

# 高精度 GLP 公式の設計

2002MM075 太田 裕久

指導教員 杉浦 洋

## 1 はじめに

本論文では、GLP 公式の性質を理解し、現在確認されている 3 次元問題での世界記録を更新するような BLP の探索をし、これを C 言語で求め、GLP 公式の精度がどのように良くなっていくのかをグラフで表す。

## 2 メリット数 ( $\rho_s(\mathbf{g}, N)$ )

$R(\mathbf{h}) = \prod_{i=1}^s \max\{1, |h_i|\}$  としたとき、GLP 法の定義より、 $\mathbf{g} \in \mathbf{Z}^s$  は  $\mathbf{g} \in (0, N-1]^s$  と制限することができるので、

$$\rho_s(\mathbf{g}, N) \stackrel{def}{=} \min_{\mathbf{h} \in \tilde{H}_s(\mathbf{g}, N)} R(\mathbf{h})$$

( $N$ : 標本点数,  $\mathbf{g}$ : 自然数ベクトル)

$$\tilde{H}_s(\mathbf{g}, N) = (-N/2, N/2]^s \cap H_s(\mathbf{g}, N)$$
$$H_s(\mathbf{g}, N) = \{\mathbf{h} | \mathbf{h} \in \mathbf{Z}^s, \mathbf{h} \cdot \mathbf{g} \equiv 0 \pmod{N}, \mathbf{h} \neq \mathbf{0}\}$$

となる。メリット数  $\rho_s(\mathbf{g}, N)$  が大きくなるということは、積分近似の精度が良くなる ( $\mathbf{g}, N$ ) であることを示している。メリット数  $\rho_s(\mathbf{g}, N)$  は  $O(N^s)$  の計算量で計算できる。

## 3 Best Lattice Point(BLP)

BLP とは、メリット数  $\rho_s(\mathbf{g}, N)$  に関して、 $\mathbf{g} = (1, g_2, \dots, g_s)$  型の LP で最良のもので、

$$\rho_s(\mathbf{g}, N) = \max_{\mathbf{g}'} \rho_s(\mathbf{g}', N) > \max_{N' < N} \max_{\mathbf{g}'} \rho_s(\mathbf{g}', N') \quad (1)$$

を満たす ( $\mathbf{g}, N$ ) のことを呼ぶ。

## 4 BLP の探索手法

BLP 探索を最も単純に行う方法は、各  $N$  について全ての  $\mathbf{g}$  につき、メリット数  $\rho_s(\mathbf{g}, N)$  を計算し、その最大値を求める方法である。しかし、このままだと探索速度が遅いので、BLP の探索速度を上げるために、3 つの高速化手法を使う。まず 1 つ目は、あきらかに BLP となり得ない  $\mathbf{g}$  を早い段階で除外することで無駄な処理を減らし、2 つ目は、 $\mathbf{h}$  の探索範囲を限定し、3 つ目は、共通計算を括り出すことで探索速度を速くする。

BLP の基本的探索法は、標本点数  $N$  を 1 ずつ増加させながら式 (1) の条件を満たす ( $\mathbf{g}, N$ ) を次々に探索していくというものである。

先ほど述べた高速化手法のなかで最も重要なものが、共通計算を括り出すことであり、具体的には、 $\rho_s(\mathbf{g}, N)$  を

関数表を利用して計算する。ここでは、3 次元の場合を例にして示すことにする。

$$\rho_3(\mathbf{g}, N) = \min_{\mathbf{h} \in \tilde{H}_3(\mathbf{g}, N)} R(\mathbf{h})$$
$$= \min_{\mathbf{h} \in \tilde{H}_3(\mathbf{g}, N)} R(h_1, h_2)R(h_3)$$

最右辺で  $\mathbf{h} \in \tilde{H}_3(\mathbf{g}, N)$  を *i*)  $h_3 = 0$ , *ii*)  $h_3 \neq 0$  と 2 通りに場合分けをすると、

$$\rho_3((1, g_2, g_3), N) = \min\left\{ \min_{(h_1, h_2) \in \tilde{H}_2((1, g_2), N)} R(h_1, h_2), \tilde{\rho}_3((1, g_2, g_3), N) \right\}$$

となる。ちなみに第 1 項は、

$$\rho_2((1, g_2), N) = \min_{(h_1, h_2) \in \tilde{H}_2((1, g_2), N)} R(h_1, h_2)$$

を表しており、ユークリッドの互除法を利用すれば簡単に求めることが出来る。第 2 項は、

$$\tilde{\rho}_3((1, g_2, g_3), N) = \min_{\substack{(h_1, h_2, h_3) \in \tilde{H}_3((1, g_2, g_3), N) \\ h_3 \neq 0}} |h_3| R(h_1, h_2)$$

である。

$\tilde{\rho}_3((1, g_2, g_3), N)$  を求めることは 3 次元探索の問題であるが、関数  $T(r)$ (次節で解説する) を次のように定義する。

$$T(r) \stackrel{def}{=} \min_{h_1 + h_2 g_2 \equiv r \pmod{N}} R(h_1, h_2) \quad (2)$$

すると、

$$\tilde{\rho}_3((1, g_2, g_3), N) = \min_{\substack{h_3 \\ h_3 \neq 0}} |h_3| T(r(h_3))$$

となる。ただし、 $r(h_3) \equiv -g_3 h_3 \pmod{N}$  である。

つまり  $\tilde{\rho}_3((1, g_2, g_3), N)$  を求めるには、あらかじめ関数  $T(r)$  の表が作成されていれば、 $h_3$  の 1 次元探索に帰着する。 $T(r)$  は、定義式 (2) より ( $g_2, N$ ) のみに依存する関数であることがわかる。[1]

## 5 関数表 $T(r)$ の作成方法

定義式 (2) より、関数  $T(r)$  は  $r$  に関して周期  $N$  の偶関数であるから、 $0 \leq r \leq [N/2]$  で  $T(r)$  を求めて表にすることができる。また、 $N \geq 2$  とする。

$T(r) \leq R(r, 0) \leq [N/2]$  であり、 $|h_2| > [N/2]$  なら  $R(h_1, h_2) = R(h_1)R(h_2) > [N/2]$  となるから、

$$T(r) = \min_{|h_2| \leq [N/2]} \left\{ \min_{h_1 + h_2 g_2 \equiv r \pmod{N}} R(h_1)R(h_2) \right\}$$
$$= \min_{|h_2| \leq [N/2]} \left\{ R(h_2) \min_{k \in \mathbf{Z}} R(r - h_2 g_2 + kN) \right\}$$

関数  $R$  の定義より,

$$T(r) = \min_{0 \leq h_2 \leq [N/2]} \{R(h_2)R(\min_{k \in \mathbf{Z}} |r \pm h_2 g_2 + kN|)\}$$

$N$  による  $hg_2$  の剰余を  $\beta$  とすると,  $0 \leq \beta < N$  より

$$\begin{aligned} \min_{k \in \mathbf{Z}} |r \pm hg_2 + kN| &= \min\{|r - \beta|, |r + \beta - N|\} \\ &= |r - \min\{\beta, N - \beta\}| \end{aligned}$$

と表せる. 従って, 整数  $m$  に対して, その  $N$  による剰余  $\beta$  により,  $[m]_N = \min\{\beta, N - \beta\}$  なる関数を定義すると,

$$\begin{aligned} T(r) &= \min_{0 \leq h \leq [N/2]} \{R(h)R(r - [hg_2]_N)\} \\ &= \min\{R(r), \min_{1 \leq h \leq [N/2]} hR(r - [hg_2]_N)\} \end{aligned}$$

となる. 表  $T(r)$  は次のようにして計算できる. まず, 表  $T_0(r) = R(r)$  を作る. 以下,  $h = 1, 2, \dots, [N/2]$  の順に

$$T_h(r) = \min\{T_{h-1}(r), hR(r - [hg_2]_N)\} \quad (3)$$

なる表を作り,  $T(r) = T_{[N/2]}(r)$  とする. 関数  $R$  の定義に従って, 式 (3) の  $T_h(r)$  を

$$T_h(r) = \begin{cases} \min\{T_{h-1}(r), h(l-r)\} & , 0 \leq r < l \\ \min\{T_{h-1}(r), h\} & , r = l \\ \min\{T_{h-1}(r), h(r-l)\} & , l < r \leq [N/2] \end{cases}$$

$$(l = [hg_2]_N)$$

と 3 つの場合に分けて計算する.

## 6 3次元の BLP 表

今まで述べてきた方法を用いて, 3次元  $\mathbf{g} = (1, g_2, g_3)$  型の BLP を探索した結果を表す. CPU は Intel Celeron 600MHz, メモリは 128MB, HD は 20GB, プログラム言語は C 言語でコンパイラは gcc でバージョンは 2.95 として実行した. そこで 3次元問題の場合の  $N$  や  $g_i^{(N)}$  とそのときのメリット数  $\rho_3$  を参考文献 [1] で示されている  $N = 23644$  の続きからを表 1 で示す. そして表 1 を使って,  $p = 5$  のときの被積分関数

$$f(\mathbf{x}) = \int_{[0,1]^3} \exp(2 \cdot \sum_{i=1}^3 x_i) dx$$

を GLP 公式を用いてグラフにすると, 図 1 のようになる.

図 1 を見てみると, メリット数の値が大きいものほど, 誤差の精度が良くなるのが分かる. 分点総数を 10 倍にしたとき, ほぼ 5 桁信頼できる値になる. このことを式で表すと,

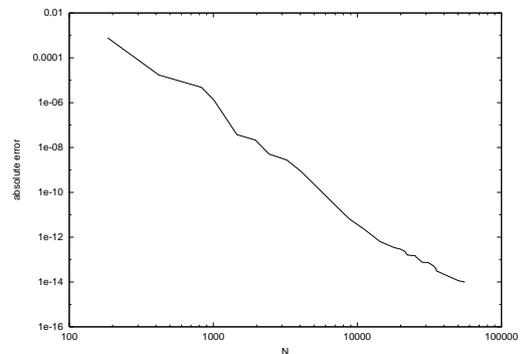
$$|I_N f - I f| = O(N^{-5})$$

となる.

表 1 3次元問題 ( $s = 3$ ) での BLP 表

$N$	$g_1$	$g_2$	$g_3$	$\rho_3$
25210	1	3500	5896	1674
25307	1	2018	10892	1677
25587	1	2231	8206	1680
25802	1	2386	8288	1712
25974	1	9578	11454	1800
⋮	⋮	⋮	⋮	⋮
55674	1	6071	26033	3520
55886	1	3876	10222	3540
56822	1	11870	15538	3552
56914	1	3758	25352	3612
57255	1	4052	14011	3840

図 1



## 7 おわりに

本研究をまとめると, 3次元 BLP の探索をメインに研究してきた. そして, BLP 表を作成することによって, 多次元数値積分の精度を良くする  $N$  や  $\mathbf{g}$  を選ぶことが容易になることが分かった. また, 3次元 BLP の探索実験では, 参考文献 [1] では  $N = 23644$  でメリット数 1666 までしか示されていなかったが, 今回  $N = 57255$  でメリット数 3840 まで探索することができたので, 3次元の BLP 表を更新することができた.

## 参考文献

- [1] 鳥居久訓, 杉浦洋: 3,4,5,6 次元 GLP の探索について日本応用数学会論文誌, Vol.3 No.3, pp.157-175 (1993).