

D 加群のアルゴリズムと その数値解析への応用*

神戸大学・理学部数学 高山信毅 (Nobuki Takayama)
Department of Mathematics,
Kobe University

ここでは多変数超幾何関数関数の数値計算の問題を考えたい。数値計算には超幾何微分方程式の差分化と級数解の計算が重要になる。超幾何と差分と聞くと、離散可積分系を連想される方もいるかもしれないが、我々の研究は残念ながらそこまでは到達していない。ここでは数値解析の伝統的な方法と微分作用素環のアルゴリズム (D 加群のアルゴリズム) を組み合わせることにより、まずまずの数値計算ができることを報告する。ここで紹介する方法は単純な方法であるが、まだあんまり研究されてない分野である。

この研究の動機の一つは、“デジタル公式集” のオーサリングツールへの応用である。それについても最後にすこし触れたい。

超幾何関数って何?

(パラメータ付きの) 指数関数や巾関数の (定) 積分で定義される関数を超幾何関数と呼ぶ。

例.

$$F(a, b, c; x) = \int_0^1 t^{b-1} (1-t)^{c-b-1} (1-tx)^{-a} dt$$

例.

$$F(\beta_1, \dots, \beta_d; x_1, \dots, x_n) = \int_C \exp(f(x, t_1, \dots, t_d)) t_1^{-\beta_1-1} \dots t_d^{-\beta_d-1} dt_1 \dots dt_d$$

ここで f は x と t の多項式。たとえば $f(x, t) = x_1 t_1^2 + x_2$ など。

巾指数に現れるパラメータはギリシャ文字で、多項式の係数としてあらわれるパラメータは x で書く。

超幾何関数と微分方程式の関係は?

*英語題名: Algorithms for D -modules and their applications to numerical analysis. この文章は講演でもちいた OHP に加筆したものであるため、文章としては不自然な部分があるかもしれない。御容赦願いたい。

定理. 超幾何関数は x についての holonomic 系を満たす. つまり超幾何関数は n 変数の多項式係数連立線形偏微分方程式系の解であり, その方程式系の主シンボルの生成するイデアルは n 次元 (方程式が十分たくさんあり).

証明: 超幾何関数の積分核は, holonomic D -加群の構造をもつ. D -加群の一般論より, “holonomic D -加群の積分はまた holonomic D -加群である”. Q.E.D.

“...” の部分の証明についてはたとえば Björk の Rings of differential operators の Chapter 1 を見よ. 超幾何関数のみならず holonomic 系を実際に構成する問題に関しては “Toshinori Oaku, Yoshinao Shiraki, Nobuki Takayama, Algebraic Algorithms for D-modules and Numerical Analysis, Z.M.Li, W.Sit (editors), Computer Mathematics, Proceedings of the sixth asian symposium, 23–39, World scientific, 2003” を参照されたい.

例.

$$F(x_1, x_2) = \int_C \exp(x_1 t + x_2 t^2) t^{-\beta-1} dt$$

$$(x_1 \partial_1 + 2x_2 \partial_2 - \beta) \bullet F = (\partial_1^2 - \partial_2) \bullet F = 0$$

D って何?

D を n 変数の微分作用素環:

$$D = \mathbf{C}\langle x_1, \dots, x_n, \partial_1, \dots, \partial_n \rangle$$

とする. ここで

$$x_i x_j = x_j x_i, \partial_i \partial_j = \partial_j \partial_i, \partial_i x_j = x_j \partial_i + \delta_{ij}.$$

D は次のように “関数の空間” に作用する.

$$x_i \bullet F(x) = x_i F(x), \partial_i \bullet F(x) = \frac{\partial F}{\partial x_i}.$$

例. $\partial_1 x_1 = x_1 \partial_1 + 1, \partial_1 x_1^2 = x_1^2 \partial_1 + 2x_1.$

線形偏微分方程式系 $L_1 \bullet f = \dots = L_m \bullet f = 0$ を D の 左イデアル $I = DL_1 + \dots + DL_m$ とみなす.

どうして数値計算が興味深いのか?

関数を数値計算することはだれでも分かる問題ではあるが, 基本的でかつ奥が深い問題である. したがって当然研究する値打ちがあると考えている. さらに次のような応用も考えている.

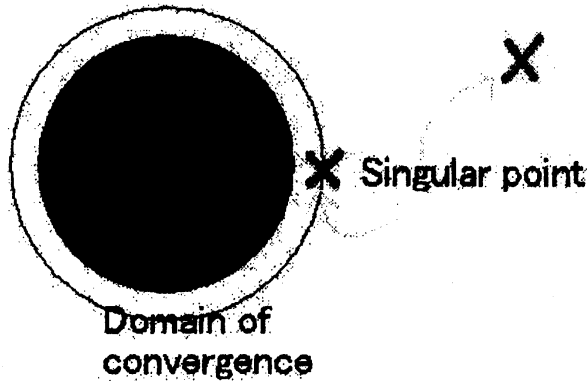
- (1) デジタル公式集のオーサリングツールで入力した公式のチェックを行う.
- (2) モノドロミ群の数値解析的な手法による決定に使えるかもしれない.
- (3) デジタル信号処理などの工学に応用できるかも (この話題については上記, Oaku, Shiraki, Takayama の論文も参照されたい).

計算方法の概略は?

積分表示ではなく微分方程式を調べることにより関数の情報を得ようというのが我々の大方針である。

われわれの方法の概要

- 超幾何関数の満たす微分方程式系を見付ける。
- (その微分方程式系の級数解を構成する。)
- みつけてきた微分方程式系は 連立線形偏微分方程式系である。級数解が収束しないところでは、ある道に沿って微分方程式を数値解析で解く。我々の方程式に (adaptive) Runge-Kutta 法などの数値解析の伝統的方法を適用できるようにするには、変形が必要。



さてこの変形をやるのが グレブナ基底

グレブナ基底の概念で大切なのは initial term と initial ideal の概念である。

Initial term と Initial ideal

- $\ell \in D$ および $u, v \in \mathbb{R}^n, u_i + v_i \geq 0$ に対して

$$\text{in}_{(u,v)}(\ell)$$

は、weight (u, v) に対して最高次の項の和。これを initial term とよぶ。

例.

$$\text{in}_{(0,0,1,1)}(\partial_1^2 + \partial_2 + \partial_1 + x_1^2 + x_2^2) = \xi_1^2 \in \mathbb{C}[x, \xi]$$

$$\text{in}_{(-1,-1,1,1)}(x_1\partial_1 + 2x_2\partial_2 - 3) = x_1\partial_1 + x_2\partial_2 - 3 \in D \quad (u + v = 0).$$

線形偏微分方程式系 $L_1 \bullet f = \dots = L_m \bullet f = 0$ を D の 左イデアル $I = DL_1 + \dots + DL_m$ とみなすのであった。例をひとつあげよう。

例. $n = 2, I = D(x_1\partial_1 + 2x_2\partial_2 - 3) + D(\partial_1^2 - \partial_2)$.

この initial term の考え方は解析で昔から用いられている主シンボル $\sigma(L)$ の考えの拡張である. 次のイデアルを I の weight (u, v) についての initial ideal とよぶ.

$$\text{in}_{(u,v)}(I) = \mathbf{C} \cdot \{\text{in}_{(u,v)}(\ell) \mid \ell \in I\}$$

グレブナ基底 (involutive basis) って何?

$G = \{g_1, \dots, g_p\}$ が I の weight (u, v) についてのグレブナ基底とは

(1) I は G で生成される. (2) $\text{in}_{(u,v)}(I)$ は $\text{in}_{(u,v)}(G)$ で生成される.

どうやって計算するの? \Rightarrow (0) “新しい元” S-poly を生成する. (1) 割算アルゴリズムで簡略化. (2) Buchberger criterion (いつ止めるか判定する方法).

たとえば上の例を weight $(0, 0, 1, 1)$ で考えるとこの計算は次のようになる.

$$L_1 = x_1 \partial_1 + 2x_2 \partial_2 - 3, \quad L_2 = \partial_1^2 - \partial_2$$

$$L_1 + 2x_2 L_2 = x_1 \partial_1 - 3 + 2x_2 \partial_1^2 =: L_3$$

$L_1 + 2x_2 L_2$ は L_1 と L_2 の S-poly である. L_1, L_2, L_3 に Buchberger criterion を適用するとこれらは criterion を満たしてグレブナ基底であることがわかる.

例. 上の例の I を考える (ちなみにこれは $(1, 2)$ -超幾何方程式). いまやった計算より

$$\sigma(I) = \text{in}_{(0,0,1,1)}(I) = \langle x_1 \xi_1 + 2x_2 \xi_2, \xi_1^2 \rangle$$

さて大事なことは

Weight (u, v) を変えると色々な基底が現れる

ことである. この性質を用いて欲しい性質のイデアルの生成元を得る. 行列を上三角行列に変形していく操作は同じたぐいの計算である.

どうやって数値解を求めるのか? (多項式系の場合の復習)

$$f_1(y_1, \dots, y_n) = 0$$

...

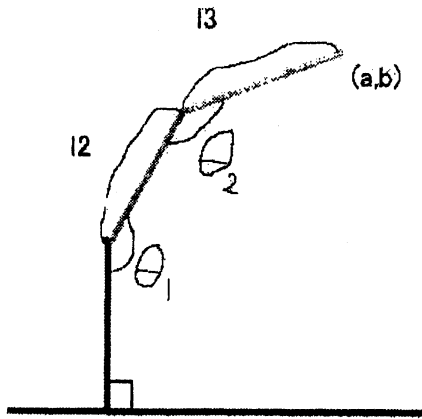
$$f_p(y_1, \dots, y_n) = 0$$

J を f_1, \dots, f_p で生成された $S_n = \mathbf{Q}[y_1, \dots, y_n]$ のイデアルとする.

定理. (WN=Well-known) J が S_n において zero-dimensional なら (i.e. $\dim_{\mathbf{Q}} S_n/J < +\infty$), $J \cap \mathbf{Q}[y_i]$ は 0 でない元を含む (つまり 1 変数 y_i についての代数方程式を含む). この元は, y_i 以外の元の weight を 1 に, y_i の weight を 0 にした順序で, J のグレブナ基底を計算すれば求まる.

この定理を基礎にして一変数代数方程式の数値解を求める方法がある。つまり上の定理で存在と構成が保証されている1変数代数方程式をまず数値的に解いてその解をもとに他の解も数値的に決めてゆくわけである(行列の三角化の類似)。

例. Cox, Little, O'Shea の入門書 "Ideal, Varieties, and Algorithms" にでている, 図のような二つの関節をもつロボットの腕の問題を考える。



問題はロボットの腕の先端の座標 (a,b) を固定したとき, 何通りの関節の曲げ方があるのか列挙することである. $c_i = \cos \theta_i, s_i = \sin \theta_i$ (プログラムでは $c1, c2, s1, s2$) とおいて制約条件を代数方程式に書くと, 次の Risa/Asir のプログラムの代数方程式 F の実数解を列挙することとなる。

```
def foo0(A,B) {
  F = [13*(c1*c2-s1*s2)+12*c1 - a,
       13*(s1*c2+s2*c1)+12*s1 - b,
       c1^2+s1^2-1,
       c2^2+s2^2-1];
  F = base_replace(F, [[12,1], [13,2], [a,A], [b,B]]);
  V = [c2,s2,c1,s1];
  GO = hgr(F,V,2);
  return GO;
/* GO = hgr(F,V,0); U = minipoly(GO,V,0,s1,s1); */
}
```

実行すると次のようになる。

```
G2=foo0(11/10,21/10);
[-56200*s1^2+55020*s1-5061,-110*c1-210*s1+131,
 2200*s2+5620*s1-2751,-200*c2+31]
pari(roots,G2[0]);
[ 0.1027736950248033986 0.8762298636940578112 ]
```

このように実数解は二つあることがわかる。

この方法の D -Analogy はありますか?

有理関数係数の微分作用素環を考える. $R = R_n = \mathbb{C}(x_1, \dots, x_n) \langle \partial_1, \dots, \partial_n \rangle$.
この環ではかけ算はたとえば次のようになる. $\partial_1 \left(\frac{x_1}{1-x_2} \right) \partial_1 = \left(\frac{x_1}{1-x_2} \right) \partial_1^2 + \frac{1}{1-x_2} \partial_1$.

I を D のイデアルとするととき $J = RI$ を考える.

定理. (WN) J が R_n の 0-次元イデアルであるとき, $J \cap \mathbb{C} \langle x, \partial_i \rangle$ は, 0 でない元を含む (つまり x_i についての常微分方程式を含む).

この元はたとえば weight vector $(0, \dots, 0; 1, 1, \dots, 1, 0, 1, \dots, 1)$ でグレブナ基底を計算することにより求めることが可能である.

さて, $\text{in}_{(0,1)}(I)$ の次元が n である D/I を holonomic という. D/I が holonomic なら $J = RI$ は R で zero-dimensional である. 超幾何方程式系は holonomic である. 結論として, 超幾何方程式系はかならず常微分方程式系を含み, それを構成することがグレブナ基底の方法を用いることにより可能である. したがって代数方程式の場合と同様に, 常微分方程式を数値的に解きそれからその解を別の方向へ延ばしていくという手法で数値解を構成することが可能である.

計算例は?

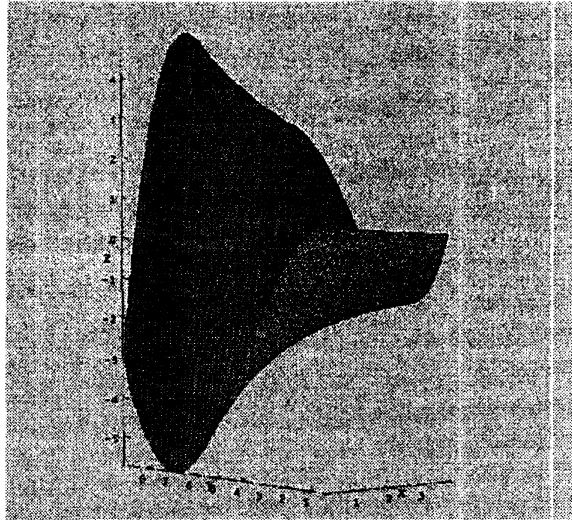
例. Okamoto-Kimura が Garnier 系の研究から再発見したいわゆる Bessel function in two variables のグラフを描画してみよう. この関数は次の積分表示で定義される.

$$f(a; x, y) = \int_C \exp\left(-\frac{1}{4}t^2 - xt - y/t\right)t^{-a-1} dt$$

where $C = 0\vec{1} + \{e^{2\pi\sqrt{-1}\theta} \mid \theta \in [0, 2\pi]\} + 1\vec{0}$. 関数 $f(a; x, y)$ は次の holonomic システムを満たす.

$$\partial_x \partial_y + 1, \partial_x^2 - 2x \partial_x + 2y \partial_y + 2a, 2y \partial_y^2 + 2(a+1) \partial_y - \partial_x + 2x$$

解空間の次元は 3 である. $a = 1/2$ とおこう. このとき上の方程式は $y^{-a}g(x, y)$ なる形の解をもつここで g は原点で正則な関数であり $g(0, 0) = 1$ である. 関数 g のグラフはつぎのようになる. いろんな方向に振動していて 2 変数ベッセル関



数とよんでいい気分がしてくる。

Risa/Asir のプログラムで y 軸方向の常微分方程式を求めるプログラムはつぎのようになる。

```
def bess2_ode_y(A) {
  F = [dx*dy+1 , dx^2-2*x*dx+2*y*dy+2*a , 2*y*dy^2+2*(a+1)*dy-dx+2*x];
  F = base_replace(F, [[a,A]]);
  G = sm1.gb([F, [x,y], [[dx,1]]]);
  return G[0];
}
```

```
bess2_ode_y(1/2);
```

```
[-dx+2*y*dy^2+3*dy+2*x, -2*y*dy^3-5*dy^2-2*x*dy-1]
```

このように 3 階の常微分方程式が含まれている。この方程式から $f = y^{-a}g$ の満たす方程式を求めると次のようになる。

```
4*y^2*dy^3+4*y*dy^2+(4*y*x-1)*dy-2*x+2*y, ...
```

$y = 0$ に特異点をもつので原点の周りでは級数解を求める必要がある。 $g(0,0) = 1$ となるような級数解は次ぎのようになる。

```
(1)+(-1/3*y^2)+(-2*y*x)+(1/210*y^4)+(2/15*y^3*x)+(2/3*y^2*x^2)+ ...
```

級数解の収束が悪くなるあたりから常微分方程式を adaptive Runge-Kutta 法で解いて書いたグラフが上のグラフである。

これで一件落着か?

実はそうではない。 $I \cap \mathbb{Q}[y_1]$ を求めるといった消去法は時間がかかる計算でありなるべく避けたい。おなじような問題は多項式環でも発生している。消去法 (lex order のグレブナ基底など) を用いない方法をまず多項式環で復習しよう。

定理. (WN) $y^{\alpha(1)}, \dots, y^{\alpha(r)}$ を $\mathbb{Q}[y_1, \dots, y_n]/I$ の \mathbb{Q} 上のベクトル空間としての基底とする。次の条件をみたす行列 $M_1, \dots, M_n \in M(r, r, \mathbb{Q})$ が存在してかつ構成できる。構成にはグレブナ基底による normal form algorithm

を用いる.

$$\begin{array}{ccc}
 y_1 \begin{pmatrix} y^{\alpha(1)} \\ \vdots \\ y^{\alpha(r)} \end{pmatrix} & = & M_1 \begin{pmatrix} y^{\alpha(1)} \\ \vdots \\ y^{\alpha(r)} \end{pmatrix} \pmod{I} \\
 \dots & & \dots \\
 \dots & & \dots \\
 \dots & & \dots \\
 y_n \begin{pmatrix} y^{\alpha(1)} \\ \vdots \\ y^{\alpha(r)} \end{pmatrix} & = & M_n \begin{pmatrix} y^{\alpha(1)} \\ \vdots \\ y^{\alpha(r)} \end{pmatrix} \pmod{I}
 \end{array}$$

この定理の D -analogy はつぎのようになる.

定理. $\partial^{\alpha(1)}, \dots, \partial^{\alpha(r)}$ を $\mathbf{Q}(x_1, \dots, x_n)$ のベクトル空間 $\mathbf{Q}(x_1, \dots, x_n)(\partial_1, \dots, \partial_n)/I$ の基底とする. このとき次の条件をみたす行列 $P_1, \dots, P_n \in M(r, r, \mathbf{Q}(x))$ が存在してかつ構成できる. 構成には R でのグレブナ基底による normal form algorithm を使う.

$$\begin{array}{ccc}
 \partial_1 \begin{pmatrix} \partial^{\alpha(1)} \\ \vdots \\ \partial^{\alpha(r)} \end{pmatrix} & = & M_1 \begin{pmatrix} \partial^{\alpha(1)} \\ \vdots \\ \partial^{\alpha(r)} \end{pmatrix} \pmod{I} \\
 \dots & & \dots \\
 \dots & & \dots \\
 \dots & & \dots \\
 \partial_n \begin{pmatrix} \partial^{\alpha(1)} \\ \vdots \\ \partial^{\alpha(r)} \end{pmatrix} & = & M_n \begin{pmatrix} \partial^{\alpha(1)} \\ \vdots \\ \partial^{\alpha(r)} \end{pmatrix} \pmod{I}
 \end{array}$$

さて f を解として, $F = (\partial^{\alpha(1)} \bullet f, \dots, \partial^{\alpha(r)} \bullet f)^T$ とおこう. すると上の行列からつくった次の連立常微分方程式をとけばその解 F の第一成分をとることにより f が求まることとなる.

$$\partial_1 \bullet F = M_1 F, \dots, \partial_n \bullet F = M_n F$$

大事な点はこの場合消去をやるような順序を選ぶ必要はなく, どんな順序を選んでも連立常微分方程式を作れるという点である. したがって計算に有利な順序を用いればよい. 差分化の場合には, 方程式に応じた差分の仕方により数値計算の様相がかわる. Holonomic 系の場合には, 差分化のみならず順序の選び方による連立系への変形方法で数値計算の様相が変わることとなる.

このあたりの詳しい事情はまだ深く調べられていない。順序の選び方以外にも基底の選び方でも連立常微分方程式化の様子は変わる。たとえば $\partial^{\alpha(i)}$ の代わりにオイラー作用素 $\theta^{\alpha(i)}$, $\theta_i = x_i \partial_i$ を基底に選ぶと連立常微分方程式は簡単になることが多い。

例. (Bessel differential equation in two variables, $a = 1/2$)
 $F = (f, \partial_y \cdot f, \partial_x \cdot f)^T$ のとき, P_1, P_2 は次のようになる。

$$P_1 = \begin{pmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ -1 & -2y & 2x \end{pmatrix},$$

$$P_2 = \begin{pmatrix} 0 & 1 & 0 \\ \frac{-x}{y} & \frac{-3}{2y} & \frac{1}{2y} \\ -1 & 0 & 0 \end{pmatrix}$$

この形の連立方程式で常微分方程式を次の x, y の範囲 $x \in [0.0, 1.4]$ (step size : 0.1), $y \in [0.2, 9.0]$ で解くと, Risa/Asir を Pentium III Xeon/Celeron (1129.43-MHz 686-class CPU), memory 2Gb PC/AT 互換機で動かすことにより 2.037sec + gc : 0.4651sec 程度の時間で数値解を求めることが可能である。

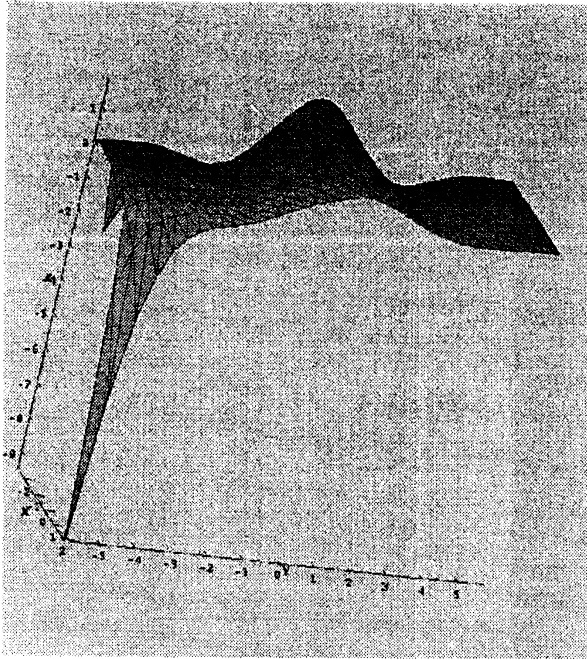
例. (Airy function in two variables, Okamoto-Kimura, Miyamoto)

$$\begin{aligned} (-\partial_x^2 + (x/2)\partial_x + (y/4)\partial_y + 1/4) \cdot f &= 0 \\ (-\partial_x \partial_y + (x/2)\partial_y + (y/2)) \cdot f &= 0 \\ (-\partial_y^2 + 2\partial_x) \cdot f &= 0 \end{aligned}$$

Miyamoto により次のような級数解の大域的な性質が調べられた。

$$z_1(x, y) = \sum \frac{x^j y^k}{(1)_j (1)_k} \left((-1)^j \sqrt{-1}^{k+1} - 1 \right) 2^{k-3/2} \Gamma \left(\frac{2j+k+1}{4} \right)$$

この解の虚数部分 $\text{Im } z_1(x, y)$ のグラフを我々の方法で書くと次のようになる。



微分作用素環を扱うのにどんなシステムがありますか?

代数幾何関連の D 加群のアルゴリズムについては,
 kan/sml (<http://www.math.kobe-u.ac.jp/KAN>),
 Macaulay2 (<http://www.math.uiuc.edu/Macaulay2>).

その他

Ore algebra packages (F.Chyzak) in Maple (一般的).
 bfct package (M.Noro) in Risa/Asir (b-関数の計算が早い).
 yang package (K.Ohara) in Risa/Asir (一般的, flexible).
 Plural (一般的, Singular の資産の継承).

どうして数値計算に級数解を構成するのが必要なのでしょうか?

さて 2 変数 Bessel 関数の解析のところでも述べたように我々の方法では級数解の構成も大事な要素となる. どうして級数解の構成が数値計算に大事かという次の超幾何関数の numerical recipes にあるように高速高精度の数値計算には不可欠であるからである.

Numerical Recipes for Hypergeometric Functions

- 級数解は収束領域の十分内部での数値計算のためにはベストの方法であることが多い. とくに高い桁まで求めようとするときは不可欠の方法である. また差分法が使えない特異点のまわりでも必須である.
- Adaptive Runge-Kutta 法などの常微分方程式の数値解法は超幾何関数の数値計算をするためにまあまあ悪くない方法である.
- 超幾何関数の Barnes 積分表示を用いて, 数値積分するのがいいときもある.

また数値計算したい解の条件が漸近的性質で与えられることも多いのでそのような場合は特異点の近くでの級数解が必須となる。

さて上に述べたような numerical recipes の証拠となるようないくつかの計算例を紹介する。これらは田村恭士のデジタル数学公式集についての博士論文 (2003, 神戸大学) でも例として使われている

(<http://www.symbolicnet.org/conferences/iamc03/program.html> も参照)。

超幾何関数の積分表示で数値計算を全部すませられないのですか？

例. 次の公式の左辺を数値的に評価して右辺の数値と近似的に同じであることを確かめよ。

$$F\left(\frac{1}{12}, \frac{5}{12}, \frac{1}{2}; \frac{1323}{1331}\right) = \frac{3}{4}\sqrt{11} \quad (\text{Beukers, 1993})$$

ここで

$$F(a, b, c; z) = \frac{\Gamma(c)}{\Gamma(b)\Gamma(c-b)} \int_0^1 t^{b-1}(1-t)^{c-b-1}(1-tz)^{-a} dt$$

●この数値計算を上積分表示を用いて数値積分で計算してみる。そのまま数値積分すると収束がきわめて悪いので、次のように contiguity relation を用いてパラメータをずらしてから計算する。

$$\begin{aligned} & F\left(\frac{1}{12}, \frac{5}{12}, \frac{1}{2}; \frac{1323}{1331}\right) \\ &= -\frac{555146934690291893170809321}{77265229938688} F\left(-\frac{31}{12}, \frac{37}{12}, \frac{13}{2}; \frac{1323}{1331}\right) \\ &+ \frac{23008497055530190854682531919}{4017791956811776} F\left(-\frac{31}{12}, \frac{37}{12}, \frac{15}{2}; \frac{1323}{1331}\right) \end{aligned}$$

10^{-4} まで正しい数値を得るのに約 9 秒かかった。なおこの数字は Pentium III 552 MHz, Memory 512 Mb のマシンで Methematca 4.0 の関数 NIntegrate を用いた結果である。ちなみに Risa/Asir 上で書いた自作数値積分関数では、いろいろなアルゴリズムをためしたもののこれより早いデータはだせなかった。

●常微分方程式を数値計算する:

関数 $F(a, b, c; z)$ は次の常微分方程式を満たす。

$$z(1-z)f'' + (c - (a+b+1)z)f' - abf = 0$$

$$f(0) = 1.$$

この場合は 2 秒で 10^{-4} まで正しい結果を得ることができた。ここで用いた方法は, adaptive Ruge-Kutta method (4th order) であり, それを Risa/Asir に実装して上記と同じマシンで実行したものである。

- 微分方程式の級数解を数値的に評価:

$$F(a, b, c; x) = \sum_{n=1}^{\infty} \frac{(a)_n (b)_n}{(1)_n (c)_n} x^n,$$

$$(a)_n = a(a+1) \cdots (a+n-1).$$

10^{-4} の精度で答えを求めるのに 1 秒以下である.

級数解の構成方法にはどのようなものがありますか?

1 変数の場合は, Frobenius の方法とか Hukuhara-Turritin 簡約とか教科書にでて (cf. Maple の DEtools). 多変数に関しては次のような結果がある.

定理. (Hosono, Lian, S.T. Yau, 1996, 97) reflexive polytopes に対応する GKZ 超幾何方程式系の級数解がアルゴリズム的に構成できる (この構成方法のアイデアは $\text{in}_{(-w,w)}(I_A)$ の活用である).

定理. (Saito, Sturmfels, Takayama, 1998. cf. vanishing cycle sheaf: Kashiwara, Laurent). D/I が regular holonomic であるとする. I の解空間の次元 (rank) は $\text{in}_{(-w,w)}(I)$ の rank に等しい.

定理. (SST, 2000). D/I が regular holonomic であるとする. このときこの系の small Gröbner fan に対応する $(\mathbb{C}^*)^n$ の toric コンパクト化の上で, I の級数解の基底を構成するアルゴリズムがある. この構成では $\text{in}_{(-w,w)}(I)$ の解を延ばしていく方法をとる.

- (Open Question) holonomic system の irregular singular point の周りで級数解を Villamayor の特異点解消のアルゴリズムを利用して求めよ. (cf. Majima, Bodnar, Schicho の仕事も参照).

級数解の数値計算について最近どのような仕事がされていますか?

J. Van der Hoeven : 常微分方程式の解の数値計算 (Binary splitting algorithm と数値的な解析接続による高速高精度数値計算). 1 変数 holonomic 関数の数値計算 (Journal of Symbolic Computation (2001), 717-743).

問題: D 加群のアルゴリズムと, 我々の級数解の構成アルゴリズムを用いると, Van der Hoeven のアルゴリズムを我々の計算に適用可能か?

さて我々が数値計算の研究を始めた動機の一つはデジタル数学公式集作成のためのオーサリングツールの開発であった. これについては先程紹介した田村の博士論文が詳しい. この論文ではさらに 1 変数超幾何関数の数値計算を考察している. ここで話題を変えてこのデジタル数学公式集についてすこし述べよう.

計算機で使える数学公式集にはどのようなものがありますか?

(1) <http://functions.wolfram.com>

(2) 1980 年代の佐々木 (筑波大) らの研究 (岩波公式集ベース). (cf. 森永, 村

上 (愛媛大) 2003)

(3) ...

MathML や OpenMath は数学表現の標準化プロジェクトと聞きました
これらのプロジェクトとデジタル数学公式集のプロジェクトは多いに関係
がある。 OpenMath, <http://www.openmath.org/> は 1997 年頃, ヨーロッ
パで Arjeh Cohen などを中心として活動開始した。 数学的意味論が欠落しな
いように注意が払われている。 たとえば公式を TeX で表現すると, 数学的意
味論が欠落してしまう。 MathML は数式を記述するためのマークアップ言
語であり, XML と同じく W3C (<http://w3c.org>) から勧告されている。 $1+x$
は Presentation MathML では

```
<mrow>
  <mn>1</mn>
  <mo>+</mo>
  <mi>x</mi>
</mrow>
```

と表現する。

OpenMath の特徴は CD(Content Dictionary) を用いて, 容易に拡張でき
ることである。

これらの新しい技術をもとにした公式集プロジェクトはありますか?

はい。それが OpenXM/fb プロジェクトです (田村 D 論 (2003), 他)。

New Projects: Digital formula book (OpenXM/fb).

<http://www.openxm.org>

このプロジェクトでは, OpenMath XML を基礎として, 超幾何関数の公式の
蓄積を行っている。 詳しいことはさておいてわれわれの公式集のソースコー
ドを見てみよう。

```
<?xml version="1.0" encoding="ISO-2022-JP"?>
<formula>
  <title>
    Quadratic transformation of an independent variable
  </title>
  <author> E. Goursat </author>
  <editor> Yasushi Tamura </editor>

  <tfb macroset="http://www.openxm.org/fb/hfb.txt">
    2 * arith1.root(nums1.pi,2)
    * hypergeo0.gamma(a + b + (1 / 2))
    / hypergeo0.gamma(a + (1 / 2))
```

```

/ hypergeo0.gamma(b + (1 / 2))
* hypergeo1.hypergeometric2F1(a,b,1 / 2,x)
~relation1.eq~
(hypergeo1.hypergeometric2F1(2 * a, 2 * b, a + b + (1 / 2),
1 + arith1.root(x,2) / 2)
+ hypergeo1.hypergeometric2F1(2 * a, 2 * b, a + b + (1 / 2),
1 ~arith1.minus~ arith1.root(x,2) / 2));
</tfb>

<description>
  Quadratic transformation of independent variable
</description>
<reference>
  <xref uri="http://www.openxm.org/fb/bib-goursat1.xml"
  linkend="goursat1" page="118"/>
</reference>
<evidence checker="Mathematica">
  @@ /. {a->1/2,b->3/5,c->-2/11,x->0.2}
</evidence>
</formula>

```

このコードは次の公式を表現している。

$$\frac{2\sqrt{\pi}\Gamma(a+b+\frac{1}{2}){}_2F_1(a,b,\frac{1}{2},x)}{\Gamma(a+\frac{1}{2})\Gamma(b+\frac{1}{2})} = {}_2F_1(2a,2b,a+b+\frac{1}{2},\frac{1-\sqrt{x}}{2}) + {}_2F_1(2a,2b,a+b+\frac{1}{2},\frac{1+\sqrt{x}}{2})$$

なおこの公式は 100 年以上前に Goursat が発見した公式である。

われわれの定義したエレメントは DocBook の拡張であり、拡張部分の根となるエレメントは <formula> である。<tfb> エレメントに公式の本体が tfb 形式ではいる。

公式集の編集にあたり公式の背後に広がる豊かな数学的世界へのリンクを挿入しておきたいという気持があるが、いまのところタグ <reference> 位しかアイデアがない。これは今後の問題であろう。

公式集の編集での微分作用素環のアルゴリズムの役割は?

(1) 公式の生成; Contiguity relation (Takayama, 1989)

$$\text{例} \quad (x\partial_x + a)F(a, b, c; x) = aF(a + 1, b, c; x),$$

Kummer 型の等式 (Takayama, 2003), Connection formula (退化した場合, Tamura, 2003).

(2) 公式の検証; Zeilberger's algorithm (本 $A = B$ を参照, 1990's).

(3) 数値計算による明らかな入力ミス等の検出 (Tamura, 2003).

など超幾何の公式集ではいろいろな場面で活躍している。