EUROSAM '79の報告

佐々木 建昭
理化学研究所 情報科学研究室

1. はじめに

本年（1979年）6月25～27日の3日間，南仏Marseilleにおいて，EUROSAM '79（EUROpean symposium on Symbolic and Algebraic Manipulation）と名づけられた，計算機による記号代数処理の国際会議が開催された。この会議は，5年毎に開かれているACM-SIGSAM主催の記号代数処理に関する会議（前回は，1976年にIBM Yorktown Hightsで開かれたSYMSAC '76）の合間に，欧州で開催されており，1974年にStockholmで開催された会議についで二度目である。EUROと銘打ってはいるものの，講演者及び出席者は全世界にまたがっており，SIGSAM主催の会議と並ぶ国際会議である。

記号代数処理の分野では，成果の多くがこの種の会議とSIGSAMの会誌に発表されるだけという（悪しき）因習があるので，現在の研究状況と将来の動向をうかがうためには，これらの会議の様子を把握することが是非必要である。（逆に言えば，会議のproceedingsとSIGSAM Bulletinを読めば，大雑把なことが分る？）

以下では，各講演につき非常に簡単に内容を紹介し，現在の研究状況を把握して頂く。ついで，重要な研究分野の講演を詳述し研究の発展の跡を追うと共に，注目を集めた講演や筆者が重要と考える講演をいくつか抽出して，今後の研究動向を占う。最後に，講演を国別に分類して分析し，各位のnationalismの発揚を促す。

なお，会議の出席者はMosesやHearnなどの常連に加えて，ZassenhausやBobrowなどの非常連著名人を含めて，総勢約120名，講演数は12セッションにわたって48件であった。

2. 各論文のひとくちメモ

重要な論文は3章で詳述し，数学的で難解な論文は題目だけを記す。

2.1. 簡単化

B. Buchberger： $x^2 - y = 0$のごときside relationsの組によって多項式を簡単化する際，余分な簡単化の手順を除く非常に有効な方法を見い出した。

J. Siekmann： x, yを変数，a, b, cを定数，f, gを交換律の成立する式（$f(x,y) = f(y,x)$など）とするとき，これらの関数である二つの項，たとえば$t_1 = f(x, f(b,c))$，$t_2 = f(f(c,y), f(a,b))$が等しくなるためのx, y

に対する条件を求めるアルゴリズムを見い出した。

M. Genesereth：簡単化規則の組により数学的表式を簡単化する際，規則の組が完備である（正準表式を保障する）かどうかの必要十分条件を論じた。

2.2. 物理への応用

J. Fitch：天体力学，量子力学，プラズマ物理，一般相対論，流体力学，光学，数値計算，その他の分野への応用を列記した。

A. Karlhede, J. Åman：一般相対論における二つの計量テンソルの（物理的）等価性を，数式処理システムで解いた例をのべた。

A. Voros：量子力学で，定常 Schrödinger 方程式を適当な近似の下で解くのに REDUCE を利用した。

M. Caffo, E. Remiddi, S. Turrini：量子電磁気学における大規模な積分を解析的に実行した例．数式処理システムの最も目ざましい成果の一つである。

2.3. 行列と方程式

D. Yun, F. Gustavson：$x_0, x_1, \cdots, x_N$ の点で $f(x)$ に一致し，$R_{mn}(x)=U(x)/V(x)$, $\deg(U) \leq m$, $\deg(V) \leq n$, $m+n=N$, なる有理式 R を決定する問題に Extended Euclidean アルゴリズムが使える。さらに，EE アルゴリズムを高速に処理する方法を提案し，種々の応用例をのべた。

R. Moenck, J. Carter：Newton の反復法で，有理関数になるはずの解の近似を中展開で計算し，しかるのち Padé 近似で求める有理式を計算する方法をのべた。実際にテストしたところ，効率は悪かった。

J. Smit：行列式の計算について，あとで詳述。

D. Lazard：$P_i(x_1, \cdots, x_n)$ を多変数多項式とするとき，連立方程式の組 $P_i(x_1 \cdots, x_n)=0$, $i=1, \cdots, k$, の共通根を，変数を同時に消去しつつ，（従来の方法に比べて）高速に計算するアルゴリズムを与えた。

2.4. 代数体

H. Zassenhaus：On the van der Waerden criterion for the group of an equation.

H. Bartz, K. Fischer, H. Folz, H. Zimmer：Some computations relating to torsion points on elliptic curves over number fields.

J. Davenport：代数幾何の計算機処理。あとで詳述。

J. Cohen, D. Yun：整数環 $\mathbb{Z}$ を non-monic 多項式（最高中の係数が 1 でない）によって代数的に拡大する話。たとえば有理整数からガウス整数（$k+il$）への拡大が $\mathbb{Z}[x]/x^2+1$ であるように，$\frac{1}{2}$ の中展開は $\mathbb{Z}[x]/2x-1$ の環で表現できる。

M. Pohst, H. Zassenhaus：On unit computation in real quadratic fields.

2.5. 微分方程式

E. Tournier：定数係数の連立線形微分方程式について，行列の単因子を利用するアルゴリズムを REDUCE に組込んだ。

P. Schmidt：一階の非線形常微分方程式 $f(x,y)y'+g(x,y)=0$ について，

heuristics に基づく置換を何組か組込んだところ，参考書の問題が95％以上解けたという話。

D. Ulery, H. Khalil ： 偏微分方程式を数値的に解くための差分方程式と安定化の条件式を，種々の境界条件に対してMACSYMAを利用して求めた。

K. Geddes ： 非線形常微分方程式を解く方法の一つとして，解析的に求まる解を出発点として，Newton法で反復的に近似の程度をあげる方法を考案した。

2.6. 多項式のアルゴリズム

J. Schwartz： 多項式の同等性を確率的に示す方法，あとで詳述。

R. Zippel ： 疎多項式に対する確率的方法。あとで詳述。

A. Hearn ： GCD計算の(modified) Euclid アルゴリズムにおいても，計算途中で何箇所か割算の検算ルーチンを入れると，(GCD＝1)のときの余計な計算が省略されて，高速化される。

2.7. システム

A. Rich, D. Sfoutemyer ： マイクロプロセッサ上のalgebra system. あとで詳述。

J. Leon, V. Pless ： Combinatorial計算用システムCAMAC '79の紹介。CAMACはGROUPと並んで，群論の計算に威力を発揮した。

A. Norman, P. Moore ： 多項式の表現として，各項をsymbol tableにhashingで割りつけ，乗算を高速化することをねらった数式処理システムの設計。

R. Cowan, M. Griss ： algebra systemにおけるパターンマッチに，ストリングのパターンマッチに用いられているsignature法（hashingの一種）を導入して高速化を計る。

L. Hörnfeldt ： 一般相対論のテンソル計算用に作成されたシステムSHEEPにさらに，一般相対論の計算に必要な種々の機能を組込んだ話。

2.8. アルゴリズム分析

P. Wang： 多変数Hensel constructionにおいて，イデアルの次数をあげる際の各因子項の計算について，新しいrecursiveな方法を提案した。この方法は従来のiterativeな方法よりはるかに効率がよい。

D. Arnon ： A celluar decomposition algorithm for semialgebraic sets.

A. Woll ： 物理に現われるDirac行列の積のTrace計算について。

G. Collins： Hensel constructionで単変数多項式を因数分解する際，mod. p^k ($k \gg 1$)での因子から実際の因子を求めるのに，できるだけ小数個の因子の積からテストするか，できるだけ低い次数となるような因子の積からテストするかの，二つの方針が考えられるが，前者の方が理論的によいことを示した。

2.9. Symbolic－Numeric インタフェイス

E. Ng： Symblic－Numeric インタフェイスについて，多数の例を引用して，reviewした。

J. Van Hulzen ： 最良近似値やresultantなどの実際的計算においては，取扱う式や数値がぼう大になり，algebra systemと固定精度浮動小数演算システム

では不十分で，任意精度浮動小数演算システムなどが必要である。

T. Sasaki： 精度の扱いをできるだけflexibleにして，計算速度と記憶容量をできるだけ効率よくした，任意精度浮動小数演算システムを作った。

G. Caplat： Interval arithmeticによる有理関数の扱いを論じ，それを実現したシステムAMFISについて述べた。

D. Matula, P. Kornerup： メモリーがきつくなると，より簡単な有理数で最良近似するrational arithmeticシステムについて述べた。最後の（正しい）答が簡単な有理数になる場合，途中の有理数がばう大でも，この方法で正しい答が得られることが多いそうである。

2.10. 積分

A. Norman, J. Davenport： 積分についてのreview. あとで詳述。

B. Trager： 代数的関数を含む初等関数の積分。あとで詳述。

J. Davenport： 〃 〃

J. Moses, R. Zippel： 特殊関数の統一的扱いについて。あとで詳述。

2.11. 応用代数

G. Havas, L. Sterling： Integer matrices and Abelian groups.

M. Felsch, J. Neubuser： An algorithm for the computation of conjugacy classes and centralizers in p-groups.

2.12. 言語と設計

R. Jenks： algebra system向きのLISPの設計。あとで詳述。

D. Bobrow, L. Deutsh： INTERLISPの拡張。reliability, modularity, efficiencyを向上させるための，INTERLISPの拡張について。

M. Griss, R. Kesseler, G. Maguire： algebra systemをミニ・マイクロコンにのせるために開発した，LISPのインタプリンターTLISPについて。

J. Campbell, Simon： 実際の問題では，対称性などの性質により式が簡潔に表現できることが多い。数式処理システムの第一ステップで，表式がいくつかの小表式に分割できぬか？，項は冗長な情報を含んでいないか？などを調べて式の圧縮を計る話。

G. Ausiello, G. Mascari： システムプログラマとユーザーのニーズを満たし，さらに加算乗算や簡単化などの操作に適した数式のデータ構造を考察した。

3. Topics

3.1. 行列式の計算

J. Smit： New recursive minor expansion algorithms, ・・・・

記号行列式の計算はおそらく理工学的に最も需要の多いもので，種々のアルゴリズムが発表された。それらの多くは，行あるいは列に関する展開を次々とrecursiveに行う，いわゆる小行列式展開法を用いる。応用問題では，0要素を

多く含む，いわゆるsparseな行列式を扱うことが多いので，sparsityを利用して次のような高速化がなされている。

1)展開に最も便利な行あるいは列を選び，0要素はskipする。

2)0であることが明白な小行列式は0とおく。

3)計算過程で多くの同じ小行列式ができることが多いので，各小行列式ごとにハッシングでuniquenessを調べ，同じ小行列式の重複計算を避ける。

4)上記の過程を少し工夫すれば，common subexpressionを新しい変数でおきかえることにより，計算途上の式と答を非常に簡潔にすることができる。

Smitは上記のアイデアを種々の形でとり入れたアルゴリズムをいくつかの工学の問題に適用し，その有用性を実証している。行列式の計算については，従来の着想に基づくアイデアはほぼ出つくした感がする。新しい着想としては次節を参照。

3.2. 多項式に対する確率的アルゴリズム

J. Schwartz： Probabilistic algorithms for verification of ·····。

R. Zippel： Probalistic algorithms for sparse polynomials。

多項式に関する従来のアルゴリズムには確率の概念は取り入れられていなかったと思う。今回の会議の一つの特徴は，確率的アルゴリズムが導入されたことである。このような斬新でflexibleなアイデアは，従来の方法の限界を乗越える種々のアルゴリズムを可能にする。

例として行列式の計算を考えよう。Vandermondeの行列式

$$\begin{vmatrix} 1 & x_1 & \cdots & x_1^{n-1} \\ & & & \\ 1 & x_n & \cdots & x_n^{n-1} \end{vmatrix} = \prod_{i<j} (x_i - x_j)$$

は，n=100の場合，次数約5000，項数約2^{5000}の多項式を与える。この例からも分るように，大きな記号行列式の展開には，時間的容量的に非常な制約が伴う。別の例としてはintermediate expression swellがある。即ち，最後の答は簡単であるにもかかわらず，計算途中の式の爆発的肥大化のために計算不能になることである。確率的アルゴリズムはこれらの困難を乗り越える可能性をもっている。

まずSchwartzのアイデアを見よう。彼は多項式$Q(z_1, \cdots, z_n) = 0$の判定を考える。Qは次数さえ容易に計算できれば行列式であっても何であってもよい。$z_1, \cdots, z_n$の各変数の個別次数を$d_1, \cdots, d_n$とする。今，各z_iに対して$-k \leq z_i \leq k$（kは正整数）内の整数点だけを考える。このとき，$(2k+1)^n$の格子点上で，Qが0となりうる格子点の個数の上限は容易に与えることができる：$(2k+1)^{n-1}(d_1 + \cdots + d_n)$。したがって，$k \gg (d_1 + \cdots + d_n)$ならば，Qが0に等しくない限り，$(2k+1)^n$の格子点の大部分ではQ≠0となる。よって，格子点のいくつかの点でQを評価してやれば，十分な精度でQ=0の正否を判定することができる。Schwartzはさらに，大きな素数pを法としてQを評価した際の精度も与えている。彼のあげた例では，60個の点でQ=0ならば，10^{-100}の精度でQ=0が主張できる。

次にZippelはmodularアルゴリズムを高速化することを考えた。modularアル

ゴリズムの一般原理を多項式に即していうと以下のようになる：d次の多項式は$(d+1)$個の点での値から構成することができる。したがって，多変数多項式$P(x_1,\cdots,x_n)$を正確に構成しようとすれば，$(d+1)^n$個の点から係数を決める必要がある。しかしながら，nが大きいときには，$(d+1)^n$は巨大である。この困難を救うため，Zippelは次のことを考えた：$f(x)$を考える。$f(x)=0$の根は係数から決まる，0を含むある限られた領域内に収まる。したがって，xに非常に大きな値を代入して$f(x)$が0になれば，$f(x)=0$である可能性が非常に高い。そこで，Pを決定するのに，まずx_1以外の変数に巨大な数を代入して$P_1=P(x_1,\tilde{x}_2,\cdots,\tilde{x}_n)$を決める。ここで，$P_1$のいくつかの項は0になるが，それは0でなかった多項式に数値を代入して偶然に0になったとは考えないで，もともとからその項が0であったと強引に仮定してしまう。次にP_1で残ったそれぞれの項についてx_2依存性を同じようにして決め，$P_2(x_1,x_2,\tilde{x}_3,\cdots,\tilde{x}_n)$を決める。等々。こうして決めた多項式$P_n$は$P$とは一致しないこともあるが，その割合は非常に小さい。もちろん，可能な場合には，P_nとPの一致を何らかの方法でチェックする。彼は実際，この方法で，非常に高速なGCDルーチンを作成した。

3.3. マイクロ algebra system

A. Rich, D. Stoutemyer： Capabiliteies of the MUMATH-78 ・・・。

我々（少なくとも筆者）はこれまで，algebra systemには大型機並の速度と容量がなければ使いものにならない，と考えていた。しかし、Stoutemyerらのシステムはこの先入観を完全にくつがえした（彼ら自身，これほど強力なシステムができるとは予想していなかった）。もちろん，これらのミニシステムは理工学の実際的計算を遂行する能力はないが，彼らのもともとの目的は，高校から大学教養程度の数学教育に使用することであり，その目的には十分にかなう。実際，システムは基本的多項式演算に加えて，微分，積分，求和，行列，方程式，微分方程式などを処理できる。さらに，ユーザーは数式処理用言語でプログラムを組むことができる。この種のシステムは，その手軽さから各所に普及し，多くの人々に数式処理への関心を呼び起こすと考えられるので，今後のなりゆきを期待をもって見守りたい。

3.4. 代数関数の積分

A. Norman, J. Davenport： Integration -- the dust settles ?

B. Trager： Integration of simple radical extensions.

J. Davenport： Integration of algebraic functions.

J. Davenport： The computerization of algebraic geometry.

現在，数式処理の分野で最も活発に研究されているテーマの一つが積分である。上記第一の論文は，有理式の積分からlog，expによる拡大体，さらに代数的拡大体での積分へと発展してきた跡を，素人にも分り易く解説しており興味深い。是非一読を進める。この分野における日本のたち遅れが如実に分る。

積分の分野で現在最もホットなテーマは，代数的拡大を含む初等関数と初等超越関数の積分である。後者は次節に述べるとして，ここでは前者について述べる。

初期の積分アルゴリズムはheuristicsに基づいていたが，Risch以後のものは

次のような決定論的手順から構成される：

1)積分の候補関数（未定関数を含む）を構造定理にもとづき書き下す。

2)候補関数を微分することにより方程式を導き，未定関数を決める。

3)未定関数に対する方程式が矛盾を含む場合は積分は存在しない。

構造定理は初等関数 f に対しては存在する：y が独立変数 x か $\exp(x)$, $\log(x)$, あるいは x の代数関数（x の多項式を係数とする多項式の根）のとき，$f \in F(y)$, ただし F は体，の積分が初等関数ならば

$$V(y) + \Sigma c_i \log W_i(y)$$

でなければならない。ここで K を F の係数を含む定数体，α を K 上での代数的数とするとき，$c_i \in K(\alpha)$, 未定関数 $V(y) \in F(y)$, $W_i(y) \in F(y, \alpha)$ である。

上記の deterministic なアルゴリズムは $y = \exp(x)$ あるいは $\log(x)$ に対しては完成している。代数関数に対しても Risch などがアルゴリズムを提案してはいるが，難点をいくつか含んでである。代数関数の場合，Riemann 面上の多値関数の処理の問題もあるが，何と言っても難しいのは候補関数の選び方である。候補関数の決定には，一般に，y の零点，や極の位置（$y=\sqrt{x^2-x^3}$ の場合，0, 1, ∞）のみならず y を定義する方程式が表わす曲線の幾何学的性質（$y=\sqrt{x^2-x^3}$ では $y^2=x^2-x^3$ が表わす曲線は，原点において直角に交わる，など）も必要である。したがって，代数関数を含む関数の積分には，一般に代数的幾何学を扱うシステムも必要になる。

Davenport はそのようなシステムを作成し，それを利用した積分プログラムを書いている。彼のプログラムはパラメータを含んだ関数も扱うことができる。たとえば，$\int \sqrt{x+\sqrt{a^2+x^2}}/x \, dx$ などを実際に実行してみせている。

これに対し，Trager は根基を一重にしか含まない代数関数（unnested radicals）のみを扱っている。この場合，log と exp に対する Risch のアルゴリズムを少し拡張するだけで積分が実行できることを彼は示した。

しかしながら，log, exp, 代数関数をすべて含む場合は今後の課題である。

3.5. 特殊関数の扱い。

J. Moses, R. Zippel : An extension of Liouville's theorem.

MACSYMA は既に特殊関数をいくつか組込んでいるが，そこに使用されたアルゴリズムはパターンマッチか heuristics に基づく。特殊関数に対しては，積分に対する Risch のアルゴリズムのようなアルゴリズムは存在しなかった。この論文は表面上，積分の範囲に特殊関数も含めようと意図しただけかに見えるが，実は特殊関数の一般的取扱いを模索しているのである。これは，積分の裏返しが微分であり，特殊関数の多くが微分方程式で定義されることから納得できよう。

Moses と Zippel の提案を略述しよう。初等関数の積分においては，有理関数体 F に ϑ：F 上での代数関数か $\exp(x)$か $\log(x)$, $x \in F$, を添加した拡大体を扱うが，彼らはさらに，ϑ として $\vartheta = f(u)$, $f(u) \notin F$, $\vartheta' = f'(u)u' \in F$ なる関数も扱おうと提案するのである。即ち，f は任意ではなく，F の適当な要素の積分とするのである。このとき，彼らの得た結果は次の通り： F の任意の要素 W に対して，その積分が $F(\vartheta_1, \vartheta_2 \cdots \vartheta_R)$ に含まれるならば，それは

$$\int W dx = U_0 + \sum_{i=1}^{n_1} c_i \log U_i + \sum_{j=1}^{n_2} d_i f(v_j), \quad U_0, U_i, V_j \in F$$

と書けなければならない。この構造定理を例で言うと，$\int e^{-x^2}dx = \mathrm{erf}(x)$ であるから，e^{-x^2} に関して有理式の積分が，e^{-x^2} と $\mathrm{erf}(x)$ の有理式で表現されるならば，それは $\mathrm{erf}(x)$ について線形にかけることを主張する。

3.5. LISPの拡張

R. Jenks: MODLISP: An introduction.

本論文は，数式処理システムの最もポピュラーなホスト言語であるLISPが，数式処理専用のdialectに向かって分化している例の一つを示す(別の例はFLATS!)。このような分化が，再び別の標準的LISPに吸収されてLISPの機能拡張につながるか，LISPのpreprocessorのようなシステムにとどまるか，あるいはのたれ死にするか，興味のあるところである。ちなみにMODLISPのMODはModern，Modular，Modest，およびModifiableを意味するそうである。

MODLISPは基本的概念として以下の6つをもつ：

1) Expression: basic-object (symbolとconstant) およびlist.
2) Type: expressionの属するクラス(symbol, realなど)を指定する。
3) Domain: 計算の領域(integer, algebraic, ringなど)を指定する。
4) Mode: domainの属するクラス(polynomialのtypeをもつすべてのdomainなど)を帰納的に指定する。
5) Operator.
6) Environment.

これらのうち，3)と4)が新しい概念である(MODの概念はMOD-REDUCEにとり入れられているが，それがstaticな概念であるのに対して，JenksのMODはdynamicな概念である)。これらの概念がどれほど有用かは，筆者の乏しい経験からは分らないが，少なくとも書きかえ規則(rewrite-rules)による数式簡単化を高速化する際などには威力を発揮するであろう。

4. 雑感

会議での講演を国別に分類してみると，米21，英7，独5，仏5(うち一つはキャンセル)，カナダ，オランダ，スエーデン，イタリア各2，オーストリアとオーストラリア各1，それに日本が1(ただし，論文の国籍は米国)である。この状況は3年前のSYMSACとほぼ同じである。この数字を見てガッカリするのは筆者だけではあるまい。しかも，内容を見るとさらにガッカリする。

一方，講演を記号処理関係と数式処理関係に大雑把に分類すると，約80%以上が数式処理に関するものである。もちろんこれには，純粋にLISPに関する研究はSIGPLANなどの他の会議で発表されるということも考慮しなければならないが，それにもせよ，記号数式処理研究の中心はLISPなどの言語の研究から，数式処理などのアルゴリズム研究へ移向したと言わざるをえない。我が国においても、LISPコンテストなどの結果，すぐれたLISPの処理系がいくつか作成されたので，今後は研究の中心を数式処理の方へ移向させるべきだと考えるが，いかがなものであろうか？

数式処理の分野では，SAINT，SIN，MATHLAB，MACSYMAを作りあげたMITが何と言っても最先端を走っているであろうが，個々のテーマ別に見るとそうでもない。MITの研究が，積分を計算機で実行させることから出発し，大規模なシステムに向ったのに対して，天文学用の応用システムCAMALから出発したCambridge（英）は，Normanがコツコツと因数分解，積分などのアルゴリズムを組込み，彼を中心にDavenportなどが集まって，積分に関する限り，MITと並ぶまでになった。同じことが日本でもできないものだろうか？