

行列超幾何関数 2F1 の数値計算とその応用
高山信毅 (神戸大学)

参考文献等 [hgm OpenXM search](#).

[Wishart 分布](#): Y : $m \times m$ non-negative definite 対称行列.

$$\frac{1}{N(\Sigma, m, n)} |Y|^{(n-m-1)/2} \exp(-\text{Tr}(\Sigma^{-1}Y/2)) \prod_{i \leq j} dY_{ij}$$

$$N(\Sigma, m, n) = 2^{mn/2} |\Sigma|^{n/2} \Gamma_m(n/2)$$

$\Sigma > 0$: $m \times m$ positive definite 対称行列. $n \geq m$.

$$\Gamma_m(a) = \pi^{\frac{1}{4}m(m-1)} \prod_{i=1}^m \Gamma\left(a - \frac{i-1}{2}\right).$$

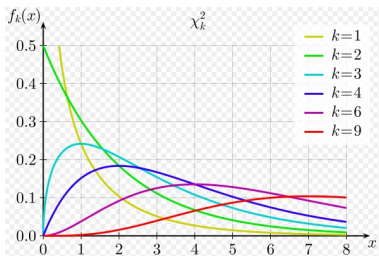
R.Muirhead, Aspects of Multivariate Statistical Theory, 2005. p.85.

Wishart 分布 $X : n \times m$ 行列 ($n \geq m$) . 行は m 次元の正規分布に従う random variable (vector) (独立, 同分布, 平均は 0, covariance matrix は Σ). $m \times m$ 値 random variable (行列) $X^T X$ に対応する確率分布が自由度 n の Wishart 分布. 特に $m = 1, \Sigma = 1$ の時は χ^2 -分布である:

$$H(t)t^{\frac{n}{2}-1} \exp\left(-\frac{t}{2}\right) / Z(k), \quad Z(k) = 2^{n/2}\Gamma(n/2)$$

$H(t)$ は $t > 0$ の特性関数 (Heaviside 関数).

χ^2 -分布密度関数 (Wikipedia より, $k = n$ (自由度))



$m = 1$ の時の証明の sketch

- X_1, X_2 : \mathbf{R} -valued random variable. $P(X_i \in [x, x + h]) \sim p_i(x)h$ ¹, すなわち $p_i(x)$ が X_i の PDF (probability distribution function) とする. X_i 達は独立.

$$P(X_1 = x_1, X_2 = x_2) = p_1(x_1)p_2(x_2)dx_1dx_2$$

$x_1 + x_2 = u, x_2 = v$ と変数変換.

$$P(X_1 + X_2 = u, x_2 = v) = P(X_1 = u - v, X_2 = v) = p_1(u - v)p_2(v)dudv$$

なので

$$P(X_1 + X_2 = u) = du \int_{-\infty}^{\infty} p_1(u - v)p_2(v)dv$$

- X : random variable で $P(X = x) \sim \exp(-\Sigma^{-1}x^2/2)dx$ なら

$$P(X^2 = y) \sim \frac{1}{2} \exp(-\Sigma^{-1}y/2)y^{-1/2}dy$$

¹ X_i が $[x, x + h]$ に値をもつ確率は $p_i(x)h$

Wishart 分布の第一固有値の分布 $\lambda_1(W)$ を行列 W の最大固有値とする.

問題: W が Wishart 分布に従うとき

$$P(\lambda_1(W) < q)$$

を計算せよ.

答え (Constantine 1963, James 1964, Muirhead 1970): Wishart 分布の積分で上記の確率は書ける. 積分は matrix hypergeometric function ${}_1F_1$ で表現できて, パラメータについて線形微分方程式系を満たす.

自由度 n , $m \times m$ 共分散行列 Σ できまる Wishart 分布に従う $m \times m$ 行列の最大固有値 ℓ_1 が x より小さい確率² は

$$P[\ell_1 < x] = C \exp\left(-\frac{x}{2} \text{Tr} \Sigma^{-1}\right) x^{\frac{1}{2}nm} {}_1F_1\left(\frac{m+1}{2}; \frac{n+m+1}{2}; \frac{x}{2} \Sigma^{-1}\right) \quad (1)$$

と書ける. C は定数. ${}_1F_1$ は行列引数の超幾何関数で次の積分で定義される.

$${}_1F_1(a; c; X) = G \int_{0 < Y < I_m} \exp(\text{Tr} XY) |Y|^{a-(m+1)/2} |I_m - Y|^{c-a-(m+1)/2} dY, \quad (2)$$

$G = \frac{\Gamma_m(c)}{\Gamma_m(a)\Gamma_m(c-a)}$. なお $0 < Y < I_m$ は行列 Y および $I_m - Y$ が正定値対称行列であることを意味. $dY = \prod_{i \leq j} dY_{ij}$,

$$\Gamma_m(a) = \pi^{\frac{1}{4}m(m-1)} \prod_{i=1}^m \Gamma\left(a - \frac{i-1}{2}\right).$$

²Q. むしろ不完全 Γ の一般化では? A. 不完全 Γ は ${}_1F_1$ で書ける

${}_2F_1$ は統計に登場しないんですか?

W_i : 自由度 n_i , covariance Σ_i に従う Wishart 分布.

Theorem

$\Sigma_1 = I_m$ とする. $U = W_2^{-1/2} W_1 W_2^{-1/2} \geq 0$ の密度関数は以下のようになる.

$$f(U) = \frac{\Gamma_m\{(n_1 + n_2)/2\} |\Sigma_2|^{n_1/2}}{\Gamma_m(n_1/2) \Gamma_m(n_2/2)} |I + \Sigma_2 U|^{-(n_1+n_2)/2} |U|^{(n_1-m-1)/2}. \quad (3)$$

$m = 1$ の場合の証明

$$g(w_1, w_2) = w_1^{n_1/2-1} \exp(-w_1/2) w_2^{n_2/2-1} \exp(-w_2\sigma^{-1}/2) dw_1 dw_2$$

に対して, 変数変換

$$u = w_1 w_2^{-1}, \quad v = w_2$$

を適用. $dw_1 dw_2 = v du dv$ なので, $g(w_1, w_2)$ は

$$\begin{aligned} f(u, v) &= (uv)^{n_1/2-1} \exp(-uv/2) v^{n_2/2-1} \exp(-v\sigma^{-1}/2) v du dv \\ &= u^{n_1/2-1} v^{\frac{n_1+n_2}{2}-1} \exp(-(u + \sigma^{-1})v/2) du dv \end{aligned}$$

u についての密度関数を求めるには v について積分.

$\int_0^\infty v^\alpha \exp(-yv) dv = \Gamma(\alpha + 1) y^{-(\alpha+1)}$ を用いて,

$$\int_0^\infty f(u, v) dv \sim u^{n_1/2-1} ((u + \sigma^{-1})/2)^{-\frac{n_1+n_2}{2}}$$

$m = 1$ では $0 \leq u \leq x$ の範囲での積分は不完全ベータ関数. これは ${}_2F_1$ で書ける.

$$\begin{aligned}
 & {}_2F_1(a, b; c; X) \\
 = & \frac{\Gamma_m(c)}{\Gamma_m(a)\Gamma_m(c-a)} \\
 & \cdot \int_{0_m < Y < I_m} |Y|^{a-(m+1)/2} |I_m - Y|^{c-a-(m+1)/2} |I_m - XY|^{-b} (dY),
 \end{aligned}$$

ここで $X \in \mathbf{C}^{m \times m}$ は symmetric で $\operatorname{Re}(X) < I_m$ を満たす. また $\operatorname{Re}(a) > (m-1)/2$, $\operatorname{Re}(c-a) > (m-1)/2$.

hgm.p2wishart.

Theorem

自由度 n_i , $m \times m$ 共分散行列 Σ_i できまる Wishart 分布に従う $m \times m$ 行列 W_i .

$$\begin{aligned}
 & P(\ell_1(W_1 W_2^{-1}) \leq x) \\
 = & \frac{\Gamma_m\left(\frac{m+1}{2}\right)\Gamma_m\left(\frac{n_1+n_2}{2}\right)}{\Gamma_m\left(\frac{n_1+m+1}{2}\right)\Gamma_m\left(\frac{n_2}{2}\right)} x^{mn_1/2} |\Sigma_2 \Sigma_1^{-1}|^{n_1/2} \\
 & \cdot {}_2F_1\left(\frac{n_1}{2}, \frac{n_1+n_2}{2}; \frac{n_1+m+1}{2}; -x \Sigma_1^{-1} \Sigma_2\right).
 \end{aligned}$$

$$\Gamma_m(a) = \pi^{\frac{1}{4}m(m-1)} \prod_{i=1}^m \Gamma\left(a - \frac{i-1}{2}\right).$$

Y.Chikuse, Asymptotic expansions for the joint and marginal distributions of the latent roots of $S_1 S_2^{-1}$, Annals of the Institute of Statistical Mathematics 29 (1977), 221–233

シミュレーションのための R のコード

```
rp2wishart
<-function(lambda=4,n1=5,n2=10,sigma=diag(c(1/1,1/2,1/4)),try=100) {
  count <- 0;
  s1<-diag(1,nrow(sigma),ncol(sigma));
  s2<-sigma
  r1<-rWishart(try,df=n1,Sigma=s1);
  r2<-rWishart(try,df=n2,Sigma=s2);
  for (k in seq(1,try)) {
    e<-eigen(r2[, ,k]); v<-e$vectors;
    b<-v %*% diag(1/sqrt(e$values)) %*% t(v);
    rr<-b %*% r1[, ,k] %*% b;
    ell1 <- max(eigen(rr)$values);
    if (ell1 < lambda) count<-count+1;
  }
  return(count/try)
}
```

hgm package (あとで).

```
t<-hgm.p2wishart(autoplot=1);
f<-splinefun(t);
curve(f,from=0,to=10);
```

${}_2F_1$ の数値計算

橋口, 高山, 竹村, (2018) では,

1. 小さい q (q_0) で上記の値を ${}_2F_1$ の級数展開から計算 (Koev-Edelman 2006).
2. q_0 での値を微分方程式を数値解析することにより大きい q へのばす.



hgm package に実装.

Q. R の hgm package は何をやるソフトウェア?

A. パラメータ付きの積分が満す線形微分方程式を数値解析して、積分の値を計算する Holonomic Gradient Method (HGM) のソフトウェア.

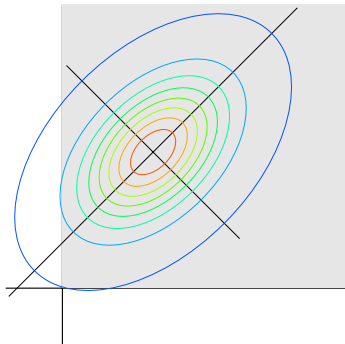
著者: 高山信毅, 小山民雄, 清智也, 小原功任, 中山洋将.

インストールの方法. `install.packages('hgm');`
Version 1.16.

何が計算できるの?

1. Wishart 分布に従う正定値対称行列の最大固有値の確率.
matrix $_1F_1$ の計算. `hgm.pwishart`
2. Wishart 分布に従う正定値対称行列二つの比の最大固有値の
確率. matrix $_2F_1$ の計算. `hgm.p2wishart`
3. A, S を行列とするとき, $\exp(\text{tr}(AS))$ を $A \in SO(3)$ で積分す
る. Fisher 積分. `hgm.ncso3`

1. 多変数の正規分布を第一象限で積分
する. `hgm.ncorthant`
2. 平均が原点の多変数の正規分布を超
球表面で積分する. `hgm.ncBingham`
3. 準備中 (小山): 多変数の正規分布を超
球内部で積分する.
4. 準備中 (小山): 多変数の正規分布を単
体で積分する.



hgm パッケージではどう計算?

Wishart matrices の比の固有値. ${}_2F_1$ の場合.

```
library(hgm)
hgm.p2wishart(m=3,n1=5,n2=10,beta=c(1,2,4),q=4)
[1] 4.0240291 0.5779336 13.1008398 7.2411721 163.8297748
[6] 4.3388397 98.0285307 54.0157128 1217.7748159
```

なぜ hgm?

1. hgm の高精度高速計算によって乱数によるシミュレーションを評価できる.
2. 乱数によるシミュレーションでできないことが短時間で出来ることがある.
3. 関数の数値計算したい.

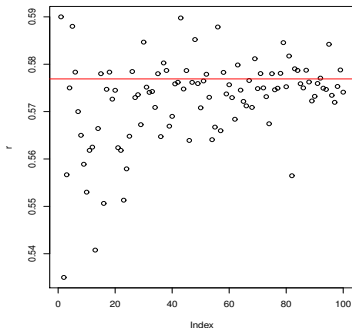
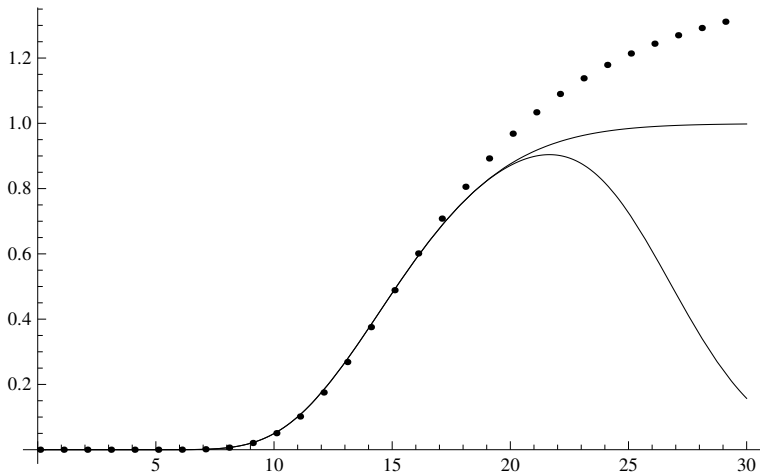


Figure: Wishart matrices の比,
10,000 回までのシミュレーション

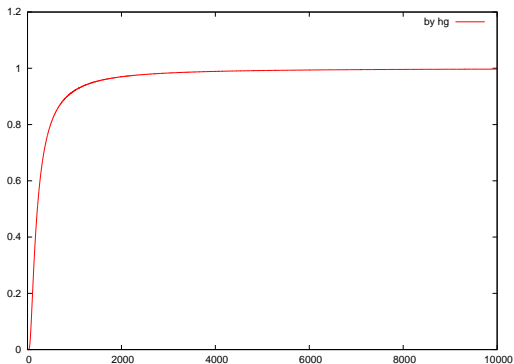
HGM の利点

$m = 2, n = 30, \Sigma = \text{diag}(1/2, 1/4)$. 上から Laplace 近似, HGM, 級数 (50 次まで).



Wishart 行列の比の第一固有値, 行列引数超幾何系 ${}_2F_1$

累積分布関数 $P[\ell_1 < x]$, $m = 10, n_1 = 11, n_2 = 12$,
 $\Sigma_2^{-1}\Sigma_1 = \text{diag}(1, 2, \dots, 10)$



出典: 橋口, 高山, 竹村, Distribution of ratio of two Wishart matrices and evaluation of cumulative probability by holonomic gradient method, <https://doi.org/10.1016/j.jmva.2018.01.002>

Q. 自由度を大きくすると default 値ではうまく行かない. どうすればいい?

```
plot(hgm.pwishart(m=3,beta=c(1,2,3),n=100,q0=0.3,q=10,  
                err=c(1e-30,1e-10),autoplot=1))
```

```
%%Warning, abserr seems not to be small enough, abserr=1e-30, y[0]=3.43846e-30.  
Increasing the starting point (q0 or X0g(standalone case)) may  
or making abserr (err[1] or abserror(standalone case)) smaller will help,  
e.g., err=c(1e-20,1e-10)  
Error: x0 < x must hold.  
スタート点 q0 での値が小さすぎる. q0 が自動的に増えるのだが q=10 を越した.  
hgm.pwishart(n = 100, q0 = 0.3, q = 10, err = c(1e-30, 1e-10), でエラー:  
mh_exit(-30) is called.
```

q を増やす. (q0 が自動で増えた場合.) 参考 verbose=1

```
plot(hgm.pwishart(n=100,q0=0.3,q=20,err=c(1e-30,1e-10),autoplot=1))
```

```
%%Warning, abserr seems not to be small enough, abserr=1e-30, y[0]=3.43846e-30.  
Increasing the starting point (q0 or X0g(standalone case)) may  
or making abserr (err[1] or abserror(standalone case)) smaller will help,  
e.g., err=c(1e-20,1e-10)
```

err を減らす

```
> plot(hgm.pwishart(n=100,q0=0.3,q=20,err=c(1e-40,1e-10),autoplot=1))  
> plot(hgm.pwishart(n=100,q0=0.3,q=200,err=c(1e-40,1e-10),autoplot=1))
```


Q. 自由度を大きくすると default 値ではうまく行かない. どうすればいい?

q0 と err を上手に調整.

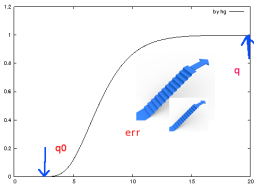
```
> plot(hgm.pwishart(m=5,n=20,beta=c(1,2,3,4,5),autoplot=1,q=100))  
%%Warning, abserr seems not to be small enough, abserr=1e-40,  
y[0]=1.4645e-53.
```

初期値が $1e-53$ なので, abs err を小さくしてみる.

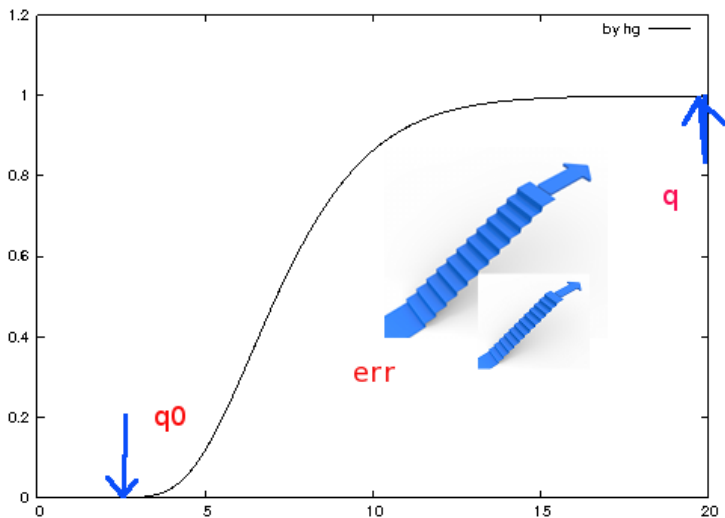
```
plot(hgm.pwishart(m=5,n=20,beta=c(1,2,3,4,5),  
err=c(1e-70,1e-20),autoplot=1,q=100))
```

これは時間がかかる. q0 を増やし, err をすこし大きく.

```
plot(hgm.pwishart(m=5,n=20,q0=0.8,  
beta=c(1,2,3,4,5),  
err=c(1e-30,1e-10),autoplot=1,q=100))
```



q0, err, q の調整 (よりよい自動化は今後の課題)



hgm の原理を χ^2 分布で説明.

$$T_n(x) = \begin{cases} x^{\frac{n}{2}-1} e^{-\frac{x}{2}} / N_T(n) & x > 0 \\ 0 & x \leq 0 \end{cases}$$

$N_T(n)$ は、上の関数を \mathbf{R} で積分したときに 1 になるための正規化定数で

$$N_T(n) = \int_0^{\infty} x^{\frac{n}{2}-1} e^{-\frac{x}{2}} = 2^{\frac{n}{2}} \Gamma\left(\frac{n}{2}\right)$$

$T_n(x)$ を確率密度関数とする分布は χ^2 分布.

$$\gamma(x) = \int_0^x t^{\frac{n}{2}-1} e^{-\frac{t}{2}} dt$$

数値計算をしたい. Heaviside 関数を用いて

$$\gamma(x) = \int_{-\infty}^{\infty} H(x-t)H(t)t^{\frac{n}{2}-1} e^{-\frac{t}{2}} dt$$

Step 1. $\gamma(x)$ の満たす線形微分方程式を求める.(詳細)

(積分表示を元に

して)

$$\begin{aligned}\ell_1 &= (x-t) \left(t\partial_t - \left(\frac{n}{2} - 1 \right) + \frac{1}{2}t \right) \\ \ell_2 &= (x-t)\partial_x\end{aligned}$$

が積分核 f の満たす方程式系.

微分作用素環 $D = K\langle x, t, \partial_x, \partial_t \rangle$ の左イデアル $I = D\ell_1 + D\ell_2$ を考える. I に属する微分作用素は f を零化. (つまり $\ell \in I$ なら $\ell \bullet f = 0$ となる).

$(I + \partial_t D) \cap K\langle x, \partial_x \rangle \ni \ell \neq 0$ とする.

$$\ell = P_1 + \partial_t P_2, \quad (P_1 \in I, P_2 \in D) \quad (4)$$

と書けるので,

$$\begin{aligned}\ell \bullet \int_{-\infty}^{\infty} f(x, t) dt &= \int_{-\infty}^{\infty} P_1 f(x, t) dt + \int_{-\infty}^{\infty} \partial_t P_2 f(x, t) dt \\ &= [P_2 f]_{-\infty}^{\infty} = 0\end{aligned}$$

となって, $\ell \bullet \gamma(x) = 0$

ここで ℓ は実は $\ell = 2x\partial_x^2 + (x-n+2)\partial_x$. (グレブナー基底を使う)

```
import("nk_restriction.rr");
L=[(x-t)*(t*dt-(n/2-1)+t/2),
  (x-t)*dx];
G=nk_restriction.integration_ideal(L, [t,x], [dt,dx], [1,0]);
```

Step 2. ℓ を Pfaffian へ. (多変数ではグレブナー基底を使う.)(詳細)

$$\partial_x \begin{pmatrix} \gamma(x) \\ \partial_x \gamma(x) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & -\frac{1}{2x}(x-n+2) \end{pmatrix} \begin{pmatrix} \gamma(x) \\ \partial_x \gamma(x) \end{pmatrix}$$

Pfaffian 形式が有用な理由

が成り立つとする.

上記のように $\partial_x F(x) = P(x)F(x)$

$$F(x+h) = F(x) + h\partial_x F(x) + O(h^2)$$

より

$$F(x+h) \simeq F(x) + hP(x)F(x)$$

誤差が $O(h^{m+1})$ になるようにしたのが m -次の Runge-Kutta 法.

Step 3. 級数展開、数値積分などを初期値として **Step 2** の Pfaffian を Runge-Kutta 法で数値解析して、広い範囲の積分値を求める。

(matrix ${}_pF_q$ では初期値計算に Koev-Edelman 2006)

$\gamma(x)$ の冪級数展開を求めると、

$$\begin{aligned} \int_0^x t^{\frac{n}{2}-1} e^{-\frac{t}{2}} dt &= \sum_{k=0}^{\infty} \int_0^x t^{\frac{n}{2}-1} \frac{1}{k!} \left(-\frac{t}{2}\right)^k dt \\ &= \sum_{k=0}^{\infty} \frac{1}{k!} \left(-\frac{1}{2}\right)^k \int_0^x t^{\frac{n}{2}-1+k} dt \\ &= \sum_{k=0}^{\infty} \frac{1}{k!} \left(-\frac{1}{2}\right)^k \frac{x^{\frac{n}{2}+k}}{\frac{n}{2}+k} \\ &= \frac{n}{2} x^{\frac{n}{2}} {}_1F_1\left(\frac{n}{2}; \frac{n}{2}+1; -\frac{x}{2}\right) \end{aligned}$$

注意: 自由度 n が大きくなってくると、単純に Runge-Kutta 法を適用するだけでは、精度が得られなくなってくる。このような場合には、 $\gamma(x) = e^{-\frac{x}{2}} x^{\frac{n}{2}} h(x)$ と置いて、 $h(x)$ に対する方程式 $\theta(\theta + \frac{n}{2}) - \frac{x}{2}(\theta + 1)$ を数値解析するなどの工夫が必要になる。

Wishart 行列の比の第一固有値, 行列引数超幾何系 ${}_2F_1$

${}_2F_1(a, b; c; X)$ の X が対角成分が x_1, \dots, x_m である対角行列であるとする. Muirhead (1970) は ${}_2F_1$ が満たす次の微分方程式 $g_i \bullet {}_2F_1 = 0$ を導いた.

$$g_i = \underline{\partial_i^2} + [p(x_i) + \sum_{j \neq i} q_2(x_i, x_j)] \partial_i - \sum_{j \neq i} q(x_i, x_j) \partial_j - r(x_i), \quad i = 1, \dots, m,$$

ここで

$$\begin{aligned} p(x_i) &= \frac{c - (m-1)/2 - (a + b + 1 - (m-1)/2)x_i}{x_i(1-x_i)}, \\ q_2(x_i, x_j) &= \frac{1}{2(x_i - x_j)}, \\ q(x_i, x_j) &= \frac{x_j(1-x_j)}{2x_i(1-x_i)(x_i - x_j)}, \\ r(x_i) &= \frac{ab}{x_i(1-x_i)}. \end{aligned}$$

どのように Runge-Kutta 法を適用するか?

Theorem

集合 $\{g_1, \dots, g_m\}$ は *rational function* 係数の微分作用素環 $R_m = \mathbb{C}(x_1, \dots, x_m)\langle \partial_1, \dots, \partial_m \rangle$ において *graded reverse lexicographic order* でのグレブナー基底となっている.

Proof sketch. $G_i := x_i(1 - x_i)g_i$.

$$[G_i, G_j] = \frac{1}{2} \frac{2x_i x_j - x_i - x_j}{(x_i - x_j)^2} (G_i - G_j).$$

rank は 2^m .

Before sparsity: Pfaffian の行列を C 言語のプログラムとして生成する asir のプログラムを書いて C 言語のプログラムで数値解析³. 欠点: 行列掛けるベクトルの計算量は $O(m(2^m)^2)$.

After sparsity: 計算量 $O(m2^m)$. C 言語だけで実装可能. この実装ではかなり高速になっているのが実感.

³20180827-for-sapporo-ref

M : the left ideal of R_m generated by g_1, \dots, g_m .

$$\partial_i F \equiv P_i(x)F \pmod{M}$$

$F = (F_i) = (F_0, F_1, \dots, F_{2^m-1})^T$ の要素は $\partial^\alpha = \prod_{k=1}^m \partial_k^{\alpha_k}$,
 $\alpha = (\alpha_1, \dots, \alpha_m)$. ここで α は i を 2 進展開した係数を次のようにベクトルにしたもの: $i = \sum_{k=0}^{m-1} \alpha_{k+1} 2^k$, $\alpha_k \in \{0, 1\}$. Pfaffian の行列 P_i は次のように帰納的に定義する. I を $[m] = \{1, \dots, m\}$ の部分集合とする. $\partial_I := \prod_{k \in I} \partial_k$. $\partial_i \partial_I$ を F の元で表したい. $i \notin I$ なら $\partial_i \partial_I = \partial_{I'}$, $I' = \{i\} \cup I$ で $\partial_{I'} \in F$. P_i の対応する元は unit vector. $i \notin J \subset [m]$ とする. $I := \{i\} \cup J$. $\partial_i \partial_I$ modulo M を求めたい. ∂_J を次のように g_i に作用.

$$\partial_J g_i = \partial_i^2 \partial_J + p(x_i) \partial_i \partial_J + \partial_J \sum_{j \neq i} q_2(x_i, x_j) \partial_i - \partial_J \sum_{j \neq i} q(x_i, x_j) \partial_j - r(x_i) \partial_J.$$

When $k \notin J$, $\partial_J q_2(x_i, x_k) \partial_i = q_2(x_i, x_k) \partial_I$. When $k \in J$,
 $\partial_J q_2(x_i, x_k) \partial_i = q_2(x_i, x_k) \partial_I + \frac{\partial q_2(x_i, x_j)}{\partial x_k} \partial_{I \setminus \{k\}}$. When $k \notin J$,
 $\partial_J q(x_i, x_k) \partial_k = q(x_i, x_k) \partial_{J \cup \{k\}}$. When $k \in J$,
 $\partial_J q(x_i, x_k) \partial_k = q(x_i, x_k) \partial_{J \setminus \{k\}} \partial_k^2 + \frac{\partial q(x_i, x_k)}{\partial x_k} \partial_J$. まとめると $i \notin J$
 の時

$$\begin{aligned}
 \underline{\partial_i^2 \partial_J} &+ p(x_i) \partial_I + \sum_{k \neq i} q_2(x_i, x_k) \partial_I + \sum_{k \neq i, k \in J} \frac{\partial q_2(x_i, x_k)}{\partial x_k} \partial_{I \setminus \{k\}} \\
 &- \sum_{k \neq i, k \notin J} q(x_i, x_k) \partial_{J \cup \{k\}} - \sum_{k \neq i, k \in J} \underline{\partial_{J \setminus \{k\}} \partial_k^2} \\
 &- \sum_{k \neq i, k \in J} \frac{\partial q(x_i, x_k)}{\partial x_k} \partial_J - r(x_i) \partial_J \equiv 0 \pmod{M}
 \end{aligned}$$

下線を引いた項 $\partial_i^2 \partial_J$ は F の元およびより小さい下線を引いた同じタイプの項 $\partial_{J \setminus \{k\}} \partial_k^2$ で書けている。

ソースコード hgm.tar.gz hgm/R (R 言語), hgm/src (C 言語), hgm/man (マニュアル), DESCRIPTION, NAMESPACE(百聞は一見にしかず) ソースコード, ほとんどが C.

1. gsl より借用してきた adaptive Runge-Kutta の code. t- で始まる. wc で 1873 行
2. matrix ${}_pF_q$ 関連. jack-n.c, code-n*.c wc で 5085 行
3. 象限積分. hgm_ko_orthant.c wc で 782 行
4. Fisher 積分. so3_nc.c wc で 2233 行 (テーブルデータも含む)
5. Bingham 分布関連. se_bingham.R wc で 100 行
6. R と C 部分とのインタフェース. wc で 303 行

Rじゃない hgm パッケージ, n_wishartd for asir. 野呂,

$P(\ell_1(W) < x)$. covariance matrix の固有値に重複度がある場合.

```
import("n_wishartd.rr");
n_wishartd.message(1);
P
=n_wishartd.prob_by_hgm(10,100,[9,1],[1/100,1],100|eps=10^(-6));
// m, n, multi, eigen, P(L_1<100)
...
0.516246912217004
266.4sec + gc : 8.277sec(276.8sec)
```

M.Noro. System of partial differential equations for the hypergeometric function ${}_1F_1$ of a matrix argument on diagonal regions. In ISSAC'16 Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation, 381-388, 2016.

$x_1 = x_2$ などへの微分方程式の制限を計算する方法がポイント.