

2元分割表に対する差分ホロノミック勾配法の実装

神戸大学大学院理学研究科 後藤良彰

Yoshiaki Goto

Department of Mathematics, Graduate School of Science, Kobe University

神戸大学大学院理学研究科 橋義仁

Yoshihito Tachibana

Department of Mathematics, Graduate School of Science, Kobe University

神戸大学大学院理学研究科 高山信毅

Nobuki Takayama

Department of Mathematics, Graduate School of Science, Kobe University

ホロノミック勾配法は [4] により提案された確率分布の正規化定数とその微分をホロノミックな微分方程式, 差分方程式を活用して高速に計算する手法である. 本稿では [1] が与えた (k, n) 型多変数超幾何関数の満たす差分方程式 (漸化式, contiguity relation) およびモジュラーメソッド (modular method) を活用して 2 元分割表の条件付き多項分布の正規化定数およびその微分を高速に有理数で計算するアルゴリズム, その計算量, およびその実装実験を報告する.

1 2元分割表

この章では $r_1 \times r_2$ 分割表に関して本稿で必要な事項を紹介する. 分割表については様々な文献があるが, ここでは [2, 4 章] を引用し解説する.

1.1 $r_1 \times r_2$ 分割表

定義 1 ($r_1 \times r_2$ 2元分割表) $r_1 \times r_2$ 2元分割表とは, 非負整数を成分とする $r_1 \times r_2$ 行列である. 分割表 $u = (u_{ij})$ に対して, 行和ベクトルを $\beta^r = \left(\sum_j u_{1j}, \dots, \sum_j u_{r_1 j} \right)^T$, 列和ベクトルを $\beta^c = \left(\sum_i u_{i1}, \dots, \sum_i u_{i r_2} \right)^T$ と定める. 分割表 u を長さ $r_1 \times r_2$ の縦ベクトルに書き直したものを u^f と書く. β^r, β^c をつないだ縦ベクトルを β として, 行和列和ベクトルまたは周辺和ベクトルと呼ぶ.

例 1 (2×3 分割表と行和, 列和) 2×3 分割表 $u = \begin{pmatrix} 5 & 3 & 6 \\ 7 & 2 & 4 \end{pmatrix}$ の行和ベクトルと列和ベクトルはそれぞれ以下ようになる:

$$\beta^r = \begin{pmatrix} 5+3+6=14 \\ 7+2+4=13 \end{pmatrix}, \beta^c = \begin{pmatrix} 5+7=12 \\ 3+2=5 \\ 6+4=10 \end{pmatrix}.$$

これらを u^f, β の形に書き直すと以下ようになる:

$$u^f = (5 \ 3 \ 6 \ 7 \ 2 \ 4)^T, \beta = (14 \ 13 \ 12 \ 5 \ 10)^T.$$

$p = (p_{ij}) \in \mathbb{R}_{>0}^{r_1 \times r_2}$, $N \in \mathbb{N}_0$ を固定し, $|u| = \sum_{i,j} u_{ij} = N$ を満たす分割表 u に対して多項分布

$$\frac{N! p^u}{u! |p|^N}, \quad p^u = \prod_{i,j} p_{ij}^{u_{ij}}, \quad u! = \prod_{i,j} u_{ij}!$$

を考える. ここで行和と列和をそれぞれ β^r, β^c に固定した条件付き分布を考えると, 分割表 u を得る確率は

$$\frac{p^u}{u! Z(\beta; p)}, \quad Z(\beta; p) = \sum_{Au^f = \beta, u^f \in \mathbb{N}_0^{r_1 \times r_2}} \frac{p^u}{u!}$$

となる. ただし, $Z(\beta; p)$ は条件付き確率の正規化定数であり, A は Au^f が u の周辺和ベクトルとなるような $0, 1$ を成分とする行列である (下の例 2 も参照). 分割表 u を得る条件付き確率を計算する問題は正規化定数の値の計算に帰着される. この正規化定数 Z の有理数 p_{ij} 達に対する効率的な計算が我々の目標である.

例 2 (A の例) $u^f = \begin{pmatrix} 5 & 3 & 6 & 7 & 2 & 4 \end{pmatrix}^T$ の場合, A は

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

である. 実際に計算すると

$$Au^f = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 5 \\ 3 \\ 6 \\ 7 \\ 2 \\ 4 \end{pmatrix} = \begin{pmatrix} 14 \\ 13 \\ 12 \\ 5 \\ 10 \end{pmatrix} = \beta$$

となることが分かる.

例 3 2×2 分割表で行和, 列和を $\beta = \begin{pmatrix} 5 & 7 & 8 & 4 \end{pmatrix}^T$ として $Au^f = \beta$ となる分割表 u を全て並べると,

$$\begin{pmatrix} 5 & 0 \\ 3 & 4 \end{pmatrix}, \begin{pmatrix} 4 & 1 \\ 4 & 3 \end{pmatrix}, \begin{pmatrix} 3 & 2 \\ 5 & 2 \end{pmatrix}, \begin{pmatrix} 2 & 3 \\ 6 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 4 \\ 7 & 0 \end{pmatrix}$$

となる. これは

$$\begin{pmatrix} 5 & 0 \\ 3 & 4 \end{pmatrix} + i \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}, \quad (i = 0, 1, 2, 3, 4)$$

と書ける.

2 2×2 分割表の正規化定数

$r_1 \times r_2$ 分割表の正規化定数の計算は多変数超幾何級数の値を計算することに帰着されることが知られている (たとえば [2, p.399, 6.13]). この節では, 2×2 分割表の場合に超幾何級数との対応を与えておく.

2×2 分割表の周辺和として $\beta = (u_{11}, u_{21} + u_{22}, u_{11} + u_{21}, u_{22})$ となるものを考える. 周辺和が β となる分割表は次の通り:

$$u = \begin{pmatrix} u_{11} & 0 \\ u_{21} & u_{22} \end{pmatrix} + i \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}, \quad (i = 0, 1, 2, \dots, n)$$

ここで, $n = \min\{u_{11}, u_{22}\}$. このとき正規化定数は

$$\begin{aligned} Z(\beta; p) &= \sum_{i=0}^n \frac{p_{11}^{u_{11}-i} p_{12}^{u_{21}+i} p_{22}^{u_{22}-i}}{(u_{11}-i)!(i)!(u_{21}+i)!(u_{22}-i)!} \\ &= \frac{p_{11}^{u_{11}} p_{21}^{u_{21}} p_{22}^{u_{22}}}{u_{11}! u_{21}! u_{22}!} \sum_{i=0}^n \frac{(-u_{11})_i (-u_{22})_i}{(u_{21}+1)_i (1)_i} \left(\frac{p_{12} p_{21}}{p_{11} p_{22}} \right)^i \end{aligned}$$

となる. よって対応する超幾何級数は Gauss の超幾何級数

$${}_2F_1(a, b, c; x) = \sum_{i=0}^{\infty} \frac{(a)_i (b)_i}{(c)_i (1)_i} x^i$$

であることが分かる ($a, b \in \mathbb{Z}_{\leq 0}$ のときは多項式となる).

3 Contiguity relation

前節では 2×2 分割表の場合について述べたが, 一般に $r_1 \times r_2$ 分割表の正規化定数は, 青本-Gelfand の $(r_1, r_1 + r_2)$ 型超幾何関数が対応している (Gauss の ${}_2F_1$ は $(2, 4)$ 型). 以下, 超幾何級数の値の計算を差分ホロノミック勾配法を用いて行うが, そのためには contiguity relations (隣接関係式) と呼ばれる関係式が必要になる.

例 4 (${}_2F_1$ の場合) $f(a) = {}_2F_1(a, b, c; x)$ とおく.

$$F(a) = \begin{pmatrix} f(a) \\ \theta_x f(a) \end{pmatrix}, \quad M(a) = \frac{1}{a-c+1} \begin{pmatrix} bx+a-c+1 & x-1 \\ -abx & a(1-x) \end{pmatrix}$$

は $F(a) = M(a)F(a+1)$ を満たす. θ_x は Euler 作用素 $x\partial_x$ である.

超幾何関数に関しては上の例の様な漸化式を contiguity relation と呼ぶ. [1] では, $F(a)$ を Gauss-Manin ベクトルと呼んでいる.

3.1 Contiguity relation を求めるための種々のアルゴリズム

Contiguity relation を求めるアルゴリズムについては次のようにさまざまな研究がある.

1. 明らかな contiguity relation (たとえば $\frac{1}{a}(x\partial_x + a): {}_2F_1(a, b, c; x)$ の a を 1 増やす作用素) の “逆” をグレブナー基底を用いて求める方法 [13]. たとえば Risa/Asir [11] のパッケージである `yang.rr` (小原) を用いれば容易に実装可能で小規模な問題を解くには最良の手法であろう. しかし, 問題のサイズが大きくなると巨大なグレブナー基底で計算が困難になる.
2. Euler 変換による方法 [9]. 実装としては Risa/Asir [11] のパッケージである `os_muldif.rr` (大島) の関数 `shiftop`. 正規化定数 Z が適当な p の方向に関して満たす常微分方程式のリーマン図式が一般には不明なため, 未評価.
3. [8] による Macaulay 型行列を用いた方法. この手法は分割表を含む一般の A -超幾何分布の正規化定数の計算に適用可能である. Z とその偏微分からなる Gauss-Manin ベクトル $F(\beta)$ の値を求める計算量は

$$O(|\beta|(m-1)^{r_1 \times r_2} r_1^2 r_2^2 m^{3d})$$

ここで $d = r_1 + r_2 - 1$, m は Macaulay 型行列を作る時に使うモノミアルの次数で, $m \geq 2$. m の上限は未解決.

4. (k, n) 型超幾何関数に対するねじれコホモロジー群と交点形式による方法 [1].

我々は超幾何関数の積分表示から定まるねじれコホモロジー群を用いて contiguity relations を導出する方法 [1] の計算量評価および実装を行った. もっとも高速な計算量をもつものと期待されたからである.

[1] の方法の概要を述べる. $(r_1, r_1 + r_2)$ 型超幾何関数 (級数) $f(\alpha; x)$ を考える ($\alpha = (\alpha_1, \dots, \alpha_{r_1+r_2-1})$ がパラメータ (β と対応), $x = (x_{ij})_{1 \leq i \leq r_1-1, 1 \leq j \leq r_2-1}$ が変数 (p と対応)). f の積分表示から定まるねじれコホモロジー群 (微分形式で代表されるベクトル空間) を考察することにより, $\alpha_i \rightarrow \alpha_i + 1$ の変化を例 4 のように記述できる:

- ねじれコホモロジー群の間のある線形写像 U_i について, 適当な基底に関する表現行列を U_i とする:

$$(\mathcal{U}_i(\varphi'_1), \dots, \mathcal{U}_i(\varphi'_r))^T = U_i \cdot (\varphi_1, \dots, \varphi_r)^T.$$

- 両辺を “積分” して級数展開することで, contiguity relation

$$F(\alpha; x)|_{\alpha_i \rightarrow \alpha_i+1} = \tilde{U}_i(\alpha; x)F(\alpha; x), \quad F = (f, \delta^{(2)} \bullet f, \dots, \delta^{(r)} \bullet f)^T$$

を得る. $\delta^{(i)}$ は適当な微分作用素, $\tilde{U}_i$ は U_i の定数倍である. Gauss-Manin ベクトル F の長さ r は, ねじれコホモロジー群の次元で $r = \binom{r_1+r_2-2}{r_1-1}$ となることが分かる.

この表現行列 U_i は, 簡単な対角行列と基底変換行列で表示できることがわかる. この基底変換行列も交点数を使って容易に計算できる. また, $\alpha_i \rightarrow \alpha_i - 1$ についても同様に導出できる.

例 5 (${}_2F_1$ の場合 ($r_1 = r_2 = 2, r = 2$)) ${}_2F_1$ のパラメータ (a, b, c) に対して

$$(\alpha_1, \alpha_2, \alpha_3) = (b, -a, c - b - 1)$$

とおく. ここで, 便宜上 $\alpha_0 = -\alpha_1 - \alpha_2 - \alpha_3 = a - c + 1$ とする. このとき, $a + 1 \rightarrow a$ は $\alpha_2 - 1 \rightarrow \alpha_2$ ($\alpha_0 + 1 \rightarrow \alpha_0$) に対応するので, 例 4 の $M(a)$ は $U_2(\alpha; x)$ と対応する. 表現行列 U_2 は以下の表示をもつ:^{*1}

$$U_2 = \frac{\alpha_1^2 \alpha_2 (\alpha_0 + 1) (\alpha_2 - 1)}{\alpha_3} \cdot \begin{pmatrix} \frac{1}{\alpha_0} + \frac{1}{\alpha_1} & \frac{1}{\alpha_0} \\ \frac{1}{\alpha_0} & \frac{1}{\alpha_0} + \frac{1}{\alpha_2} \end{pmatrix} \begin{pmatrix} -\frac{1}{\alpha_2} & \frac{1}{\alpha_1} \\ 0 & \frac{1}{\alpha_1} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1-x \end{pmatrix} \begin{pmatrix} \frac{1}{\alpha_0+1} + \frac{1}{\alpha_1} & \frac{1}{\alpha_0+1} \\ \frac{1}{\alpha_0+1} & \frac{1}{\alpha_0+1} \end{pmatrix} \begin{pmatrix} \frac{1}{\alpha_0+1} + \frac{1}{\alpha_2-1} & -\frac{1}{\alpha_0+1} \\ -\frac{1}{\alpha_0+1} & \frac{1}{\alpha_0+1} + \frac{1}{\alpha_1} \end{pmatrix}.$$

交点数の行列は $\text{diag}(1, 1-x)$ 以外の部分である. 実は $\delta^{(2)} = \frac{1}{\alpha_2} \theta_x$ となっているので, 例 4 とは若干のずれが生じるが, それを修正すると $M(a)$ が導出できる.

[1] の方法はたとえば r_1 を固定すると r_2 についての多項式時間アルゴリズムであり, 既知の方法に比べて高速である. 詳しい計算量は下記ようになる.

定理 1 $n \times n$ 行列のかけ算に要する計算量を $O(n^3)$ とするとき, [1] の方法で $r_1 \times r_2$ 分割表の正規化定数 Z の contiguity relation の行列を得るための計算量の上限は,

1. $r \geq r_i$ のとき, $O(r^3)$. ここで $r = \binom{r_1+r_2-1}{r_1-1}$.
2. r_1 を固定し, $r_2 \rightarrow \infty$ を考えるとき, $O(r_2^{3r_1})$.
3. r_2 を固定し, $r_1 \rightarrow \infty$ を考えるとき, $O(r_1^{3r_2})$.
4. $r_1 = r_2$ の条件下で $r_1 \rightarrow \infty$ を考えるとき, $O(2^{6r_1})$.

^{*1} Appendix で [1] および Risa/Asir [11] パッケージ gtt_ekn に即してこの行列の導出をもうすこし詳しく説明する.

注意 1 ねじれコホモロジー群や交点数の議論はパラメータ α (β たちと対応) が generic でないと使えない. 今回の応用については, 整数パラメータでの正当性に関しても保証されている [1, Corollary 7.5].

なおねじれコホモロジーの方法は強力であるが問題に応じた理論的な考察が必要であり, たとえば一般の (条件付き) A 分布 [14] に対するねじれコホモロジーの方法は知られていない.

3.2 ${}_2F_1$ の contiguity relation による差分ホロノミック勾配法

小さいサイズの問題に対して差分ホロノミック勾配法 (difference holonomic gradient method, 差分 HGM) を適用するには [13] の方法は有用である. 差分 HGM の紹介のためにこの方法を ${}_2F_1$ の場合に詳しく述べよう. $f(a) = {}_2F_1(a, b, c; x)$ とする. このとき次が成り立つ

$$\frac{1}{a}(a + \theta_x) \bullet f(a) = f(a + 1), \quad (1)$$

$$\{\theta_x(c - 1 + \theta_x) - x(a + \theta_x)(b + \theta_x)\} \bullet f(a) = 0. \quad (2)$$

$F(a) = (f(a), \theta_x f(a))^T$ とすると (1), (2) は次のように書ける.

$$\begin{aligned} \frac{1}{a}(a + \theta_x) \bullet F(a) &= F(a + 1), \\ \theta_x F(a) &= \begin{pmatrix} 0 & \frac{1}{\frac{ax+bx-c+1}{1-x}} \end{pmatrix} F(a) \\ &= A(a)F(a). \end{aligned}$$

これらより

$$\begin{aligned} \frac{1}{a}(a + \theta_x) \bullet F(a) &= \frac{1}{a}(aE + A(a))F(a) \\ F(a) &= \left(\frac{1}{a}(aE + A) \right)^{-1} F(a + 1) \\ &= M(a)F(a + 1) \end{aligned}$$

となる. ここで, E は単位行列である. この $M(a)$ を計算して求めると

$$M(a) = \frac{1}{a - c + 1} \begin{pmatrix} a + bx - c + 1 & x - 1 \\ -abx & a(1 - x) \end{pmatrix}$$

となる (これが例 4 の $M(a)$). このとき $a \in \mathbb{Z}_{<-1}$ なら

$$\begin{aligned} F(a) &= M(a)F(a + 1) \\ &= M(a)M(a + 1)F(a + 2) \\ &\vdots \\ &= M(a)M(a + 1) \cdots M(-2)F(-1). \end{aligned}$$

となり, $F(-1) = (1 - \frac{b}{c}x, -\frac{b}{c}x)^T$ の値から $F(a)$ の値が一次変換を繰り返すことにより求まる. この場合には 2×2 分割表の正規化定数とその微分の値が求まる. Contiguity relation は差分方程式と見なせるので, この方法を用いて Gauss-Manin ベクトル $F(a)$ の値を求めることを差分ホロノミック勾配法と呼んでいる.

4 Modular method

差分 HGM により有理数で正確に計算を行うには, 有理数ベクトルに対する一次変換を高速に行う必要がある. これについては LINBOX [3] などさまざまな実装研究がある. 我々は Risa/Asir [11] の分散計算機能を用いて実装研究を行った. またこの実装に即した計算量評価も行った.

4.1 計算量評価

計算の実行における計算量の評価と、具体的なデータを用いた実験結果を紹介する。今までの議論で、分割表の正規化定数の計算を対応する超幾何関数の値の計算に帰着し、その計算を差分ホロノミック勾配法を用いて行うことは述べた。差分ホロノミック勾配法を用いると実行する計算は行列とベクトルの積、つまりは一次変換の繰り返しである。この定義どおりの計算方法をアルゴリズム plain と呼ぶこととすると、その計算量評価は次のとおりである。

命題 1 行列のサイズを $r = \binom{r_1 + r_2 - 1}{r_1 - 1}$ 、一次変換の回数を n 、 m 桁同士の四則演算の計算量を $O(m^2)$ とする。入力は全て同じ桁数の整数であり、一次変換を一度行くと d 桁増えるものとする。これらの仮定の下でのアルゴリズム plain の計算量の上限は $O(r^2 d^2 n^3)$ 。

証明 一度の一次変換は r^2 回の積、 $r(r-1)$ 回の和からなる。初期値が x 桁の整数であったとするとアルゴリズム plain の計算量は

$$(2r^2 - r) \sum_{i=0}^n (x + di)^2 = \frac{2r^2 d^2 n^3}{6} + O(rn^2)$$

となる。 □

定義どおりに計算を実行すると一次変換の回数、つまり周辺和のサイズの 3 乗オーダーの計算量となるため周辺和の増加に対する計算時間の増加も激しくなる。また、有理数を用いて厳密な値を計算しようとする通分、約分の計算が発生するため計算時間がかかってしまう。これらの問題を解決するため次の二つの手法を用いた。

1. 分母・分子を個別に計算し最後に割り算を行うことで通分、約分にかかる時間を少なくする。
(`g_mat_fac_int`, 整数計算)
2. 複数の素数 P_i を用意し有限体 $\mathbb{F}_{P_i}$ 上で一次変換を行い、計算結果に対して中国剰余定理とアルゴリズム `IntegerToRational` ([7, アルゴリズム 6.25]) を適用する。各計算において必要に応じて分散計算を行う (`g_mat_fac_itor`, モジュラーメソッド)。

モジュラーメソッドはグレブナー基底の計算等に有効性があり [6][12]、今回は差分ホロノミック勾配法への適用を試みた。また、分散計算は `Risa/Asir` の `ox_rpc` 等の関数を用いて行った (詳しい説明は [5, 20 章])。それぞれのアルゴリズムは次のとおり。

アルゴリズム 1 (`g_mat_fac_int` (generalized matrix factorial by integers), 整数計算)

入力: $M(k)$: 変数 k を含む行列, F : ベクトル, S : 始点, E : 終点, Inc : 刻み幅

出力: $M(E) \cdots M(S + Inc)M(S)F$: 一次変換を繰り返した結果

1. $E = \frac{E-S}{Inc}$, $S = 0$, $Inc = 1$ と正規化する。合わせて M の変数変換も行う。 F , M の分母と分子を分け、それぞれを DNF , NMF , DNM , NMM とする (DNF , DNM は数字, NMF はベクトル, NMM は行列)。 $I = 0$ 。
2. $NMF = NMM(I) \times NMF$, $DNF = DNM(I) \times DNF$ 。
3. $I = I + 1$ 。 I が E 以下なら 2 へ。 そうでないなら $\frac{NMF}{DNF}$ を返す。

アルゴリズム 2 (`g_mat_fac_itor` (generalized matrix factorial by itor), モジュラーメソッド)

入力: $M(k)$, F , S , E , Inc , $PList$: 素数のリスト, $IDList$: 使用するプロセスのリスト

出力: $M(E) \cdots M(S + Inc)M(S)F$ の候補となる値

1. $E = \frac{E-S}{Inc}$, $S = 0$, $Inc = 1$ と正規化する. 合わせて M の変数変換も行う. F , M の分母と分子を分け, それぞれを DNF , NMF , DNM , NMM とする. $I = 0$.
2. $PList$ の各素数 P_i について, $\mathbb{F}_{P_i}$ 上で一次変換を分散計算を用いて行い, その計算結果を G^i とする.
3. 中国剰余定理を用いて, $G \equiv G^i \pmod{P_i}$ を満たすベクトル G を計算する. $P = \prod_{P_i \in PList} P_i$.
4. $\text{IntegerToRational}(G, P)$ によって得られる欲しい値の候補を返す.

整数計算の計算量の上限はアルゴリズム plain と同じだが, モジュラーメソッドの計算量の上限は次で与えられる.

定理 2 一次変換の回数を n , 行列のサイズを $r = \begin{pmatrix} r_1 + r_2 - 2 \\ r_1 - 1 \end{pmatrix}$, 全ての素数は d_p 桁で N_p 個使用し, プロセス数を C とする. m 桁同士の四則演算の計算量を $O(m^2)$ とすると, g_mat_fac_itor の計算量の上限は

$$\max \left\{ O \left(\frac{nr^2 N_p d_p^2}{C} \right), O \left(r(N_p d_p)^4 \right) \right\}$$

となる.

4.2 2×2 分割表の場合

アルゴリズム plain, g_mat_fac_int (整数計算のみ), g_mat_fac_itor (modular method) の実装を行い

$$f = {}_2F_1(-36N, -11N, 2N; \frac{1 - \frac{1}{N}}{56}), N \in \mathbb{N}$$

の値を求める計算を実行し, 計算時間の計測を行った. また, 計算に使用した環境は以下の通りである.

cpu	Intel(R) Xeon(R) CPU E5-4650 2.70GHz
cpu 数	32
コア数	8
os	debian 7.8
メモリ	256GB
使用言語	Risa/Asir version20150126

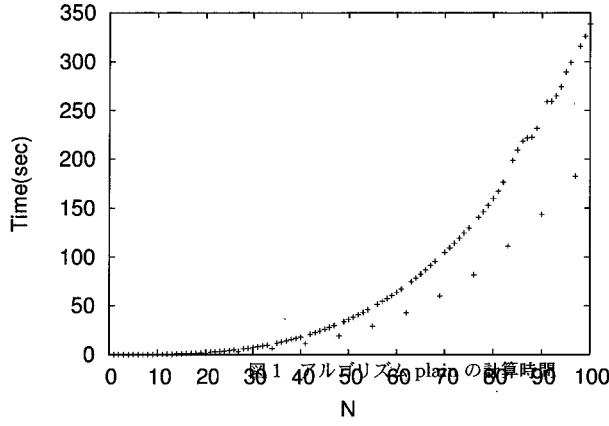


図1 アルゴリズム plain の計算時間

図1からも分かるように計算時間は N の2次以上の多項式の形になっており, $N = 100$ において6分程計算に要している.

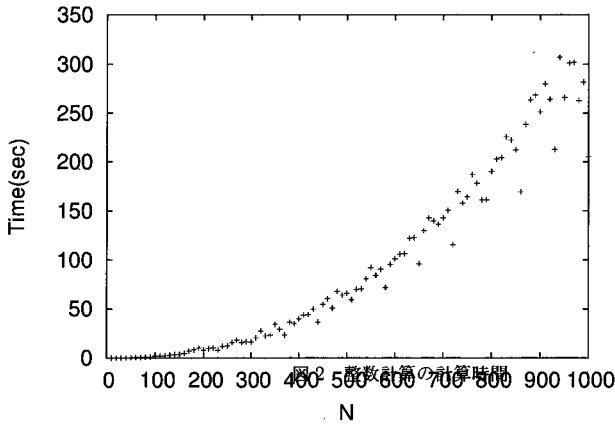


図2 整数計算の計算時間

整数計算での計算時間の増加はアルゴリズム plain と同様に N の2次以上の多項式の形になっている. しかし plain と比べると同じ N について計算時間の短縮が見られる分改良されたと言える (N のスケールが変わっていることに注意).

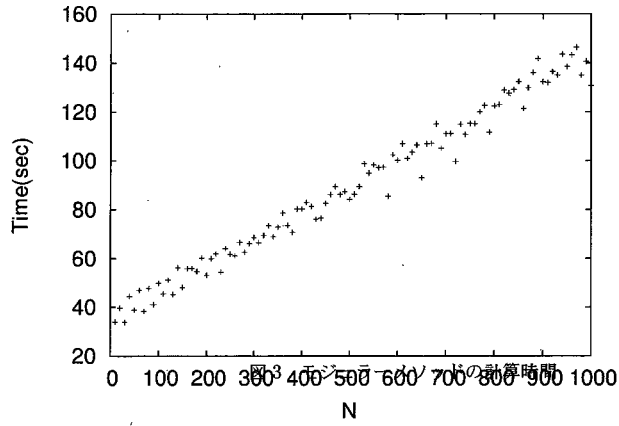
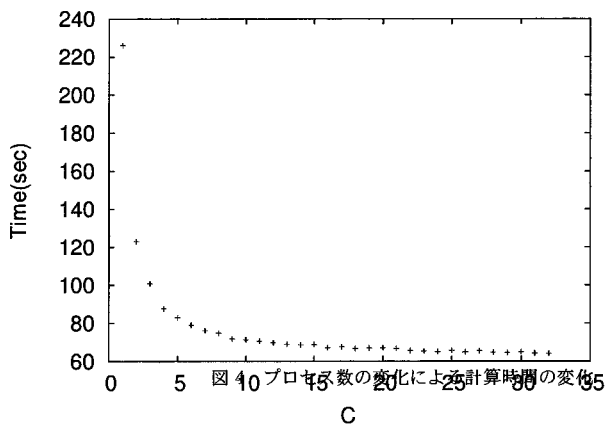


図3はアルゴリズム2の `g_mat_fac_itor` (モジュラーメソッド) をプロセス数 ($C=$)16, ($d_p=$)100 桁の素数 ($N_p=$)1000 個使用して実行したものである。計算時間の増加が N の線形時間となっていることが分かる。プロセス数の変化による実行時間の変化を計測したものが図4である。



プロセス数 C に対して実行時間はおおよそ $\frac{1}{C}$ の形になっており、一次変換の回数とプロセス数に関して理論値と一致する結果が得られた。次にモジュラーメソッドが有効であるかを見るため図2と図3を重ね合わせたグラフを掲載する。

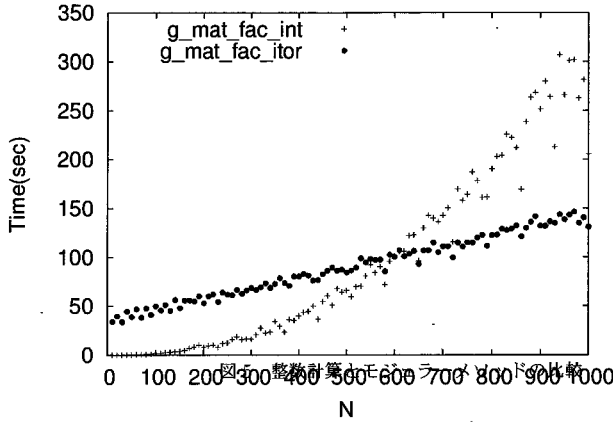


図5 整数計算とモジュラ計算の比較

今回の条件においては $N = 600$ あたりでモジュラーメソッドの実行時間が整数計算よりも短くなっている。定理 2 による理論的計算量の評価による漸近的な有効性のみならず，計算代数の分野で標準的な手法であるモジュラーメソッド ([6], [12] など) が差分ホロノミック勾配法の実際の適用にも有効な場合があることが確かめられた。

4.3 $5 \times 5, 7 \times 7$ 分割表の場合

次の図 6 は 5×5 分割表の結果で，このときの行列のサイズは $r = \binom{8}{4} = 70$ である。step1 はねじれコホモロジー群を用いた contiguity relations の導出の計算時間で，step2 においてモジュラーメソッドを用いて数値計算を行っている。

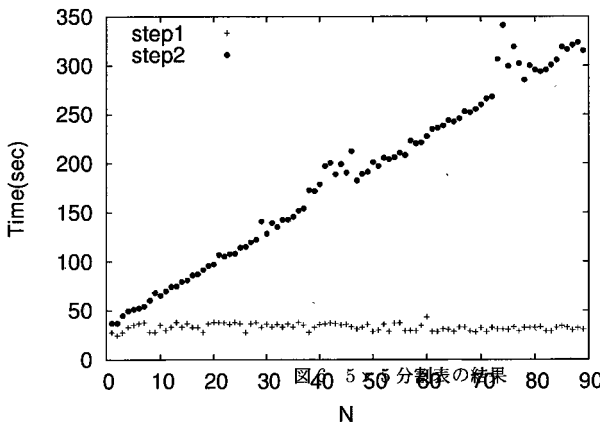


図6 5×5 分割表の結果

用いた周辺和は $(4N, 4N, 4N, 4N, 4N, 2N, 3N, 5N, 5N, 5N)^T$ で 100 桁の素数 200 個，8 個のプロセスを使

用して計算を行った。 N の線形時間で計算が実行されていることが分かる結果となっており 5×5 分割表の正規化定数の計算においても計算量評価の理論値と一致する結果を得ることができた。

最後に 7×7 分割表に対して計算を実行した場合である。このときの行列のサイズは $r = \binom{12}{6} = 924$ となる。周辺和は $(4N, 4N, 5N, 5N, 5N, 5N, 6N, 3N, 4N, 4N, 6N, 6N)^T$ で 100 桁の素数 200 個、16 個のプロセスを使用して計算を行った。計算に要する時間が膨大であるので $N = 1, 10, 20, 30$ におけるデータを計測した。 7×7 分割表の場合においても計算量評価の理論値と一致する結果が得られている。

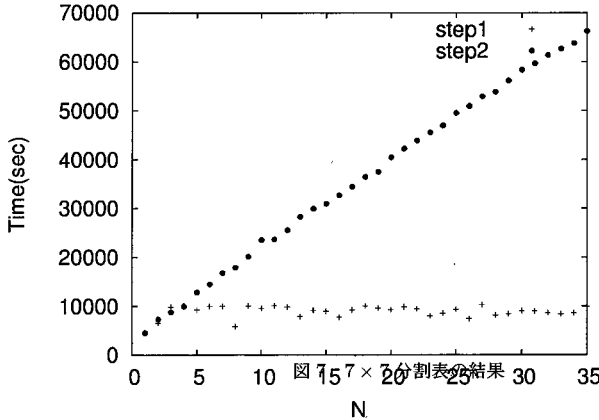


図 15 7×7 分割表の結果

5 Appendix

この Appendix では例 5 の行列 U_2 の導出を [1] および Risa/Asir パッケージ `gtt_ekn` のファイル `gtt_ekn/ekn_pfaffian.8.rr` の関数に従い解説する。

ここでは、 ${}_2F_1$ の積分表示

$$\int_0^1 t^{b-1}(1-t)^{c-b-1}(1-xt)^{-a} dt = \int_0^1 t^b(1-t)^{c-b-1}(1-xt)^{-a} \frac{dt}{t}$$

を利用する。このとき、

$$(\alpha_0, \alpha_1, \alpha_2, \alpha_3) = (a - c + 1, b, -a, c - b - 1)^{*2}$$

が対応するので、 a を減らすことは α_2 を増やす (α_0 を減らす) ことに対応 (よく使われる積分表示 (a と b を入れ替えたもの) を使うと、 $(\alpha_0, \alpha_1, \alpha_2, \alpha_3) \rightarrow (\alpha_0, \alpha_1 - 1, \alpha_2, \alpha_3 + 1)$ を考えなくてはいけないので、2 ステップ必要になる)。目標の式は、 $F(a) = M(a)F(a+1)$ なので、論文 [1, Corollary 6.3] の

$$S(\alpha; x) = \frac{1}{\alpha_2} U_2(\alpha_{(2)}; x) S(\alpha_{(2)}; x), \quad \alpha_{(2)} := (\alpha_0 + 1, \alpha_1, \alpha_2 - 1, \alpha_3)$$

が対応 ($\alpha_{(2)}$ は $a+1$ に対応している)。プログラムは $\frac{1}{\alpha_2} U_2$ を求めるように作ってある (`upAlpla(2, 1, 1)`)。 $S(\alpha; x)$ は [1, 6 章] で定義されている超幾何級数 $S(\alpha; x)$ とその偏微分が並ぶベクトル (Gauss-Manin ベクトル) であるが、 $c \in \mathbb{N}_0$ であれば

$$S(\alpha; x) = \begin{pmatrix} S \\ \frac{1}{\alpha_2} \theta_x S \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1/\alpha_2 \end{pmatrix} \begin{pmatrix} S \\ \theta_x S \end{pmatrix} = \frac{1}{(-a)!(-b)!(c-1)!} \begin{pmatrix} 1 & 0 \\ 0 & 1/\alpha_2 \end{pmatrix} \begin{pmatrix} {}_2F_1 \\ \theta_x {}_2F_1 \end{pmatrix}$$

^{*2} $\alpha_0 = -\alpha_1 - \alpha_2 - \alpha_3$ は無限遠点の exponent に対応。

と ${}_2F_1$ を用いて表示できる。差分作用素で係数が少しずれることに注意すれば、

$$M(a) = -a \begin{pmatrix} 1 & 0 \\ 0 & \alpha_2 \end{pmatrix} \left(\frac{1}{\alpha_2} U_2(\alpha_{(2)}) \right) \begin{pmatrix} 1 & 0 \\ 0 & 1/(\alpha_2 - 1) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & \alpha_2 \end{pmatrix} U_2(\alpha_{(2)}) \begin{pmatrix} 1 & 0 \\ 0 & 1/(\alpha_2 - 1) \end{pmatrix}$$

となることがわかる。[1, Theorem 5.3] より、表現行列 U_2 は

$$U_2(\alpha_{(2)}; x) = C(\alpha) P_2(\alpha)^{-1} D_2(x) Q_2(\alpha_{(2)}) C(\alpha_{(2)})^{-1}$$

と表示される。ここで、 $\tilde{x} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & x & 1 \end{pmatrix}$ の第 i, j 列 (0, 1, 2, 3 列と数える) を取り出した小行列を $\tilde{x}(ij)$ と書く記法などを用いて (詳しくは論文参照),

$$\begin{aligned} D_2(x) &= \text{diag} \left(\frac{|\tilde{x}(21)|}{|\tilde{x}(01)|}, \frac{|\tilde{x}(23)|}{|\tilde{x}(03)|} \right) = \text{diag}(1, 1-x) = \begin{pmatrix} 1 & 0 \\ 0 & 1-x \end{pmatrix}, \\ C(\alpha) &= \begin{pmatrix} \mathcal{I}(\varphi(01), \varphi(01)) & \mathcal{I}(\varphi(01), \varphi(02)) \\ \mathcal{I}(\varphi(02), \varphi(01)) & \mathcal{I}(\varphi(02), \varphi(02)) \end{pmatrix} = 2\pi\sqrt{-1} \begin{pmatrix} \frac{1}{\alpha_0} + \frac{1}{\alpha_1} & \frac{1}{\alpha_0} \\ \frac{1}{\alpha_0} & \frac{1}{\alpha_0} + \frac{1}{\alpha_2} \end{pmatrix}, \\ Q_2(\alpha) &= \begin{pmatrix} \mathcal{I}(\varphi(01), \varphi(01)) & \mathcal{I}(\varphi(01), \varphi(02)) \\ \mathcal{I}(\varphi(03), \varphi(01)) & \mathcal{I}(\varphi(03), \varphi(02)) \end{pmatrix} = 2\pi\sqrt{-1} \begin{pmatrix} \frac{1}{\alpha_0} + \frac{1}{\alpha_1} & \frac{1}{\alpha_0} \\ \frac{1}{\alpha_0} & \frac{1}{\alpha_0} \end{pmatrix}, \\ P_2(\alpha) &= \begin{pmatrix} \mathcal{I}(\varphi(21), \varphi(01)) & \mathcal{I}(\varphi(21), \varphi(02)) \\ \mathcal{I}(\varphi(23), \varphi(01)) & \mathcal{I}(\varphi(23), \varphi(02)) \end{pmatrix} = 2\pi\sqrt{-1} \begin{pmatrix} \frac{1}{\alpha_1} & -\frac{1}{\alpha_2} \\ 0 & -\frac{1}{\alpha_2} \end{pmatrix}. \end{aligned}$$

プログラムでは外積構造を用いて逆行列を求める方法 (`invintMatrix.n`) を用いて (詳細は略),

$$\begin{aligned} P_2(\alpha)^{-1} &= -\frac{\alpha_1\alpha_2}{(2\pi\sqrt{-1})^2} \begin{pmatrix} \mathcal{I}(\varphi(23), \varphi(02)) & -\mathcal{I}(\varphi(21), \varphi(02)) \\ -\mathcal{I}(\varphi(23), \varphi(01)) & \mathcal{I}(\varphi(21), \varphi(01)) \end{pmatrix} = -\frac{\alpha_1\alpha_2}{2\pi\sqrt{-1}} \begin{pmatrix} -\frac{1}{\alpha_0} & \frac{1}{\alpha_1} \\ 0 & \frac{1}{\alpha_1} \end{pmatrix}, \\ C(\alpha)^{-1} &= -\frac{\alpha_0\alpha_1\alpha_2}{(2\pi\sqrt{-1})^2\alpha_3} \begin{pmatrix} \mathcal{I}(\varphi(02), \varphi(02)) & -\mathcal{I}(\varphi(01), \varphi(02)) \\ -\mathcal{I}(\varphi(02), \varphi(01)) & \mathcal{I}(\varphi(01), \varphi(01)) \end{pmatrix} = -\frac{\alpha_0\alpha_1\alpha_2}{2\pi\sqrt{-1}\alpha_3} \begin{pmatrix} \frac{1}{\alpha_0} + \frac{1}{\alpha_2} & -\frac{1}{\alpha_0} \\ -\frac{1}{\alpha_0} & \frac{1}{\alpha_0} + \frac{1}{\alpha_1} \end{pmatrix} \end{aligned}$$

と計算している。[1, Appendix] で述べられている逆行列を計算する方法は `invintMatrix.k` として実装してある。この方法で計算すると、

$$\begin{aligned} P_2(\alpha)^{-1} &= \frac{1}{2\pi\sqrt{-1}} \begin{pmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{pmatrix} \begin{pmatrix} \frac{1}{\alpha_1} & -\frac{1}{\alpha_3} \\ 0 & -\frac{1}{\alpha_3} \end{pmatrix} \begin{pmatrix} \alpha_1 & 0 \\ 0 & \alpha_3 \end{pmatrix}, \\ C(\alpha)^{-1} &= \frac{1}{2\pi\sqrt{-1}} \begin{pmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{pmatrix} \begin{pmatrix} \frac{1}{\alpha_3} + \frac{1}{\alpha_1} & \frac{1}{\alpha_3} \\ \frac{1}{\alpha_3} & \frac{1}{\alpha_3} + \frac{1}{\alpha_2} \end{pmatrix} \begin{pmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{pmatrix} \end{aligned}$$

となる ($\alpha_0 = -\alpha_1 - \alpha_2 - \alpha_3$ に注意して整理すれば上と一致する)。ちなみに、上記の行列は

$$\begin{aligned} D_2(x) &= \text{repMatrix}(2, 1, 1), & C(\alpha)/(2\pi\sqrt{-1}) &= \text{intMatrix}([0, 3], [0, 3], 1, 1), \\ P_2(\alpha)/(2\pi\sqrt{-1}) &= \text{intMatrix}([2, 0], [0, 3], 1, 1), & Q_2(\alpha)/(2\pi\sqrt{-1}) &= \text{intMatrix}([0, 2], [0, 3], 1, 1), \\ (2\pi\sqrt{-1})P_2(\alpha)^{-1} &= \text{invintMatrix.n}([2, 0], [0, 3], 1, 1), & (2\pi\sqrt{-1})C(\alpha)^{-1} &= \text{invintMatrix.n}([0, 3], [0, 3], 1, 1) \end{aligned}$$

などとして計算させることができる (引数 $(1, 1)$ は $(r_1 - 1, r_2 - 1)$ に対応)。

参考文献

- [1] Goto, Y., Matsumoto, K., Pfaffian equations and contiguity relations of the hypergeometric function of type $(k+1, k+n+2)$ and their applications, arXiv:1602.01637.
- [2] JST CREST 日比チーム編, グレブナー道場, 共立出版, 2011.
- [3] LINBOX: exact computational linear algebra. <http://www.linalg.org>
- [4] Nakayama, H., Nishiyama, K., Noro, M., Ohara, K., Sei, T., Takayama, N., Takemura, A., Holonomic Gradient Descent and its Application to Fisher-Bingham Integral, Advances in Applied Mathematics 47 (2011), 639-658.

- [5] 野呂正行, 高山信毅, Risa/Asir ドリル 2012, <http://www.math.kobe-u.ac.jp/Asir>, 2012.
- [6] Noro, M., Yokoyama, K., A Modular Method to Compute the Rational Univariate Representation of Zero-Dimensional Ideals. *Journal of Symbolic Computation* 28/1(1999), 243-263.
- [7] 野呂正行, 横山和弘, グレブナー基底の計算 基礎篇 計算代数入門, 東京大学出版会, 2003.
- [8] Ohara, K., Takayama, N., Pfaffian Systems of A -Hypergeometric Systems II — Holonomic Gradient Method, arXiv:1505.02947.
- [9] Oshima, T., Fractional Calculus of Weyl Algebra and Fuchsian Differential Equations, *MSJ Memoirs* 28, 2013.
- [10] References for HGM
<http://www.math.kobe-u.ac.jp/OpenXM/Math/hgm/ref-hgm.html>
- [11] Risa/Asir, a computer algebra system. <http://www.math.kobe-u.ac.jp/Asir>
- [12] Sasaki, T., Takeshima, T., A modular method for Gröbner-basis construction over $\mathbb{Q}$ and solving system of algebraic equations. *Journal of Information Processing*, 12(1989), no.4, 371-379.
- [13] Takayama, N., Gröbner basis and the problem of contiguous relations, *Japan Journal of Applied Mathematics*, 6 (1989), 147-160.
- [14] Takayama, N., Kuriki, S., Takemura, A., A -hypergeometric distributions and Newton polytopes. arXiv:1510.02269