解　説

数式処理による積分†

Anthony C. Hearn†† 渡　辺　隼　郎†††

1. はじめに*

本題に入る前に積分の問題を3つ出しておきます．

(a) $\displaystyle\int\frac{dx}{x^6-1}$,

(b) $\displaystyle\int x^7(\log x)^6 dx$,

(c) $\displaystyle\int\frac{1}{x}\left(\frac{1}{\sqrt{x-1}}+\frac{1}{\sqrt{x-4}}\right)dx$.

この解説が退屈になったり，難しくて分からなくなったら取り組んでみて下さい．終りに誰がどれだけ解けたか聞いてみたいものです．本解説の内容は数式処理による積分です．つまり式をコンピュータが読んで，その積分を式の形で求めます．それも不定積分の話が中心で，定積分ではありません．物理学という分野で実際に現れる積分のほとんどは定積分です．しかしもしも上端下端のない不定積分を求めることができれば，定積分は代入と極限操作により求めることができます．ですから明らかに不定積分を求めることの方が定積分を求めることより基本的です．しかし皆さんの中には「定積分はあるけれども対応する不定積分の閉形式による解はないような積分は沢山ある」という事実を思い出す人もいることと思います．閉形式は後で分かります．こういう定積分を解くには複素解析的な方法を用います．つまり積分経路を上手に選んでCauchy の定理を利用したりします．この解説では解析的方法でなく代数的方法を説明します．もとは解析的である問題を代数的な問題に直します．

数式処理による積分計算の手続きを説明する前に，問題の歴史を少し振り返ってみましょう．最初の積分プログラムは1960年代の初めに書かれました．最も有名な1つは MIT の Slagle が書いた SAINT で，人工知能の学位論文として世に出ました．「積分を解くには知能が必要です．したがって積分を解くコンピュータのプログラムを書けるなら，それは明らかに知能を持たねばなりません．機械の上に作るのですから，自然知能ではなく人工知能ですが．」これが彼のアイデアでした．それから長い年月が過ぎ去り，今では積分を人工知能として考える人はなく，みな手続きと考えます．しかもこの手続きの通りに作業するのに知能は不必要で，ただ適当なコンピュータのシステムがあればよいのです．しかし長い間，閉形式による積分を解く手続きは存在しないと考えられていました．実際 G. H. Hardy は1916年に書かれた彼の有名な著書「一変数関数の積分」の中で「初等関数の積分を完全に行う手続きは存在しないだろう」と予想しました．したがって初期の積分プログラムはすべて，型合わせ，発見的プログラム，特殊目的プログラムという3つの技法を採用してきました．

型合わせは積分表を引くことにたとえられる技法です．まず被積分関数である式を積分表の対応する項目と比較します．その際変数の名前を変えたり，x を $x/2$ で置き換えたり，あるいはもっと複雑な変数の変換をしたりします．うまくいけば積分表の中に丁度同じ式が見つかるわけです．Slagleが初めて作り，Mosesがその後1968年にもっと強力なものにした積分プログラムはこういう方式でした．つまり表と比較して一致すればそれを解とし，一致するものがなければあきらめるという部分があり，その上部に解くべき積分を表中の式に最も良く適合させるような変換を見つける部分，すなわち発見的プログラム，があります．ある意味でこれは渡辺の常微分方程式プログラムとも似ています．つまりある問題とか微分方程式を，答えを書き下せる良く知られた標準の問題もしくは微分方程式にうまく移す変換を見つける工夫がその内容の1つだからです．しかしこの2つの技法だけでは全ての問題

† Analytic Integration by Computer by Anthony C. HEARN (Information Science Department, the RAND Corporation) and Shunro WATANABE (Department of Mathematics, Tsuda College).

†† RAND Corporation　情報科学部門

††† 津田塾大学数学教室

* 本解説は1978年11月理化学研究所における A. C. Hearn の講演記録に渡辺が手を入れたものです．内容，文ともに優れたところは Hearn に属し，責任は渡辺にあります．

に対しては十分でありませんでした．そこでこの2つの技法のほかに，特定の範囲の問題に対する特殊な手続きが必要になります．積分でいうと有理式の不定積分を求める手続きが必要になります．要するに1968年頃までは積分に対して基本的な技法が3つあったわけです．

とにかくこの3つの基本的技法を用いた方法は大変成功しました．それで物理学の問題に含まれる積分を解くために沢山のプログラムが作られました．その中では型合わせが主な技法であったことを文献で確かめることができます．(1)カナダの Waterloo 大学の W. Gentleman が数年前にした立派な仕事が今 SIGSAM の会報に載っています．内容は陽子とヘリウム原子が衝突するときの境界状態に関するものです．この特殊な問題は，よく設定された積分の閉形式解を求めるという問題を含んでいます．後者は表を引くことで解決できるのです．(2)あるプラズマ系が種種の幾何学的解釈の下で安定か否かを判定するという問題に対して，Kerner と Grimm が行った計算があります．この問題はある関数の零点を見つけることです．この特殊な計算は引き続いて何回か不定積分を行った後端点を代入します，というのは定積分だからです．そうして非常に複雑な結果を得ます．そしてこれを4もしくは5変数で数値積分する代りに1つの変数上で数値積分をします．この方法で全変数による数値積分を不可能にしていた不安定性を克服することができます．(3) C. Andersen は1974の ACM 会議と1978年の MACSYMA 利用者会議において有限要素法について発表しました．彼はこの分野で必要な積分のすべてが型合わせによりかなり容易に求められることを示しました．つまりこの分野の積分を計算するのに非常に大きな一般目的のプログラムはいりません．(4)今から数年前，量子電磁力学に現れる関数の族に対する積分プログラムを開発していた2つのグループがありました．Maison と Petermann は SCHOONSHIP 上で，Fox と Hearn は REDUCE 上で作業をしました．この関数は次の形をしています．

(1)　$I(k, m, n, r, s)=$

$$\int_0^1 dy\int_0^1 dx\int_0^1 dw\int_0^1 dz\,\frac{z^r w^s x^{2k-3+n-m} y^{k-2+n} D^m}{B^n G^k}.$$

ここに B, D, G は次の関係式

$$(1.2)\quad \begin{cases} B=1-xy(1-wz+w^2z^2), \\ D=1-x\{1-wz(1-w)\}, \\ G=w^2xB+yD^2, \end{cases}$$

をみたします．したがって(1)には4つの積分変数があり，被積分関数はその4つの変数の有理式です．この特別な問題の唯一の困難な点は1回積分を行うと対数関数が現れることから派生します．もちろん有理関数を積分する場合これは普通です．ですからもう1回積分すると種々の対数関数上の種々の積分が出て来ます．これらは重対数関数，Spence 関数，Nielsen 関数その他に分類されます．4回の積分を終るまでに解を記述するのに必要な関数の族は全く大きいのです．この場合については，今世紀のごく初めに今では見つけるのが大変なスカンディナヴィア紀要に発表した Nielsen の仕事が，この解のすべてを分類するのに十分ではあります．また積分(1)の5つのパラメータに1，0，2，2，3を代入すると次のような4つの項から成る式を得ます．

(1.3)　$I(1, 0, 2, 2, 3)=$

$$\frac{1}{6}\pi^2-\frac{1}{3}\pi^2\log 2+\frac{1}{2}\zeta(3)+\frac{1}{3}.$$

ここでは Riemann のゼータ関数 $\zeta(x)$ が現れていますが，ほかは何もむずかしい式はありません．

こういう積分を計算するプログラムを書くという動機の方が興味を引きます．1970年代の初めに量子電磁力学の分野において実験と理論の間に誰も説明できない食い違いがあることが分かりました．当時の物理学評論の手紙の欄を見ると，この食い違いを新素粒子の導入そのほかで説明しようとする一群の論文が出ていたことが分かります．しかしどの説明も皆を納得させることはできませんでした．しかし遂にカリフォルニアの Stanford 大学の加速機 SLAC にいる S. Brodsky が決着をつけました．この理論は約10個の引き続く計算に依存しています．そのうち1つを除いては独立した2つのグループが計算したのですが，残りの1つの計算は1つのグループだけが行ったのです．しかも最初の計算は誤りであることが分かったので修正されたのですが，その再計算を検算した人は1人もいなかったのです．結局この修正計算で用いられた幾つかの積分計算が間違っていたのです．これが彼の調査で分かりました．この積分の値を正しく直したということは，計算の値が変わったというだけでなく今や理論と実験が正確に一致したということです．これで物理学評論の手紙欄に出た20幾編かの論文は，それを書く動機そのものが失われました．私達は論文を発表する場合1年間は遅らした方が良いという教訓を得まし

た．とうすれば雑誌が間違った論文ばかりという事態は避けられると思います．

以上が型合わせの事例です．型合わせは今日でもなお重要な役割を演じているのが事実であると言いたかったのです．REDUCE もしくは同様なシステムに簡単な型合わせの機能があれば，かなり広い範囲の積分，たとえば $\int x^k dx$ を計算できます．このために難しいプログラムを REDUCE で書く必要はありません．しかし型合わせがすべての問題に対して十分でないことは良くお分かりのことと思います．

2. Hermite の手続き

完全な積分技法が要るなら完全な積分手続きが必要です．1968 年カリフォルニア大学 Berkeley 校の Robert Risch が記号論理学の学位論文として発表した立派な仕事が完全な積分手続きの提案でした．したがって G. Hardy の「初等関数の積分を閉形式で求める手続きは存在しない」という予想に対する最初の反証でした．これから私は Risch の手続きを，まず元の学位論文に対する 1968 年版はもとより，1976 年に初めて述べられた新しい手続きについても説明するつもりです．今日でもまだ，新しい方の手続きの詳細は論文として公表されていません．ただ1つの文献は 1977 年フランスで開催されたあまり有名でない会合の報告に載ったものです．

Risch の手続きを説明するより前に，もっと有名な手続きから始めたいと思います．つまり特別な場合の方法が自然に Risch の方法へ移る所を見せたいのです．特別な場合とは有理関数の積分手続きのことで，普通 Hermite の方法と呼ばれています．こういう場合，正確に誰が発見したかというのは難しいと思います．非常に多くの人が同時にこの問題と取り組んでいたのですから．しかし，しばらくは Hermite の手続きと呼ぶことにします．

私達が扱う問題は1変数の有理関数 Q/S の積分です．ここに分子 Q と分母 S の係数は整数です．まず初めに Q を S で割り，商 P と余り R を求めます．この過程は Knuth の本にあるようによく知られています．割り算を行うと有理数を係数とする答えが出て来ます．困難は何もありません．次にすることは商である多項式 P の積分です．これは項別に積分すればよく，型合わせがあればできます．この場合は複雑な手続きは必要ではありません．

今度は R/S という分子の次数が分母より低い有理関数を積分しなければなりません．まずすべきことは，分母 S を重根のない因子の積

(2) $S=S_1^1\cdot S_2^2\cdots S_k^k$

に分解することです．その意味は，第1の多項式 S_1^1 の全ての根の重複度は1であり，第2の多項式 S_2^2 の全ての根の重複度は2であり，等々，ということです．あるいは各因子 S_i は単根のみを持ち，互いに素である，すなわち S_i と S_i' の最大公約因子は1であると言い換えることもできます．この分解を行う鍵は「ある多項式とその1階の導関数の最大公約因子は初めの多項式と同じ根を持つが，その重複度は1つずつ少ない」という事実を意識することにあります．たとえば，$S=x^4-x^2$ を考えましょう．これを重根のない因子の積の形に書くと $S=(x^2-1)x^2$ となります．$x=1, x=-1$，と $x=0$ は単根であってその重複度はそれぞれ 1，1，2と考えます．さて $S'=(x^4-x^2)'=4x^3-2x=2x(2x^2-1)$ から，S と S' の最大公約因子は x であることが分かります．これは S と同じ根を持ちますが，x^2-1 と x の重複度はそれぞれ0と1に減っています．この事実を用いて，S からその特別な表現である(2)を得るための非常に単純な手続きを構成することができます．この分解を行うのに因子分解を完全に行う手続きを必要としなかったことに注意して下さい．このことは大切です．この分解を得るにはただ最大公約因子を求める計算のみが必要でした．この注意は実際的な観点から述べています．完全な因子分解は計算上非常に高価な方法であることが分かっています．それに対して最大公約因子のみによるのはずっと安価な方法です．したがって完全な因子分解を避けて最大公約因子の計算で代用しようとするのが普通です．

今度は互いに素な(共通因子のない) 2つの多項式に対して成り立つ非常にうまい性質を用いて，多項式の比 R/S を2つの項の和として表します．R/S の積分は各項の積分の問題に移ります．この特殊な分割の鍵は S_k^k と $S_1\cdot S_2^2\cdots S_{k-1}^{k-1}$ が互いに素であること，互いに素である2つの多項式に対して

(3) $A\cdot S_k^k+B\cdot S_1^1\cdot S_2^2\cdots S_{k-1}^{k-1}=1,$

をみたす多項式 A と B が存在することです．A と B を求める方法は Euclid の手続きの拡張として知られています．これは完全な因子分解を必要としない，最大公約因子を求める型の手続きです．(3)を S で割り R を乗じると初めの多項式の比は次の

$$\frac{R}{S}=\frac{A\cdot R}{S_1^1\cdot S_2^2\cdots S_{k-1}^{k-1}}+\frac{B\cdot R}{S_k^k}$$

のように2つの項の和となります．最大の重複度を持つ項を分離して，それより低い重複度を持つ項が残っているわけです．そこでこれを反復すると遂には

(4) $$\frac{R}{S}=\sum_{i=1}^{k}\frac{A_i}{S_i{}^i}$$

となり，もとの積分は分母が $S_i{}^i$ の形の有理式の積分に帰着しました．S_i の重複度が1ですから，S_i と S_i' との最大公約因子は1です．というのはこれは S_i より重複度が1少ない同じ根を持つ一方 S_i の重複度は1だからです．したがって再び Euclid の手続きの拡張を用いて $B_i\cdot S_i+C_i\cdot S_i'=1$ となる B_i と C_i を求めることができます．両辺を $S_i{}^i$ で割って積分すると

$$\int\frac{A_i}{S_i{}^i}dx=\int\frac{B_i}{S_i{}^{i-1}}dx+\int\frac{C_i\cdot S_i'}{S_i{}^i}dx$$

となります．さらに右辺の第2項を部分積分して整理すると，分母の重複度を1つ減らす次の公式

$$\int\frac{A_i}{S_i{}^i}dx=-\frac{B_i}{(i-1)S_i{}^{i-1}}+\int\frac{B_i+\frac{1}{i-1}C_i'}{S_i{}^{i-1}}dx$$

を得ます．この操作を反復すると，この積分は最後には有理部分と超越部分の2つの和から成る

$$\int\frac{A_i}{S_i{}^i}dx=\frac{T_i}{S_i{}^{i-1}}+\int\frac{W_i}{S_i}dx$$

の形になります．このことと(4)から初めの積分は

(5) $$\int\frac{R}{S}dx=\sum_{i=1}^{k}\frac{T_i}{S_i{}^{i-1}}+\sum_{i=1}^{k}\int\frac{W_i}{S_i}dx$$

と表されます．これが Hermite の方法です．

S と S' の最大公約因子を V, S/V を U としますと $U=S_1\cdot S_2\cdots S_k$, $V=S_2{}^1\cdot S_3{}^2\cdots S_k{}^{k-1}$ ですから，

(6) $$\int\frac{R}{S}dx=\frac{T}{V}+\int\frac{W}{U}dx$$

と書くことができます．R と S に対して U と V は微分と最大公約因子の計算と除算により求めることができます．多項式 T と W の次数の上限は簡単に分かりますから，T と W を未定係数法で決定することができます．

さて(5)または(6)式の中で積分のまま残してある部分を普通超越部分と呼びます．その分母の根の重複度は1なので，この部分を積分するには分母を整数上で既約因子の積に分解し，必要ならさらに代数的数の上で1次因子の積に分解しなければなりません．この超越部分を前と同様の方法で部分分数に展開し，項別に積分すると対数関数の和となって終るわけです．したがって計算を完成するには完全な因子分解が，それもできたら代数的数の上の因子分解が必要です．それはたとえば分母に x^2-2 のような項が現れた場合，$\sqrt{2}$ を含む1次因子を考えなければならないからです．一般に分母に n 次の既約多項式が現れると，n 次の代数的数である根を n 個扱わなければなりません．

今述べたことが「積分プログラムを書くのになぜ長い期間がかかるのか」という疑問に答えてくれます．実際もし積分プログラムを作る立場になったら，途方もなく多数の技法が必要であることが分かるでしょう．明らかに多項式の処理が必要です．有理関数の処理も必要です．多項式の最大公約因子を計算できなければなりません．積分を完成するには完全な因子分解が必要です．部分分数展開が必要です．代数的数も必要です．そしてさらに Risch の手続きを考える場合には，必要な技法の一覧表はますます長くなります．最後にもし初等関数の十分大きな族を扱うとすると初等関数の簡単化プログラムが，またゼロを認識するために初等関数の標準表現法をうまく扱えるようにすることなどが積分を計算するのに必要です．

3. Risch の手続き

以上で有理式の積分を計算する方法を説明する序論を終えますが，この方法はそのまま Risch の方法に接続します．Risch の手続きを2つの部分に分けて説明します．まず元の手続きを説明します．これには元の論文がありますから詳細を知りたければ参照できるわけです．次に新しい手続きを説明し，新旧2つの手続きの差を強調します．

この手続きを完全に述べると極めて複雑になります．一方 Risch の学位論文にもどりますと，この手続きは基本的には19世紀の Liouville の仕事の拡張であることが分かります．Liouville の手続きと方法は少し前に述べた Hermite の方法の基礎です．この手続きが代数的独立性にいかに依存しているかを示します．

代数的独立性を使おうとすれば，初めは超越関数を扱うことに限定する必要があります．代数(的)関数と超越(的)関数との違いを思い出して下さい．代数的とは代数方程式の解で表されることを意味します．たとえば$\sqrt{x}$は代数的です，$y^2-x=0$ がこれを解とするからです．ですから多項式を0とおいた方程式の解として表される，たとえば平方根関数は代数的であり，この方法で定義できない関数，たとえば対数関数とか指数関数は超越的であるといいます．対数関数や指数関数の方程式を定義する多項式を書き下すことはでき

ません．無限級数で表すことはできますが多項式ではできません．この事実はたとえば x と $\log x$ との間に代数的独立性があることを意味しています．この概念が必要なことは間もなく明らかになると思います．

この手続きを log の場合に説明したいと思います．x と $\log x$ の有理式全部からなる関数の族を考えます．問題は次のようになります．「被積分関数がこの関数の族に属するとき，その積分はどうなりますか？」答えは次のようになります．「積分が閉形式で存在するならば，つまり積分がこの関数の族で表されるなら，それは $q(x)$ と $r_i(x)$ をこの関数の族に属する関数，C_i は x に無関係な定数として

$$(7)\quad \int f(x)dx=q(x)+\sum_i C_i \log(r_i(x)),$$

つまり有理部分と対数部分の和でなければなりません．」これは有理式の場合と同じ結論であることを思い出して下さい．有理関数の積分はもしそれが存在すれば，この場合はたまたま存在するのですが，有理部分と対数部分の和です．したがって Risch の命題の証明は Hermite の方法の命題の証明に用いられた，Liouville の定理の証明の拡張になります．この時点で相似性が明らかになったと思います．しかし，Risch の命題を証明することは容易ではありません．命題の内容はかなり明解ですが，特別な場合の証明でさえ沢山のほかの命題の証明を必要とします．しかし Risch の命題の証明は正しいと信じられています．証明を見た人は誰でも正しいと信じていますし，指数関数と対数関数の場合は確かに疑問の余地はないと思います．

そこで $\log x$ の積分を例にとって，Risch の命題の特別な場合が積分を求める際にどう使われるかを見てみたいと思います．Risch の命題は $\log x$ の積分はもしそれが存在するならば，(7)の形すなわち有理部分 q と log の和でなければならないと言っています．ここで q と g の問題があります．(7)の右辺が考えている関数の族に属するとき g とよびます．したがって g か q かは何と読みたいかに依存しますが，x と $\log x$ の有理式です．しかしこれは任意のもので良いのではなく，Risch の構成法の基礎となる「微分して比較する」方法による制限がつきます．もしも g が $\log x$ の有理式なら微分すると分かるように被積分関数 f も有理式でなければなりません．有理式を微分してなくしてしまうことはできません，別の有理式になるだけです．被積分関数は有理式ではないのですから積分は有理式ではありません．したがって g は多項式に限定できます．次に g は $\log x$ の2次式であることが分かります．というのは g がもし3次式であるとすると微分してみれば分かるように被積分関数は $\log x$ について2次の項がなければなりません．ところが被積分関数は $\log x$ について1次ですから，関数 g は $\log x$ について2次でなければならないわけです．

以上のことから $y=\log x$ について得られる

$$(8)\quad \int y\,dx=\frac{A_2}{2}y^2+A_1y+A_0,$$

の両辺を x で微分すると $y=\log x$ は次の式

$$(9)\quad y=\frac{A_2'}{2}y^2+\left(\frac{A_2}{x}+A_1'\right)y+\left(\frac{A_1}{x}+A_0'\right)$$

を満たさなければなりません．ここで代数的独立性を用いる全体の鍵といえる段階に来ました．x と $\log x$ は代数的に独立ですから，方程式(9)の両辺の $\log x$ の次数が等しい項を等置することができます．こういうことは y が $\sqrt{x}$ だったらもちろんできません．しかし超越関数の場合には等置が許されるのです．それで $\log x$ の最高次から等置していきます．これが最良の方法であることが分かります．(9)の左辺に y^2 の項がないので $A_2'=0$ となり，これから $A_2(x)$ は定数 a_2 であることが分かります．y の1次の項を比較して $A_1'+A_2/x=1$ となり，これから次式が分かります．

$$(10)\quad A_1(x)=\int\left(1-\frac{a_2}{x}\right)dx.$$

さて，皆さんの中には「少し待って下さい！積分の問題を解こうとしていたのにそれを別の積分の問題に置き換えてしまいました．これは循環論法ではありませんか？」と聞く人もいるでしょう．しかし答えは「否！」です．ここでしてきたことは，積分の仕方を知っている有理関数体から始めて拡大を反復した体から，帰納的に積分を還元することでした．いまは被積分関数と積分を $\log x$ の多項式あるいは有理式で表すことから始めました．(10)では $\log x$ と独立なより狭い領域での積分を考えます．これから a_2 は0であり $A_1(x)=x+$定数 a_1 が分かります．最後に(9)で $\log x$ と独立な項を比較して $A_0'+A_1/x=0$ となります．同様にして a_1 は0であり $A_0(x)=-x+$定数 a_0 が分かります．ここでこの特別な問題において q は g でした．すなわち(7)の C_i は全て0でなければなりません．というのはもしそうでないとすると積分は $\log(r_i(x))$ という項を持ち，被積分関数が有理式になってしまうからです．したがって $\log x$ の積分は $x\log x$

$-x+$定数 であるという結論が出たわけです．

これが元の Risch の手続きです．非常に直接的に追跡できることを見ました．なぜ，さらに新しい手続きがいるのでしょうか？効率がもっと良く，もっと直接的な方法があれば，それが新しい手続きです．元の Risch の手続きにおいて計算上最も高価なものの1つは，基礎体から多様に拡大した体における積分を1歩1歩解いていく解法であることが分かっています．この場合は，x と $\log x$ の有理式全体から成る体から，x の有理式全体から成る体に移りました．しかし対数関数，指数関数，そのほかを含む体に属する関数の積分を考えるときには，それを1つ1つ取り上げ順々に被積分関数の領域を縮小させて，最終的には有理式の体にまでしなければなりません．

4. Risch-Norman の手続き

Risch は 1976 年に彼の手続きの新版を提案しました．以来これは Risch-Norman の手続きと言われています．というのは1976年の夏に REDUCE グループの一環であるイギリスの Cambridge 大学の Arthur Norman が IBM の York Town Heights 研究所に Risch を訪ねました．二人は新しい手続きについて議論を始め，共にこの手続きの欠けた部分を補い，Norman がそれをプログラムに移したからです．

ここで注目に値するのは，元の Risch の手続きは1973年頃まで実際にはプログラムされなかったということです．ある部分は1972年に完成しましたが，実際には1973年に完成したのです．これをプログラムしたのは MIT の MACSYMA の親分 Joel Moses です．Moses は大変立派なプログラマであり，長期間積分と取り組んでおり，かつこの問題について沢山のことを知っていたのですが，Risch が述べた元の手続きをプログラムに直すのにほとんど5年かかりました．これには2つの理由があると思います．1つは元の手続きはプログラムに直すのがとても難しいのです．それが分かって来ました．対策を必要とする沢山の特殊な場合があります．また指数関数と対数関数を共に扱うのはどろどろと言いたい感じです．もう1つは積分を行うのに本質的な沢山の技法，たとえば完全な因子分解，良い最大公約因子の計算法等々が1968年においては使えませんでした．こういうものが使えるようになったのは後のことです．ですから Norman が積分プログラムを6週間で書けたのは，彼が Yorktown Heights の IBM で使ったシステム SCRATCHPAD がこのような機能を全て持っていた事実を示しています．彼は余計なものを沢山作る必要なしに，積分を求める良いプログラムを開発することができました．しかしこの事実は新しい手続きがずっと単純であることも示しています．

これから説明する新しい手続きも出発点は全く同じです．問題に現れる指数関数，対数関数，そのほかの代数的に独立なものの集りと，その要素の有理式を被積分関数とする積分から出発します．再び $\log x$ の積分という問題を考えましょう．$y=\log x$ として $\int y\,dx$ を考えます．y に関して知っていなければならないことは $y'=1/x$ だけです．さて主な違いは次の段に生じます．思い出して下さい，元の Risch の手続きでは，積分を x の関数を係数とする $\log x$ の有理式として書き表しました．2回以上拡大をしているなら，主たる拡大であると選んだ第1のものたとえば対数関数から出発し，対数関数で表現します．その係数は拡大の第2のものたとえば指数関数を含み，再びその有理式になります．

しかし新しい手続きの場合，関数を帰納的な形ではなく分配的な形に書き表します．全ての変数が指数を持ちまた一様な係数 u_{ij} を持つ

$$(11)\quad \int y\,dx=\sum_{i,j}u_{ij}x^iy^j$$

という形に書き表します．前の方法と違って今度は全部を一度に解きます．全ての拡大を，単に係数 u_{ij} を計算することで解こうとしています．さてここで(11)の右辺が有理式ではなく多項式であるという仮定をしました．元の手続きの場合も同じ仮定をしました．この積分がどちらの変数についても有理式であるとすれば，微分してみると分かるように被積分関数も有理式でなければならないことになります．被積分関数は有理式でないのですから積分も有理式ではありません．ですから(11)の右辺は x と y の多項式です．

さて前と同じく積分すなわち(11)の右辺を微分して被積分関数に等しいと置くと $y'=1/x$ から

$$(12)\quad y=\sum_{i,j}u_{ij}(ix^{i-1}y^j+jx^{i-1}y^{j-1})$$

を得ます．この方法の威力はある意味で次の段階から派生しますが，これは元の手続きにはなかったものです．(12)の和をいじって整級数の変換を行い

$$(13)\quad y=\sum_{i,j}x^iy^j\{(i+1)u_{i+1,\,j}+(j+1)u_{i+1,\,j+1}\}$$

を得ることができます．もちろんこの種の変換を行ったとき端点で何が起きるか注意しなければなりませ

ん．つまり特別な場合を考慮に入れる必要があります．しかしこの手続きについて端点の考察は明らかに容易です．

さてここで前回行ったのとほとんど同じことをします．全ての変数に関する最大次数の項を分離するために漸化関係式を用います．今の場合(13)より最大次数は x^0y^1 すなわち $i=0, j=1$ です．したがって(13)より $1=u_{11}+2u_{12}$ を得ます．さて前に $(\log x)^2$ の項を積分から除いたのと同様な議論を使います．今の場合それは xy^2 すなわち $i=1, j=2$ に対応します．もしも積分が項 xy^2 を含むとすると，微分してみれば分かるように被積分関数は項 y^2 を含まなければなりません．しかしこれは y の1次の項ではないので $1=u_{11}+2u_{12}$ から対応する u_{12} を消去できます．

前に新しい手続きを完全に述べた論文はまだ現れていないと言いました．その理由は，この場合に $u_{12}=0$ を導いたような次数の上限を定めることが，少し後で話をする tan を扱う場合とか代数的拡大を扱うような幾つかの場合において明白でないことにあります．言い換えるとどれだけの項を捨てられるのか分からないのです．上限がないほとんどの場合にはその理由が分かりましたが全ての場合ではないのです．Risch も Norman も2人とも正しくない論文を公にしたくはないので，こういった全ての点が明らかになるまで，発表を延期しているのです．もちろん新しい手続きのある部分に何か基本的な誤りがあると考えることはできます．しかし誰もそうは思っていませんし，新しい手続きによるプログラムは関数の大きな族に対して正しく動いています．

もう一度実際的な観点に立つと，コンピュータのプログラムが積分をして具合が良いことは，結果をいつも微分できますから，それを元の被積分関数と比較して，同じなら積分が正しいと言えることです．実はかつてある会合でこう言いましたら，誰かが手を挙げて「もし微分プログラムが正しくなかったらどうなりますか？」と質問しました．お分かりのごとくこの議論はプログラムの証明のそれと似ています．いつも「証明プログラムが正しいことを証明しましたか？」と聞く人がいます．答えはもちろんこうなります．「積分プログラムと微分プログラムが同一の関数に対してともに間違っていて，積分した結果を微分すると元通りになるということはごくごく小さな確率であり得ることです．しかしとてもあり得るとは思えません．というのはこの2つのプログラムは完全に独立で，同じ多項式の表現を用いている上に，微分プログラムは非常に簡単なのですから．」

話を $\log x$ の積分にもどします．u_{12} という項を消去できたので $u_{11}=1$ を得ます．これを(11)の右辺に代入して xy の項を消去すると $\int y\,dx=xy-\int dx$ となります．これを直接積分して $x\log x-x+$定数という解が得られました．

これが新しい手続きです．この簡単な例では計算の主な違い方，計算の複雑さの差があまりはっきり出ませんでした．しかし新しい方法が前の方法よりずっと直接的であることは分かると思います．実際全ての場合に，新しい方法は見かけよりずっと良い効率を示します．さて2節のはじめに Risch が発表したのは積分の完全な手続きであるといいました．それは何を意味するでしょうか？積分がある関数の族に属するときそれを見つけることができ，属さないときそう言うことができることを意味します．言い換えると，Risch の手続きは決定方法でもあります．つまり積分が存在するかしないかを決定し，存在すれば値を求めることができます．Risch の手続きのような型のものは，それ以前の方法とは全く違うことに注目して下さい．昔の積分プログラムは積分を見つけられなかったとき，「この積分が存在するかしないか分かりません，私の積分表はこれを解く程完全ではありません」というだけでした．ところが積分手続きが出現してからは，新しい積分プログラムは「分かりません」とは言わず，「これが答えです」と言うか「積分はこの関数族の中にはありません」と言うかのどちらかです．

次の例は新しい手続きが今述べたことを非常に簡単に行うのを見せてくれます．有理関数体に $y=e^{x^2}$ を添加した体，すなわち x と y の有理式全体において，$x^2\cdot y$ の積分を考えます．積分が今考えている関数族の中にあるなら，それは x と y の整級数で

$$\int x^2y\,dx=\sum_{i,j}u_{ij}x^iy^j$$

と書けなければなりません．次の微分する段階で，$y'=2xy$ を用いて整級数の変換をすると

$$x^2y=\sum_{i,j}x^iy^j\{(i+1)u_{i+1,j}+2ju_{i-1,j}\} \tag{14}$$

を得ます．最大次数の項 x^2y を比較して $i=2, j=1$ より $1=3u_{31}+2u_{11}$ を得ます．u_{31} という項は，x^3y という項に対応します．この項が積分にあるとすると微分してみると分かるように x^4y という項が被積分関数の中にあるはずです．ところが被積分関数の中に

は高々 x^2y しかありませんから u_{31} は0で $u_{11}=1/2$ という結果になりました．この項を積分から除くと

(15) $\int x^2y\,dx=\frac{1}{2}xy-\frac{1}{2}\int y\,dx$

となります．右辺の第2項を $\sum u_{ij}x^iy^j$ とおき微分すると，

(16) $-\frac{1}{2}y=\sum_{i,j}x^iy^j\{(i+1)u_{i+1,j}+2ju_{i-1,j}\}$

を得ます．(16)で x^0y の項を比較して $i=0, j=1$ より $-1/2=u_{1,1}+2u_{-1,1}$ を得ます．$u_{-1,1}=0$ ですから $u_{1,1}=-1/2$ となります．このようにもはやないはずの項 xy が再び現れたわけです．これは x^2y の積分が x と $y=e^{x^2}$ の有理式全体という関数族の中では閉形式の解を持たないことを示しています．

新しい手続きによる積分プログラムはこのような答えを返します．誤りを返すことはありません．$\log x$ の積分に対しては単に答え $x\log x-x$＋定数を返し，$y=e^{x^2}$ のとき x^2y の積分に対しては(15)を返して閉形式解はないといいます．(15)を返す理由は y の積分は x^2y の積分より簡単だということです．x^0 の指数は x^2 の指数より小さく，y の積分の方が役に立つからです．さてこういうとき「あなたは何ができますか？」と聞かれたら「新しい関数を導入します」と答えます．ほかの方法はないのですから．今の場合は $\int e^{x^2}dx$ を誤差関数と呼び，関数族を x と $y=e^{x^2}$ と $z=\int y\,dx$ の有理式全体，すなわち有理関数体に y と z を添加した体，に拡げてみます．x^2y の積分を解こうとするとき，2つの関数 x と e^{x^2} のほかに第3の関数である誤差関数 z があるわけです．このとき x^2y の積分は新しい関数族の中で

$$\int x^2y\,dx=\frac{1}{2}xy-\frac{1}{2}z$$

という閉形式解を持つことになります．これが新しい Risch の手続きのやり方です．

5. 代数的拡大その他

さて次の段階はもちろん有理関数体の代数的拡大について考察することです．この場合にも Risch の命題は成立しますが，ここで問題になるのは(7)の展開式を作り上げるような独立した関数の族をどうやって知るかということです．こういう展開式を成り立たせるどんな代数関数の族に対しても基底があるという良く知られた数学的事実があります．しかし特定の基底を求めるというのはほとんどの場合，非常に複雑な数学的問題です．多くの場合，存在しない数学的理論に入り込んでしまいます．問題の困難さは

$$\int\frac{dx}{\sqrt{x^2+1}}=\log(x+\sqrt{x^2+1})$$

という非常に簡単な例からも推測できます．「log の引数がこの形をしているのをどうして予測したのですか？ x はなぜ $2x$ や $3x$ ではなかったのですか？なぜ答えはこの項だけなのですか？ですからこれが代数関数に対する基底を構成するとき困難が生じる理由です．これは代数幾何学という分野の問題です．

現在この問題を解決しようとする試みが2つあります．1つは Cambridge 大学の Norman の学生の1人だった Davenport によるもので，もう1つは MIT の Trager によるものです．現在 Davenport のプログラムは相当広い関数の族に属する関数を積分することができます．平方根と立方根を扱うことはできますが4乗根を導入しようとすると理論は完全ではありません．彼のプログラムは1節の例題(c)と次の(d)，(e)のような問題を解くことができます．

(d) $\int\frac{\sqrt{x+\sqrt{a^2+x^2}}}{x}dx$

(e) $\int\frac{dx}{x\sqrt{x^3-3x^2+x+1}}$

(d)は閉形式の解を持ち，(e)は閉形式の解を持ちません．(d)を解くには，$x=\cosh y$ とおき，次に $t=e^{\frac{1}{2}y}$ とおけばうまくいきます．

さてプログラムを書くときのもう1つの問題は，新しい数学を実際に始めなければならないことです．J. Davenport は有名な数学者 Harold Davenport の息子です．彼は時間を3通りに使い分けています．1つは彼の父の友人であるイギリスの著名な数学者達との会話です．彼が幼少の時から顔見知りです．彼等の頭を完全に活用して数学的に分かっていることを聞き出そうとします．新しい事実を見つけるとそれをプログラムに直します．ですから彼のプログラムはある意味で代数幾何学という分野ではいかなる1人の数学者より優れているといえます．2つは，Cambridge 大学の図書館に行くことです．数学書に関しては，おそらく世界で最大の蔵書があると思います．ここで彼は前世紀に代数幾何学について書かれた本を片っぱしから目を通していきます．というのは大多数の読者の方はおそらくお気付きでしょうが，数学は20世紀に入ると何かが有るとか無いとかいう議論に熱中しはじめました．さて存在理論は非常に立派なのですが，何も計

算してくれません．19世紀の数学者は構成的な数学を作りました．定理の証明はひねくれた存在証明によるのではなく，Risch の手続きのようになにかを構成することによって証明します．Risch の手続きでは積分を構成することで存在を証明しました．それで彼は前世紀に書かれた本を眺めて，そこで行われていることを学びます．それから3つめの時間の使い方として，学んだことをプログラムに直します．今やプログラムは5,000行にもなり日々に増えています．ですからこの分野は難しいのですが進歩はしているのです．

Risch の手続きについてまだ完全には話をしていない問題があります．理論上ではなく実際上の問題です．前に tan は問題が生じるといいました．新しい手続きが周期関数を扱う方法によると，$\tan x$ と e^x は実数上で独立です．複素数上では $\tan x$ と e^x は独立ではありません．このときは $\cos x$ と $\sin x$ は e^x で表されるからです．実数上で独立ではありますが，微分してみると分かるように $\tan x$ と $\log x$ と e^x は同じではありません．$(\log x)^n$ を微分すると指数が1つ減り，$(e^x)^n$ の指数は変わりません．$(\tan x)'=1+(\tan x)^2$ ですから，$(\tan x)^n$ の場合は1つ増えます．ですから「この議論から $\tan x$ に関しては収束しない列を扱うことになりませんか？」という疑問が出ます。tan は次数の上限に関しては問題を発生させます．新しい手続きのプログラムの現在の版では，sin と cos の高次の項の計算はそれを tan に変換しているので非常に時間がかかるという計算上の困難さも生じます．

しかしもしM 180級の機械があれば積分(a)～(e)は1～2秒で計算できます．時間のかかる Davenport のプログラムで(d)を求めるのに1.4秒必要です．ですからこの位の機械があれば積分は問題ありません．まもなく M 180 級の性能を有する個人用機械が手に入るでしょう．そうすれば積分に要する時間など問題でなくなります．

以上が今日「数式処理による積分」が占める位置についての手短な概観です．もう1度内容を検討すると，今説明したばかりの新手続きを有する今日でも，多くの場合に型合わせで十分立派な仕事ができます．つまり今説明し終った計算機構を全く使う必要がないような積分の族があります．型合わせは掛値なしに役立つので使えます．決して見下してはなりません．まだ非常に重要な役割を演じています．実際アメリカの Utah 大学の S. Harrington は Risch-Norman のプログラムを核として，それが積分できない関数を扱う型合わせをその上においたプログラムを書きました．1978年の5月 Harrington はシベリアの Novosivirsk にいる1人の数学者から手紙を受け取りました．その手紙には彼とその仲間が，Gluschkov が設計した MNP 2（ミール）という機械で積分を計算しているということが書いてありました．この機械は多項式の演算を含む代数的手続きをマイクロ・プログラムではなく直接金物で実現しています．しかも10年も前に作られたということです．この機械で計算した積分の表を見ると Harrington は自分のプログラムはそのうちの3つを計算することができないことに気が付きました．それでもちろん彼はこの紳士に「確かに私のプログラムもあなたと同じ積分を計算できます」と返事を出す前に彼のプログラムを直しました．しかし代数的手続きを行うマイクロ・プログラムとか金物があれば便利であることは確かですし，積分プログラムの修正も容易でしょう．

こんな風に型合わせはなお重要な役割を持っていますが，しかし長期的に見ると手続きによる方法が，私達が関心を持つ積分をすべてその範囲に収めてゆくと思います．私達はまだ Bessel 関数や楕円関数を扱うには程遠い所にいます．しかしこの分野はなお進歩しており，したがって希望的観測によると5年かそこいらのうちに，たとえば物理学者や化学者が関心を持つほとんどの積分が私達のプログラムにより計算されることでしょう．それゆえ数式処理による積分はなお心踊る分野なのです．有難う！

後記：理化学研究所情報科学研究室の相馬嵩，井田哲雄，佐々木建昭，稲田信幸の皆様それから学生の村尾裕一，古川昭夫の諸君の助力がなければこの稿は完成しなかったと思います．心から感謝します．

積分の解答

A. Norman と J. Davenport が作成した積分プログラムを FACOM M 180 の REDUCE 2 上で動かした結果です．結果を微分すると元通りになりました．REDUCE 2 は Hearn が作成しました．Norman が作成した Cambridge 大学 LISP システム上で動いているものを用いました．

(a) $\displaystyle\int\frac{dx}{x^8-1}$ (0.76 秒)

$$
\begin{aligned}
&=\{-2\sqrt{2}\ \mathrm{atan}\,(\sqrt{2}x-1)\\
&\quad -2\sqrt{2}\ \mathrm{atan}\,(\sqrt{2}x+1)-4\ \mathrm{atan}\,(x)\\
&\quad +\sqrt{2}\log(-\sqrt{2}x+x^2+1)\\
&\quad -\sqrt{2}\log(\sqrt{2}x+x^2+1)+2\log(x-1)
\end{aligned}
$$

$-2\log(x+1)\}/16.$

(b) $\int x^7(\log x)^6 dx$ (0.13 秒)

$=x^8\{16384(\log x)^6-12288(\log x)^5+7680(\log x)^4$
$-3840(\log x)^3+1440(\log x)^2$
$-360\log x+45\}/131072.$

(c) $\int \frac{1}{x}\left(\frac{1}{\sqrt{x-1}}+\frac{1}{\sqrt{x-4}}\right)dx$ (0.64 秒)

$=[\, i\,\{-2\log(\sqrt{x-1}-i)+2\log(\sqrt{x-1}+i)$
$-\log(\sqrt{x-4}-2i)+\log(\sqrt{x-4}+2i)\}]/2.$

(d) $\int \frac{\sqrt{\sqrt{a^2+x^2}+x}}{x}dx$ (1.37 秒)

$=\log(\sqrt{\sqrt{a^2+x^2}+x}-\sqrt{a})\sqrt{a}$
$+\log(\sqrt{\sqrt{a^2+x^2}+x}-\sqrt{ai})i\sqrt{a}$
$-\log(\sqrt{\sqrt{a^2+x^2}+x}+\sqrt{a})\sqrt{a}$
$-\log(\sqrt{\sqrt{a^2+x^2}+x}+\sqrt{ai})i\sqrt{a}$
$+2\sqrt{\sqrt{a^2+x^2}+x}$

(e) $\int \frac{dx}{x\sqrt{x^3-3x^2+x+1}}$ (0.5 秒)

閉形式の解はない！

参 考 文 献

1) Andersen, C.: Use of Computerized Symbolic Integration in Finite Element Development, Proc. of the ACM National Conference, San Diego, Calif., pp. 554-562 (Nov. 1974).
2) Bierens de Hann, D.: Nouvelles Tables D'Intégrales Définies, Leide (1867).
3) Davenport, J.: The Computerisation of Algebraic Geometry, Proc. of EUROSAM '79, Marseille, France, pp. 119-133 (Jun. 1979).
4) Davenport, J.: Algorithms for the Integration of Algebraic Functions, Proc. of EUROSAM '79, Marseilles, France, pp. 415-425 (Jun. 1979).
5) Davenport, J.: Anatomy of an Integral, SIGSAM Bull. No. 52, pp. 16-18 (Nov. 1979).
6) Fox, J. and Hearn, A.: Analytic Computation of Some Integrals in Fourth Order Quantum Electrodynamics, J. Comp. Physics 14, pp. 301-317 (Mar. 1974).
7) Gentleman, W.: On Computing Certain Integrals and Implication for Symbolic Algebraic Manipulation, Proc. of the ACM National Conference Boston, pp. 840-843 (Aug. 1972).
8) Goursat, É.: Cours D'Analyse Mathematique, Tome I, pp. 241-264 (1927).
9) Hardy, G.: The Integration of Functions of a Single Variable (2nd. ed.) Cambridge Tract 2, C.U.P. (1916).
10) Harrington, S.: A New Symbolic Integration System in REDUCE, Utah Computer Report UCP-57 (1978).
11) Horowitz, E.: Algorithms for Symbolic Integration of Rational Functions, Ph. D. Diss., Univ. of Wisconsin, Madison (1969).
12) Horowitz, E.: Algorithms for Partial Fraction Decomposition and Rational Function Integration, Proc. of 2nd Symp. on Symbolic and Algebraic Manipulation, pp. 441-457 (Mar. 1971).
13) Jefferys, W.: Automated, Closed Form Integration of Formulas in Elliptic Motion, Celes. Mech., pp. 390-394 (Mar. 1971).
14) Kerner, W. and Grimm, R.: Magneto-hydrodynamic Spectra for Tokamaks with Non-circular Cross Sections, Proc. of 7th Conference on Numerical Simulation of Plasmas, Courant Institute, New York (Jun. 1975).
15) Liouville, J.: Mémoire sur la classification des transcendentes, et sur l'impossibilité d'exprimer les racines de certaines équations en fonction finie explicite des coefficients, Journal de Math. 1(1836), 2(1837), 3(1838), 4(1839), 5(1840).
16) Liouville, J.: Mémoire sur la Détermination des Intégrales dont la Valeur est Algébrique, Journal de l'École Polytechnique 14, cahier 22, pp. 124-193 (1833).
17) Mack, D.: On Rational Integration, Univ. of Utah, Comp. Phys. Report, UCP-38 (1975).
18) Maison, D. and Petermann, A.: Subtracted Generalized Polylogarithms and the SINAC Program, Comp. Phys. Comm. 7, pp. 121-134 (Mar. 1974).
19) Moses, J.: Symbolic Integration, Doc. Diss., MIT (1967).
20) Moses, J.: The Integration of a Class of Special Functions with the Risch Algorithm, Memo. MAC-M-421, (1969). SIGSAM Bull., Issue No. 31, ACM, pp. 14-27 (Dec. 1972).
21) Moses, J.: Symbolic Integration, The Stormy Decade, Comm. ACM, 14, pp. 548-560 (1971). Proc. of 2nd Symp. on Sym. and Alg. Manipulation pp. 427-440 (Mar. 1971).
22) Moses, J.: An Introduction to the Risch Integration Algorithm, ACM Annual Conf., pp. 425-428 (1976).
23) Moses, J. and Zippel, R.: An Extension of Liouville's Theorem, Proc. of EUROSAM '79, Marseille, France, pp. 426-430 (Jun. 1979).
24) Ng, E. and Polajnar, D.: A Study of Alternative Methods for the Symbolic Calculation of Elliptic Integrals, Proc. of SYMSAC '76,

New York, pp. 372-376 (Aug. 1976).
25) Ng, E.: Symbolic Integration of a Class of Algebraic Functions, Proc. of EUROSAM '74, SIGSAM Bull. Vol. 8, No. 3, NASA Tech. Memo. pp. 33-713 (Aug. 1974).
26) Norman, A. and Moore, P.: Implementing the New Risch Integration Algorithm, 4 th Int. Coll. on Adv. Comp. Meth. in Theoret. Phys., Marseille, France (1977).
27) Norman, A. and Davenport, J.: Symbolic Integration-The Dust Settled?, Proc. of EUROSAM '79, Marseille, France, pp. 398-407 (Jun. 1979).
28) Petit, B.: Tables of Indefinite Integrals, Dover, New York (1961).
29) Risch, R.: Symbolic Integration of Elementary Functions, Proc. of 1968 Summer Institute on Sym. Math. Computation, IBM, pp. 133-148 (1968).
30) Risch, R.: The Problem of Integration in Finite Terms, Trans. AMS, Vol. 139, pp. 167-189 (1969).
31) Risch, R.: The Solution of the Problem of Integration in Finite Terms, Bull. AMS, Vol. 76, pp. 605-608 (1970).
32) Risch, R.: A Generalization and Geometric Interpretation of Liouville's Theorem on Integration in Finite Terms, IBM Research Report RC 4834 (May 1974).
33) Risch, R.: Integration of Implicitly Elementary Functions, IBM Tech. Rep. RC-5406 (1975).
34) Ritt, J.: Integration in Finite Terms, Liouville's Theory of Elementary Methods, N. Y. Columbia Univ. Press (1948).
35) Rosenlicht, M.: On Liouville's Theory of Elementary Functions, Pacific J. of Math. 65, pp. 485-492 (1976).
36) Rothstein, M.: Aspects of Symbolic Integration and Simplification of Exponential and Primitive Functions, Doc. Diss., Univ. of Wisconsin, Madison (1976).
37) Slagle, J.: A Heuristic Program That Solves Symbolic Integration Problems in Freshman Calculus, Symbolic Automatic Integrator, SAINT, Doc. Diss., MIT (1961). J. of ACM, 10, pp. 507-520 (1963).
38) Stoutemyer, D.: A Diminutive REDUCE Program for Symbolic Integration, Univ. of Utah, Comp. Phys. Group Report, UCP-35 (1975).
39) Trager, B.: Algebraic Factoring and Rational Function Integration, Proc. of SYMSAC '76, New York, pp. 219-226 (Aug. 1976).
40) Trager, B.: Integration of Single Radical Extensions Proc. of EUROSAM '79, Marseille, France, pp. 408-414 (Jun. 1976).
41) Wang, P.: Evaluation of Definite Integrals by Symbolic Manipulation, Ph. D., Thesis, MIT, Project MAC (1971).
42) Wang, P.: Symbolic Evaluation of Definite Integrals by Residue Theory in MACSYMA, Proc. of IFIP Congress 74, pp. 823-827 (1974)
43) Yun, D.: Fast Algorithm for Rational Function Integration, IBM Research Report, RC 6563 (Jan. 1977).

(昭和 56 年 3 月 11 日受付)