3X-5

数式処理システムREDUCE3への和分機能のインプリメント

山本　潔，岸本一男，翁長健治
広島大学工学部

1．はじめに

級数の和の計算は解析学における最も基本的な計算の一つである。従って、数式処理処理システムの上でも、和分機能が基礎的な位置の一角を占めて良いはずであるが、現在のところ、記号微分や記号積分等と比較すると記号和分のインプリメントは遅れている[3]。本研究では数式処理システム REDUCE 3.2 [2]上へ、Gosper のアルゴリズム[1]に基づく和分機能をインプリメントしたので報告する。

2．不定和分と計算アルゴリズム

不定和分とは積分における不定積分に対応するもので、数式 a_n の不定和分 S_n が求まると、数式 a_n の有限項の和は

$$\Sigma_{i=1}^{n} a_i = S_n - S_0 \qquad (1)$$

で与えられる。初等関数の不定和分は必ずしも初等関数解であるとは限らない。また、不定和分が初等関数解になるか否かアルゴリズミックに確認するには、現在のところかなりの困難がともなう。本研究では適用範囲はかなり狭くなるが、a_n / a_{n-1} が有理式である場合に適用できる Gosper のアルゴリズム[1][3]を用いて和分を実現した。このアルゴリズムは次の Gosper の結果に基づいている。

n の関数 a_n の和分を $S_n = \Sigma_i a_i$ とし、S_n / S_{n-1} がnの有理式であるとする。nの多項式 p_n、q_n、r_n を

$$\frac{a_n}{a_{n-1}} = \frac{p_n \cdot q_n}{p_{n-1} \cdot r_n} ,$$

$$gcd(q_n , r_{n+j}) = 1 \ , \ j \in N \qquad (2)$$

によって定義するとき、a_n の不定和分 S_n は

$$S_n = \frac{q_{n+1}}{p_n} \cdot a_n \cdot f(n) \qquad (3)$$

で与えられる。ただし、f(n) は

$$p_n = q_{n+1} \cdot f(n) - r_n \cdot f(n-1) \qquad (4)$$

を満たす多項式である。

[3] による Gosper のアルゴリズムは

1）　a_n から（2）をみたす多項式 p_n, q_n, r_n を求める；
2）　条件（4）をみたす多項式 f(n) の次数の上限を求める；
3）　条件（4）をみたす多項式 f(n) を未定係数法で求める；
4）　多項式 p_n, q_n, f(n) と a_n を式（3）に代入して和分の結果を求める。

の手順で実行される。

このアルゴリズムは n の多項式に対する未定係数法による不定和分の拡張になっており、例えば4節の図1の行番号12、13、14、16に示すように有理式の和分や $\Sigma_n a^n$ といった等比級数の和分も計算可能である。

3．インプリメンテーション

本研究では、和分機能を附加する数式処理システムとして REDUCE を取り上げた。REDUCE はユタ大学の A.C.Hearn によって開発された、ALGOL 風の Lisp 言語 R-LISP で書かれた汎用の数式処理システムである。現在、広島大学総合情報処理センターのHITAC M-200 H 上でVersion 3.2 が稼働中である。REDUCE は記号積分、記号微分、因数分解、行列計算等のかなり高級な処理能力を持っているが記号和分の機能は持っていなかったものである。この REDUCE 中で不定和分を実行するには本研究で付加した関数：

SUMM (<EXPR>, <VAR>) ;

を呼び出せばよい。この SUMM は式 <EXPR> を変数

Implementation of Gosper's Summation Algorithm in Algebraic System REDUCE3
Kiyoshi YAMAMOTO, Kazuo KISHIMOTO, Kenji ONAGA
HIROSHIMA Univ.

<VAR> による関数と見なし和分を計算し、その値として返す。式 <EXPR> 中に和分の計算ができない部分が含まれている場合には、その和分が計算できない部分だけ関数 SUMM の形のまま残して、計算できる部分に関してのみ和分を実行する。これは他の不定積分機能などの場合と同様である。

実際に Gosper のアルゴリズムに従って計算するためには

1） 式（1）をみたす多項式 p_n、q_n、r_n を求めるために多項式最大公約数の計算
2） 式（1）をみたす多項式 p_n、q_n、r_n を求めるために多項式終結式の計算
3） 式（4）をみたす多項式 f(n) の係数を未定係数法によって求めるための連立方程式の解法
4） 多項式、有理式の四則演算、代入などの基本的演算

などが必要となってくるが、それらは REDUCE の Module を呼び出すことにより実現した。

ところで、和分の実際の計算に際しては、入力された式 <EXPR> 全体に対して Gosper のアルゴリズムを適用したのでは和分できる項とできない項とが分離できない。そこで、REDUCE 内部では Standard-Quotient と呼ばれる有理式の形をした標準型で式全体を表しているが、和分計算ではまずその有理式を次のように分解してから和分を実行している。

1） 有理式を多項式の部分と"真分数式"の部分とに分離する；
2） 多項式を単項式に、"真分数式"を分子に関して単項式にそれぞれ分解する；
3） 分解した各項に対して Gosper のアルゴリズムを適用する。

4．実行結果

HITAC M-200H上での実行結果とその計算時間を図1に示す。行番号１１は多項式の和分を計算、行番号１２、１３は有理式の和分を計算した例である。行番号１２、１３、１４、１６は多項式に対する単純な未定係数法等の方法では計算できない例である。また、行番号１５は和分のできない項を含んだ式の計算例であり、計算できない１／ｎはそのまま関数 SUMM(1/n,n) の形のまま残される。

計算時間は一回の Gosper のアルゴリズムにかかる実行時間と Gosper のアルゴリズムを実行する回数である式の項数の積になる。行番号１１、１５、１６は項数が多いため計算が長くかかっている。

```
11: SUMM(3*N**3+5*N**2+N+8,N);
    (N*(9*N^3 + 38*N^2 + 45*N + 112))/12
    TIME: 436  (GBC: 96) ms
12: SUMM(1/N-1/(N+1),N);
    N/(N+1)
    TIME: 117  (GBC: 95) ms
13: SUMM(1/(N*(N+1)*(N+2)),N);
    (N*(N + 3))/(4*(N^2 + 3*N + 2))
    TIME: 146  (GBC: 0) ms
14: SUMM(N*A**N,N);
    (A*(A^N*A*N - A^N*N + A^N + 1))/(A^2 - 2*A + 1)
    TIMES: 129  (GBC: 0)ms
15: SUMM(N+5+1/N,N);
    (2*SUMM(1/N,N) + N^2 + 11*2)/2
    TIMES: 196  (GBC: 0) ms
16: SUMM(3*N**2+5+N*A**N,N);
    (2*E^N*E^2*N-2*E^N*E*N-2*E^N*E+2*E^2*N^3+
     3*E^2*N^2+11*E^2*N-4*E*N^3-6*E*N^2-22*E*N+
     2*E+2*N^3+3*N^2+11*N)/(2*(E^2-2*E+1))
    TIMES: 183 (GBC: 97) ms
```

図　1　実行結果

5．終わりに

本研究では、Gosper のアルゴリズムを用いて数式処理システム REDUCE に和分機能の付加を行った。３節末での単項式への分解は必ずしも最適なものではなく、項を適当に組み合わせれば Gosper のアルゴリズムで解けるはずの式でも、本稿の方式では解けない場合がある。従って、パターン認識と記憶した特殊解を操作する方法、Heuristic あるいは総当たり的な方法等を組み合わせて用いることにより和分可能な式の範囲を広げることができると思われるがそれは今後の課題である。

参考文献

[1] R. W. Gosper, Jr., "Decision procedure for indefinite hypergeometric summation", Proc. Nat. Acad. Sci. USA 75/1, pp.40-42, 1978.
[2] A. C. Heran, "REDUCE USER' MANUAL", The Rand Corp., 1984
[3] L. C. Lafon, "Summation in finite terms", Computer Algebra Symbolic and Algebra Comptation, Computing Suppl. 4, pp.71-77, 1982.