数理解析研究所講究録
第 685 巻 1989 年 81-89

ハイブリッド処理システム SYNC とアルゴリズム選択問題

浅川 秀治、野田 松太郎
(Shuji Asagawa, Matutaro Noda)
(愛媛大学・工)

1. アルゴリズム選択問題

数値計算の特色の一つに膨大な量のライブラリの充実がある。しかし、問題を解く場合にライブラリ中のアルゴリズムの種類が多すぎ、どれを使用すべきかがわからなくなる。しかも、問題の性質によって有効なものと有効でないものとがある。そこで、これらを整理し、与えられた問題に適応的にライブラリを選択しようという試みが多くなされている[1]。さらにアルゴリズムの選択自体を研究材料とすることも考えられている[2]。この場合のアルゴリズムの良さを示す尺度として、計算速度や計算精度がとられるのが通常である。このため、アルゴリズム選択問題では伝統的に数値計算アルゴリズムのみが興味の対象とされてきた。しかし、多くの悪条件問題でも知られているように、数値計算のみでは、得られた結果に対する信頼性が十分ではない。そこで、いかに誤差保証をすべきか、いかに正確計算の知識を導入すべきか等に関する研究が重要になりつつある。アルゴリズム選択に対しても従来の尺度以外に、数式処理との結合を重視する研究が広く行われつつある[3]。このような数値数式融合アルゴリズム(ハイブリッドアルゴリズム)の開発は、ハイブリッド処理システムが広く用いられ始めた現在、そのようなシステム上での新しいアルゴリズム選択には欠かすことができない。

本論では、数値計算アルゴリズムの世界に数式処理の技法を取り入れることの必要性、特にハイブリッドアルゴリズムの有効性を積分計

算について考察する。また、数式処理で確立された積分アルゴリズムのハイブリッド化による、新しい算法を提唱し、具体的な計算例も述べる。アルゴリズム開発はハイブリッドシステムへの移植が容易なように行なわれる。

2．数値計算だけで大丈夫なのか

数値計算は確かに高速で、現在までは問題もなく過ぎている。しかし、精密計算の要が生じる場合には必ずしも安閑としてはおられない。悪条件代数方程式の解はもちろんのこと、振動積分、特異点を含む積分など数値計算だけでは十分でない問題が多く存在する。以下に被積分関数が激しく振動する場合の具体的な数値例を示す。数値解法の中、GAUSS-k は分点数 k のガウス型公式、DE は２重指数型公式をいう。

$$\int_0^{2\pi} x \sin(50x) \cos(nx)\, dx$$

n	Exact	Gauss-32	Romberg	DE
1	0.00251428	2.156186	0.00251430	1.63901971
2	0.00503460	0.8756967	0.00503465	-0.27939798
4	0.01011785	-0.0974224	0.01011795	0.70417174
10	0.02617994	0.1058508	0.02618020	1.15139973
20	0.05983986	-0.8128637	0.05984051	0.05596205
30	0.11780972	-0.6453440	0.11781106	0.50006480

なお、この場合は表にはないが適応型積分公式も Romberg積分とほぼ同じ結果を与える。通常良好な解を与える DE 公式は、この例では適用不可能である。従って、このような問題のクラス（被積分関数が激しく振動する積分）に対してはアルゴリズム選択の尺度を精度におく限り、Romberg積分公式または適応型公式が最適アルゴリズムとなる。しかし、Romberg 積分公式もまた必ずしも万能ではない。積分の真値は、表では Exact と表わされている。この例の場合では、真値は数式

積分を遂行し、積分の上・下限を代入して求める。当然ながら、いかなる数値解法より安心出来る解が得られる! 次の例では DE 公式と適応型公式(Adaptive)は非常に近い結果を与えるが、Romberg 積分や Gauss 積分公式の結果は大きく異なっている。なお、表中の Exact は再び積分の真値を表わす。

(a) $$\int_0^1 \frac{dx}{(x+0.0001)(x+1.0001)+0.0001}$$

Exact	Gauss-10	Romberg	DE	Adaptive
7.8257613	5.1224142	8.4281843	7.8257612	7.8257612

ところが、文献4で述べられている

$$\int_0^1 \log(1-\cos(2x))\,dx \doteqdot -1.420293$$

の計算は全ての数値計算アルゴリズムでは結果を得ることができない。REDUCE等の数式処理システムを用いても記号積分もできない。上の数値は MAPLE によって得たものである。ここでは、特異点除去の過程で有効に数式処理システムの級数展開機能が用いている。詳細は文献4にある。

3. 数式処理の取入れ方

2の例でもわかるように、積分計算においてもアルゴリズム選択の尺度に信頼性を取り入れる限り数式処理の活用の可能性を避けることはできない。少なくともかなりの問題に対しては数式処理を利用すると、当然ながら信頼性は大いに高まる。積分計算への数式処理の利用方法は次にあげるような、いくつかのレベルに分けられる。

①最も単純な方の数式処理の利用は、与えられた被積分関数の積分

可能性をまず調べるだけ良い。積分可能なら数式処理システムへ制御を渡し、不可能なら数値積分公式に対する尺度によって最適なアルゴリズムを見いだす。

②より進んだ方の利用の例は特異点を含んだ被積分関数に対して見いだされる。すでにこの一つの例は2で示した。Taylor展開を有効に利用したアルゴリズムが発表されている[4]。被積分関数の特異性を変数変換によって解消し、以後は積分可能なら数式処理で、不可能なら数値計算で処理するような手法も容易に考えられる。

③数式処理で開発されたアルゴリズムの適用範囲を拡大し、誤差を考慮しつつ浮動小数係数の場合にも使用し得るようにする。

等である。③がアルゴリズムのハイブリッド化といえる。このようなアルゴリズムの実行には、ハイブリッド処理システムの存在が重要となる。ハイブリッドアルゴリズムの例は、積分解法ではないが、悪条件代数方程式の解法として提唱された「近似的ＧＣＤ」算法がある[5]。

数式処理的技法を積分に取り入れると不定積分を得ることができるという大きな利点がある。このため、積分可能な問題に対しては非常に正確な定積分の値も得ることができる。

4．数式処理積分のアルゴリズム

数式処理で積分計算を行うためのアルゴリズムは、本来は Risch のアルゴリズムに基礎を置くが、Davenport 等によると次のようにまとめられている[6],[7]。今、A(x)、B(x) を有理係数多項式とし、A(x) の次数 (deg(A)) < deg(B) とする。Horowitzのアルゴリズムによれば、

$$\int \frac{A(x)}{B(x)}\,dx = \frac{T(x)}{U(x)} + \int \frac{q(x)}{r(x)}\,dx$$

と有理項と超越項への分解が可能になる。ここで、$\deg(T) < \deg(U)$ である。超越項は $r(x)$ の因数分解により、

$$\int \frac{q(x)}{r(x)} dx = \sum c_i \log v_i$$

と積分できる。一方、c_i，v_i は終結式理論を用いて求め得る。この過程の詳細は文献6に述べられているが、その概要は次の通りである。

1. $q - yr'$ と r の終結式 $\mathrm{Resul}_x(q-yr', r)$ を求める。
2. y の多項式 $\mathrm{Resul}_x(q-yr', r) = 0$ を満足する $y = c_i$ を求める。この過程は因数分解または代数的求根による。
3. 各 ck に対し、関係式

   $$\gcd(q - c_k r', r) = v_k$$

   より v_k を求める。

ここでの問題点は以下の点にある。

2.の過程は y に関する代数方程式を解くことになるが、求める根の制約は大きい。$r(x)$ が因数分解可能な場合とか、きわめて低次である場合に限定されるのが現実である。代数方程式の求根のための有効なアルゴリズムの大半は、実数計算に関連している。しかし、ここではそのようなアルゴリズムを用いることができない。実際、例え浮動小数の根を求めたとしても、3.の過程での gcd 計算が不可能になる。

当然、より広い範囲の被積分関数を積分するためには、上の数式積分の過程で巧妙に浮動小数計算を導入し、数値・数式融合アルゴリズム（ハイブリッドアルゴリズム）を構築する必要がある。

5. ハイブリッド積分アルゴリズムと計算例

<u>4</u>で述べた数式処理アルゴリズムのハイブリッド化を考える。当然、yに関する代数方程式の根を高精度な数値解に置き換える。これによって、代数方程式が複雑になっても、すなわち $r(x)$ が因数分解不可能でも浮動小数係数を持っても求根過程は遂行され得る。しかし、ここで得られた根は一般には浮動小数であるので、上のアルゴリズムの３.の過程の gcd 計算が変更されねばならない。これに対し悪条件問題に対して研究された近似的ＧＣＤ算法を用いる。このハイブリッドアルゴリズムでは、１変数代数方程式を考えた場合、近接根分離のための gcd計算が浮動小数係数に拡大されている。計算は誤差を考慮して進められるので、得られた結果に対する保証もされる。これらによって、<u>4</u>のアルゴリズムを次のように書き直す。

1. $q - yr'$ と r の終結式 $\mathrm{Resul}_x(q-yr', r)$ を記号的に求める。
2. y の多項式を $Y(y) = \mathrm{Resul}_x(q-yr', r)$ とする。
3. $Y(y) = 0$ の根がすべて実根である場合に５．以下を行なう。
4. $Y(y) = 0$ の根が一つでも虚根であるときは積分不能*)。
5. $Y(y) = 0$ を満足する $y = c_i$ を高精度のニュートン法によって数値的に求める。
6. 各 c_k に対し、関係式

   $$\gcd(q - c_k r', r) = v_k$$

   より v_k を求める。この gcd計算では「近似的ＧＣＤ」算法を用いる。

現状では、実数のみを取り扱っているのでこの制約が加わると思われる。

アルゴリズム中の数値計算で混入する誤差はニュートン法と近似的GCDの2種の計算のみから生じる。しかし、前者には厳密な誤差評価が存在し、後者もその特色として計算過程で誤差を評価しえる。一般の数値積分におけるような誤差不定性の問題は生じない。もちろん、不定積分を求め得る利点がある。

このアルゴリズムを用いたハイブリッド積分の実行例を以下に示す。表にはハイブリッド積分の結果に加えて、MAPLE 上の数式処理積分および適応型公式による数値積分の結果を示す。被積分関数として<u>2</u>で考えた(a)に加えて次の2つを考える。

(b) $$\int_0^1 \frac{dx}{(x+1)(x+2)(x+3)+0.00001}$$

(c) $$\int_0^1 \frac{5x^4+60x^3+255x^2+450x+275}{x^5+15x^4+85x^3+225x^2+274x+121}\,dx$$

例題	ハイブリッド積分	数式積分	数値積分
(a)	7.8257612959	impossible	7.8257612958
(b)	0.08494943438	2662.226436	0.08494943437
(c)	1.7884621990	impossible	1.7884621990

数式積分は、(b)でおこしな結果を出力する以外は積分不能である。MAPLE のかわりに MACSYMA や REDUCE を用いても、すべての問題が積分不能になる。数値積分としては、アルゴリズム選択の対象として適用範囲の広い適応型公式がこれらの例でも優れているが、ハイブリッド積分結果もこれと十分に対抗できることが分かる。もちろん、ハイ

ブリッド積分では不定積分が求まる利点が数値計算に比較して大きい。例えば例題（b）に対する不定積分は次のようになる。

```
-1.00000000030 log ( 1.99999000000 + 1.00000000000 x )
+.499992500150 log ( 3.00000499996 + 1.00000000000 x )
+.500007500150 log ( 1.00000500004 + 1.00000000000 x )
```

5. まとめ

本論では新しいハイブリッド積分のアルゴリズムを提案し、従来からのアルゴリズム選択問題の対象を広げることを考えた。数式処理の積分アルゴリズムをニュートン法と近似的ＧＣＤ算法によって書き換えたハイブリッド積分の特長は以下のようにまとめるられる。

①数式処理の有理式上での積分機能を完全に包含し、かつ数式処理で不可能な積分も行える。

②解の精度が十分に保証される。

③数値積分計算では不可能な不定積分も求めることができる。

また、ここで議論したアルゴリズムは主に SUN-4 上の MAPLE で開発されたが、小型ハイブリッドシステム SYNC[8] の持つ機能はその全てを包含するので、移植は容易である。

このハイブリッドアルゴリズムの有理数体から実数体への拡大の厳密な証明は現時点では未解決な問題である。また、ハイブリッド積分アルゴリズムでも示したが、積分可能性についてのより深い検討が終結式理論にもとずきなされる必要がある。

参考文献

1) K.Schultze and C.W.Cryer, NAXPERT:A prototype expert system for numerical software, SIAM J.Sci.Stat.Comput., vol.9, no.3, pp.503-515(1988).
2) E.N.Houstis, J.R.Rice, C.C.Christara and E.A.Vavalis, Performance of scientific software, in "Mathematical Aspects of Scientific Software"(ed. by J.R.Rice), Springer-Verlag, pp.123-155(1988).

3) M.T.Noda and S.Asagawa, Algorithm Selection Problem in the Hybrid Computation, Proc. Conference on Expert Systems for Numerical Computing, 1988, pp.56-59.
4) K.O.Geddes, Numerical integration in a symbolic context[1], Proc.SYMSAC'86, pp.185-191.
5) T.Sasaki and M.T.Noda:Approximate square-free decomposition and root-finding of ill-conditioned algebraic equations, J.Inf.Proc., to appear.
6) J.H.Davenport, Y.Siret and E.Tournier, 'Computer algebra', Academic Press, 1988.
7) J.H.Davenport, Integration in closed form, in "Computers in mathematical research"(ed. by N.M.Stephens and M.P.Thorne), Clarendon Press, pp.119-134(1988).
8) 野田 松太郎、岩下 英俊，パーソナルなハイブリッド処理システム SYNC の設計，情報処理，vol.30, No.4, 1989(掲載予定).