (1968).
21) Risch, R. H.: The Problem of Integration in Finite Terms, Trans. AMS. 139 (1969).
22) Zassenhaus, H.: On Hensel Favtorization I, J. Number Theory 1 (1969).
23) Buchberger, B.: Groebner Bases: An Algorithmic Method in Polynomial Ideal Theory, Recent Trends in Multidimensional Systems Theory, D. Reidel Publ. Comp. (1985).
24) Cole, C. A. and Wolfram, S.: SMP-A Symbolic Manipulation Program, Proc. SYMSAC 1981 (1981).
25) Rich, A. and Stoutemyer, D. R.: Capabilities of muMATH-79 Computer Algebra System for the Intel-8080 Microprocessor, Proc. EUROSAM 1979 (1979).
26) 池原 悟, 岡田 博, 他：数式処理言語 AL プログラム説明書, 日本電信電話公社横須賀電気通信研究所 (1975).

(昭和 60 年 12 月 27 日受付)