



Title	モンテカルロ法による偏微分方程式解法の並列演算手法について
Author(s)	小山, 昭一; Koyama, Shoichi; 土肥, 俊 他
Citation	北海道大學工学部研究報告, 98, 41-49
Issue Date	1980-05-30
Doc URL	https://hdl.handle.net/2115/41616
Type	departmental bulletin paper
File Information	98_41-50.pdf



モンテカルロ法による偏微分方程式解法 の並列演算手法について

小山 昭一* 土肥 俊*
(昭和 54 年 12 月 27 日受理)

On a Parallel Computation Technique for the Monte Carlo Solutions of Partial Differential Equations

Shoichi KOYAMA and Shun DOI
(Received December 27, 1979)

Abstract

This paper is a study of a parallel computation technique for the Monte Carlo solutions of partial differential equations using MIMD multiprocessor systems.

We deal with the case in which the number of mesh-points for a Monte Carlo solution is larger than that of the processors used there. Since, in that case, the efficiency of the processor utilization and computing time are greatly affected by the manner in allocating those mesh-points to the processors; thus we paid special attention to the problem of allocation.

In the paper we proposed a technique by which to allocate the mesh-points to the processors in an attempt to obtain a high processor utilization and thus a short computing time. Effectiveness of the proposed method is evaluated in terms of the processor utilization by numerical tests.

1. はじめに

偏微分方程式の数値解を求めるには、一般に多くの計算時間を要する。一方、計算の高速化の一手法として多数のプロセッサを有する並列演算装置を用いる方法があり、これは最近の LSI 技術の発達に伴い有効な手段となりつつある。本論文は、このような並列演算装置を用いて偏微分方程式を高速に解く手法を確立することを目的とする。

本稿では、MIMD (Multiple Instruction stream Multiple Data stream) タイプの並列演算装置を用いた、モンテカルロ法による偏微分方程式の解法を検討する。モンテカルロ法による偏微分方程式の解法の特徴の 1 つは、任意の点の解を他の点の解とは独立に計算できることである。従って、演算の並列化が容易である。

Bhavsar と Kanetker³⁾ は SIMD (Single Instruction stream Multiple Data stream) タイプの並列演算装置でモンテカルロ法による解法を実行することを提案した。彼らの方法は、各格子点にプロセッサを 1 台ずつ割当て、酔歩を全格子点において同時並列に行なわせる。各格子点の試行はプロセッサ間で同期して行なわれ、1 回の試行は各格子点から始まる酔歩が全て境界に到

* 精密工学科 自動制御工学講座

達したときに終了し、次の新たな試行が始まる。しかしながら、各格子点の試行における酔歩のステップ数は格子点毎に異なり、かつランダムな値であるため、この方法では、プロセッサの遊び時間が生じ、高いプロセッサ利用効率が期待できない。

一方、MIMD タイプの並列演算装置では、各試行を同期して行なう必要がないので、プロセッサの遊び時間の減少を計ることが可能となる。本稿では、プロセッサの台数は与えられたものとし、格子点数がそれよりも多い一般的な場合を扱う。このときに生じる、格子点をプロセッサに割当てる割当ての問題を考察し、これを3節に述べる。この割当ての最適解を求めることは困難であり、多くの計算時間を要するので、ここではその準最適解を得る解法と、それに必要な平均ステップ数の推定方法を提案する。数値実験の結果から、格子点数がプロセッサの台数に較べ十分多いとき、高い計算効率が得られることが確認された。

境界値問題に対する数値解法の多くは、解の存在する領域の形状と境界条件が共に対称である場合に、計算時間を減少することができる。これは、領域の形状と境界条件とが共に対称な場合、境界値問題の解もまた対称となることから明らかである。しかしながら、特にモンテカルロ法においては、領域の形状が対称であるだけで、計算時間を減少できる。領域の形状が対称性を有する場合の解法を4節で述べる。

2. モンテカルロ法

本論に入る準備として、モンテカルロ法による偏微分方程式の解法について簡単にふれる。

2.1 基本的な解法

2次元2階の線形偏微分方程式

$$a \frac{\partial^2 u}{\partial x^2} + 2b \frac{\partial^2 u}{\partial x \partial y} + c \frac{\partial^2 u}{\partial y^2} + d \frac{\partial u}{\partial x} + e \frac{\partial u}{\partial y} + f = 0 \quad (1)$$

を、境界条件

$$u|_c = g(x, y) \quad (c \text{ は解の存在する領域の境界を表わす}) \quad (2)$$

のもとで解くことを考える。 a, b, c, d, e, f は x, y の関数であってもよい。(1) 式の差分近似は次のように表わされる。

$$u(P) = \sum_{i=1}^5 p_i(P) u(P_i) + \frac{h^2}{D(P)} f(P) \quad (3)$$

ここで、 $P = (x, y)$, $P_1 = (x+h, y)$, $P_2 = (x, y+h)$, $P_3 = (x-h, y)$, $P_4 = (x, y-h)$, $P_5 = (x+h, y+h)$ であり

$$p_1(P) = \frac{1}{D(P)} (a - 2b + hd), \quad p_2(P) = \frac{1}{D(P)} (c - 2b + he)$$

$$p_3(P) = \frac{a}{D(P)}, \quad p_4(P) = \frac{c}{D(P)}, \quad p_5(P) = \frac{2b}{D(P)}$$

$$D(P) = 2(a - b + c) + h(d + e)$$

今、1点 P_0 の解 $u(P_0)$ を求めるために、いわゆる酔歩をこの点から出発させる。格子点 P に来た酔歩は確率 $p_1(P)$, $p_2(P)$, $p_3(P)$, $p_4(P)$, $p_5(P)$ でそれぞれ、 P_1 , P_2 , P_3 , P_4 , P_5 に移るものとする。このようにして、酔歩は P_0 から出発して、格子点をランダム歩行する。そして、境界上の点 Q_i に来たとき、そこで止まるものとする。このようにしてできた径路の1つを R_i とし、和

$$Z_i = \sum_{P_j \in R_i} f(P_j) + g(Q_i) \quad (4)$$

を作る。ここで、 $\sum_{P_j \in R_i} g(P_j)$ は径路 R_i 上の全ての格子点 P_j についての和を意味する。ただし、この中には出発点 P_0 は含まれるが、終点 Q_i は含まれないものとする。 $g(Q_i)$ は境界点 Q_i における境界値を表わす ((2) 式)。上のような径路 R_i を N 個作り、そのおのおのについて Z_i を計算し、 Z_i の平均をとり、 $N \rightarrow \infty$ とすると

$$u(P_0) = \frac{1}{N} \sum_{i=1}^N Z_i \quad (N \rightarrow \infty) \quad (5)$$

すなわち、求める $u(P_0)$ になる。

2.2 適用範囲

このような酔歩によるモンテカルロ法においては、(3) 式の係数 $p_i(P)$ は、酔歩が点 P から次に遷移する点 P_i への遷移確率という意味を持っている。従って、この方法が適用できるための必要十分条件は、任意の点 P において

$$(1) \quad p_i(P) \geq 0, \quad i=1, 2, \dots$$

$$(2) \quad \sum_i p_i(P) = 1$$

となることである。線形偏微分方程式 (1) の場合、この方法が適用できるための必要十分条件は、 $D > 0$ とすると

$$\begin{aligned} a &\geq 0, & b &\geq 0, & c &\geq 0, \\ a - 2b + hd &\geq 0, & c - 2b + he &\geq 0, \\ 2(a - b + c) + h(d + e) &> 0 \end{aligned}$$

となることである。この条件を満たす代表的な方程式として、Laplace 方程式、Poisson 方程式、熱伝導方程式があげられる。

一方、 $d=e=0$ とすると、これらの条件より

$$b^2 - ac \leq 0$$

となり、従ってこの方法は波動方程式には適用できない。

2.3 試行のステップ数

N 回の試行によって、1 点 P_0 の解を得るために必要な計算時間 $T(P_0)$ は

$$T(P_0) = t\{\nu_1(P_0) + \dots + \nu_N(P_0)\} \quad (6)$$

で与えられる。ただし、 t は酔歩の 1 ステップに要する計算時間、 $\nu_i(P_0)$ は i 回目の酔歩におけるステップ数である。 $\nu_i(P_0)$ はランダムな値であり、これらの和は近似的に期待値の N 倍に等しい。すなわち、

$$T(P_0) \simeq tN\bar{\nu}(P_0) \quad (7)$$

ただし、

$$\bar{\nu}(P_0) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \nu_i(P_0) \quad (8)$$

ここで、 $\bar{\nu}(P_0)$ を解とする方程式を求める。ここでの結果は 3 節で用いられる。 $\bar{\nu}(P_0)$ は、(4)、(5) 式に従って

$$Z_i = \sum_{P_j \in R_i} 1 \quad (9)$$

$$\bar{\nu}(P) = \frac{1}{N} \sum_{i=1}^N Z_i \quad (N \rightarrow \infty)$$

と表わすことができる。(4)、(9) 式を比較すると、(3) 式に相当する式が、 $\bar{\nu}(P_0)$ の満たす差分方程式として得られる。すなわち、

$$\bar{v}(P) = \sum_{i=1}^5 p_i(P)u(P_i) + 1$$

格子間隔 $h=1$ とすると、結局、次の (10)、(11) 式が $\bar{v}(P_0)$ を近似的に解とする方程式と境界条件となる。

$$a \frac{\partial^2 \bar{v}}{\partial x^2} + 2b \frac{\partial^2 \bar{v}}{\partial x \partial y} + c \frac{\partial^2 \bar{v}}{\partial y^2} + d \frac{\partial \bar{v}}{\partial x} + e \frac{\partial \bar{v}}{\partial y} = -2\{(a-b+c) + (d+e)\} \quad (10)$$

$$\bar{v}|_c = 0 \quad (11)$$

例えば、原方程式 (1) が Laplace 方程式あるいは Poisson 方程式のとき、平均ステップ数の方程式 (10) は Poisson 方程式となる (後掲 (13)、(14)、(15) 式)。

3. 割当ての問題

モンテカルロ法では、任意の点の解が他の点の解と独立に計算される。従って、格子点数を n とすると、 n 個の独立なプログラムが得られる。今、プロセッサの台数を m とすると、一般に $m < n$ となり、 n 個の演算ブロックを m 台のプロセッサに割当てる割当ての問題が生じる。このとき、計算の効率、計算速度が割当て方法によって左右される。

ここで、割当て方法の平均効率 (割当ての効率) μ を次のように定義する。

$$\mu = \frac{T_1 + \dots + T_m}{m \cdot T_{\max}} \quad (12)$$

ただし、 $T_i (i=1, \dots, m)$ は i 番目のプロセッサ EP_i に割当てられた演算ブロック $\{E_j | j \in EP_i\}$ の計算時間であり、次のように表わされる。

$$T_i = tN \sum_{j \in EP_i} \bar{v}(P_j)$$

また、 $T_{\max} = \max\{T_1, \dots, T_m\}$ である。

従って、各格子点の平均ステップ数 $\bar{v}(P)$ が知られていれば、この割当ての問題は割当ての効率 μ を評価関数とする最適組合せを求める問題に帰着される。以下でこの解法について検討する。

3.1 割当てのアルゴリズム

上述の最適化問題を解くことは、一般にかなり困難であり、複雑な計算を要する。ここでは、演算ブロックをプロセッサに割当てるための前処理としてこの解を求めるわけなので、このために多くの計算時間を費やすことは好ましくない。そこで、 $\bar{v}(P_i)$ が既に求まったものとして、次のような準最適方法によって割当てを行なう。

Step 1 平均ステップ数 $\bar{v}(P_i) (i=1, \dots, n)$ を大きい順に並べる。これを改めて $\bar{v}_1, \dots, \bar{v}_n$ で表わす。

Step 2 $\tau_k = \bar{v}_k (k=1, \dots, m)$ とする。

Step 3 $\min\{\tau_1, \dots, \tau_m\}$ を求める。

Step 4 $\tau_l = \min\{\tau_1, \dots, \tau_m\}$ とすると、次に大きい平均ステップ数を τ_l に加え、これを新しく τ_l とする。ここで、Step 3 に戻り、全平均ステップ数 $\bar{v}_k (k=m+1, \dots, n)$ を割当てるまで繰り返す。

このとき、割当ての効率 μ は

$$\mu = \frac{\tau_1 + \dots + \tau_m}{m \cdot \tau_{\max}}$$

で与えられる。ただし、 $\tau_{\max} = \max\{\tau_1, \dots, \tau_m\}$ である。

3.2 平均ステップ数の推定

上述の割当ての計算を行なうために、 $\bar{v}(P)$ を求めなければならない。 $\bar{v}(P)$ は (10), (11) 式の解であり、一般に厳密解は解析的には求まらない。また、前述のような理由から、精度の高い解を求めることは必ずしも得策でない。そこで、次のようにして解の推定を行なう。

2次元の Laplace 方程式及び Poisson 方程式

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \quad (13)$$

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y) \quad (14)$$

の Dirichlet 問題に対する平均ステップ数の方程式と境界条件は

$$\frac{\partial^2 \bar{v}}{\partial x^2} + \frac{\partial^2 \bar{v}}{\partial y^2} = -4 \quad (15)$$

$$\bar{v}|_c = 0 \quad (16)$$

となる。

(i) 円領域の場合 解の存在する領域の形状が円 ($0 \leq r \leq R$, $0 \leq \theta \leq 2\pi$) の場合、平均ステップ数は解析的に求まり、

$$\bar{v}(r) = R^2 - r^2 \quad (17)$$

となる。

(ii) 矩形領域の場合 領域の形状が矩形 ($0 \leq x \leq i_0$, $0 \leq y \leq j_0$) の場合、領域内の点 $(x, y) = (i, j)$ における平均ステップ数 \bar{v}_{ij} を

$$\bar{v}_{ij} = \min\{i(i_0 - i), j(j_0 - j)\} \quad (18)$$

で推定する。

(iii) 矩形の和で表わされる領域の場合 この場合、平均ステップ数を次のようにして推定する。

Stage 1 領域を矩形の部分領域 D_1, \dots, D_L に分解する。

Stage 2 部分領域 D_l が $0 \leq x \leq X$, $0 \leq y \leq Y$ で表わされるとする。点 $P = (x, y)$ が D_l に属するとき、

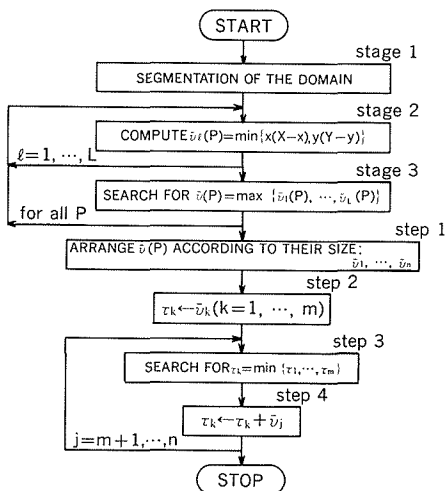


Fig. 1 The algorithm to allocate n mesh-points to m processors for domains composed of rectangles

$$\bar{v}_l(P) = \min\{x(X-x), y(Y-y)\} \quad (19)$$

とし、属さないとき $\bar{v}_l(P) = 0$ とする。

Stage 2 点 P の推定値を

$$\bar{v}(P) = \max\{\bar{v}_1(P), \dots, \bar{v}_L(P)\} \quad (20)$$

とする。

この推定法は自由境界条件を区分的に含む問題や3次元以上の問題に容易に拡張できる。また、熱伝導方程式に対する(18)式のような推定関数を得ることもさほど難しいことではない。

平均ステップ数の推定 (iii) を含めた割当てのフローチャートを Fig. 1 に示す。

3.3 数値実験

プロセッサの台数 $m=32$ としたときの正方形領域及び矩形の和で表わされる領域 (Fig. 2) における割当ての効率を Fig. 3, 4 に示す。図の横軸は格子点数 n 、縦軸は割当ての効率 μ を表わす。Fig. 3 で、破線は平均ステップ数の方程式 (15) を境界条件 (16) のもとで解いてその値をもとに割当てを行ったときの効率であり、実線は推定値を用いて割当てを行ったとき得られた効率である。図の破線から、3.1 で述べた割当て方法は、格子点数がプロセッサ台数のほぼ2倍以上のとき、良い結果を与える。特に、格子点数がプロセッサの台数に較べ十分多いとき、効率はほぼ1となるのがわかる。一方、図の破線と実線の比較から、3.2 (ii) の推定方法が十分有効なものであることが立証できる。Fig. 4 は Fig. 2 の3種類の領域形状における割当ての効率を Fig. 3 の正方形領域の場合と比較したものである。ここでは、3.2 (iii) の推定方法は有効であると結論できる。

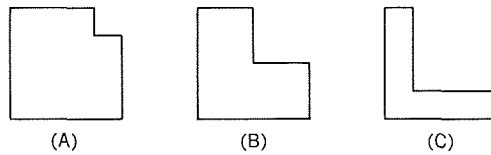


Fig. 2 3 types of domain composed of rectangles
(Used in Fig. 4)

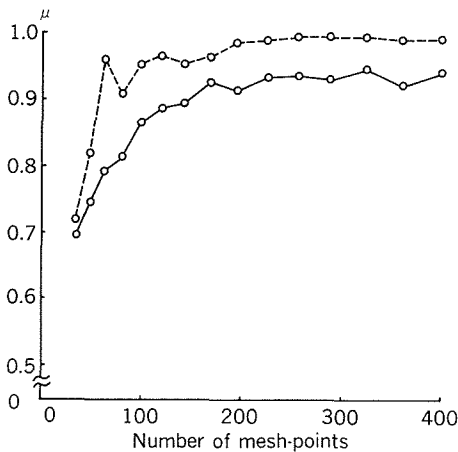


Fig. 3 Efficiency of the proposed method for square domains

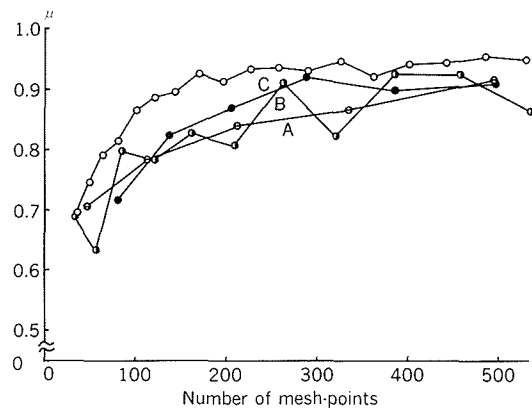


Fig. 4 Efficiency of the proposed method for domains composed of rectangles shown in Fig. 2

4. 境界形状の対称性を利用した解法

この節では、Laplace 方程式の Dirichlet 問題を考える。モンテカルロ法を用いると、解 $u(P)$ は

$$u(P) = \frac{1}{N} \sum_{i=1}^N Z_i \quad (21)$$

$$Z_i = g(Q_i) \quad (22)$$

で与えられる。ただし、 $g(Q_i)$ は i 回目の試行で酔歩が到達した境界点 Q_i における境界値を表わす。さて、点 P を出発した酔歩が境界点 Q に到達する確率を $u(P, Q)$ で表わし、点 P を出発した酔歩が、 N 回試行後境界点 Q_j に到達した回数を L_j とすると

$$\lim_{N \rightarrow \infty} \frac{L_j}{N} = u(P, Q_j) \quad (23)$$

となる。従って、解 $u(P_i)$ は

$$u(P_i) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=1}^N Z_k = \lim_{N \rightarrow \infty} \sum_{l=1}^s \frac{L_l}{N} g(Q_l) = \sum_{l=1}^s g(Q_l) u(P_i, Q_l) \quad (24)$$

ここで、 $P_i (i=1, \dots, r)$ 、 $Q_j (j=1, \dots, s)$ は、それぞれ解の存在する領域 D 内の網目点及び D の網目境界点を表わす。(24) 式を、行列を用いて表わすと

$$\mathbf{u} = \mathbf{U} \mathbf{g} \quad (35)$$

ただし、

$$\mathbf{U} = \begin{pmatrix} u(P_1, Q_1) & \cdots & u(P_1, Q_s) \\ \vdots & & \vdots \\ u(P_r, Q_1) & \cdots & u(P_r, Q_s) \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} u(P_1) \\ \vdots \\ u(P_r) \end{pmatrix}, \quad \mathbf{g} = \begin{pmatrix} g(Q_1) \\ \vdots \\ g(Q_s) \end{pmatrix}$$

次に、 $0 \leq x \leq 2n+1$ 、 $0 \leq y \leq 2n+1$ で表わされる正方形領域を考える。ただし、格子間隔 $h=1$ とする。Fig. 5 のように、領域を 8 個の対称な部分 (D_0, \dots, D_7) に分割する。 D_0 内の $n(n-1)/2$ 個の格子点を適当に $P_{01}, \dots, P_{0n(n-1)/2}$ で表わす。 D_1 と共有する n 個の格子点を R_{01}, \dots, R_{0n} で表わし、 n 個の境界点を Q_{01}, \dots, Q_{0n} で表わす。同様に、 D_k 内の対応する点をそれぞれ $P_{k1}, \dots, P_{kn(n-1)/2}$ 、 R_{k1}, \dots, R_{kn} 、 Q_{k1}, \dots, Q_{kn} で表わす。こうすると、点 R_{0i} と R_{1i} 、 R_{2i} と R_{3i} 、 R_{4i} と R_{5i} 、 R_{6i} と R_{7i} は、それぞれ同一の点を表わす。この重複を認めると、(25) 式は次のように表わすことができる。

$$\begin{pmatrix} \mathbf{u}_0 \\ \vdots \\ \mathbf{u}_7 \end{pmatrix} = \begin{pmatrix} U_{00} & \cdots & U_{07} \\ \vdots & & \vdots \\ U_{70} & \cdots & U_{77} \end{pmatrix} \begin{pmatrix} \mathbf{g}_0 \\ \vdots \\ \mathbf{g}_7 \end{pmatrix} \quad (26)$$

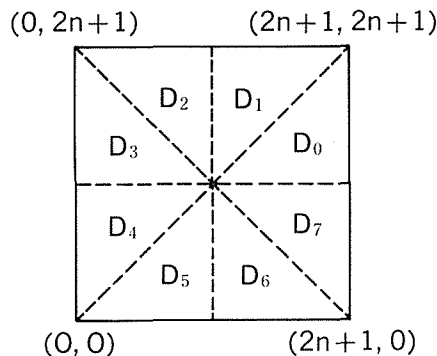


Fig. 5 Partition of the square domain into 8 symmetrical parts

ただし

$$U_{kl} = \begin{pmatrix} u(P_{k_1}, Q_{l_1}) & \cdots & u(P_{k_1}, Q_{l_n}) \\ \vdots & & \vdots \\ u(P_{k_{n(n-1)/2}}, Q_{l_1}) & \cdots & u(P_{k_{n(n-1)/2}}, Q_{l_n}) \\ \vdots & & \vdots \\ u(R_{k_1}, Q_{l_1}) & \cdots & u(R_{k_1}, Q_{l_n}) \\ \vdots & & \vdots \\ u(R_{k_n}, Q_{l_1}) & \cdots & u(R_{k_n}, Q_{l_n}) \end{pmatrix}, \quad \mathbf{u}_k = \begin{pmatrix} u(P_{k_1}) \\ \vdots \\ u(P_{k_{n(n-1)/2}}) \\ \vdots \\ u(R_{k_1}) \\ \vdots \\ u(R_{k_n}) \end{pmatrix}, \quad \mathbf{g}_l = \begin{pmatrix} g(Q_{l_1}) \\ \vdots \\ g(Q_{l_n}) \end{pmatrix}$$

領域の対称性から、確率 $u(P, Q)$ は次の関係を満たす。

$$u(P_{k_i}, Q_{k+m_j}) = u(P_{k+1_i}, Q_{k+1-m_j}), \quad u(R_{k_i}, Q_{k+m_j}) = u(R_{k+1_i}, Q_{k+1-m_j})$$

従って

$$U_{0m} = U_{11-m} = U_{22+m} = \cdots = U_{77-m} \quad (27)$$

ここで、添字 $k+1$, $k+m$, $k+1-m$, m , $1-m$, $2+m$, ..., $7-m$ は 8 のモジュールをとるものとする。簡単のために、 U_{00}, \dots, U_{07} を U_0, \dots, U_7 で表わし、(27) 式を用いると、(26) 式は次のように書き換えられる。

$$\begin{pmatrix} \mathbf{u}_0 \\ \vdots \\ \vdots \\ \vdots \\ \mathbf{u}_7 \end{pmatrix} = \begin{pmatrix} U_0 & U_1 & U_2 & U_3 & U_4 & U_5 & U_6 & U_7 \\ U_1 & U_0 & U_7 & U_6 & U_5 & U_4 & U_3 & U_2 \\ U_6 & U_7 & U_0 & U_1 & U_2 & U_3 & U_4 & U_5 \\ U_3 & U_2 & U_1 & U_0 & U_7 & U_6 & U_5 & U_4 \\ U_4 & U_5 & U_6 & U_7 & U_0 & U_1 & U_2 & U_3 \\ U_5 & U_4 & U_3 & U_2 & U_1 & U_0 & U_7 & U_6 \\ U_2 & U_3 & U_4 & U_5 & U_6 & U_7 & U_0 & U_1 \\ U_7 & U_6 & U_5 & U_4 & U_3 & U_2 & U_1 & U_0 \end{pmatrix} \begin{pmatrix} \mathbf{g}_0 \\ \vdots \\ \vdots \\ \vdots \\ \mathbf{g}_7 \end{pmatrix} \quad (28)$$

従って、酔歩により、 $U = [U_0 \dots U_7]$ を求めれば、行列 U の残りの要素は容易に求まる。

さて、格子点 R_{2k_i} と R_{2k+1_i} ($k=1, 2, 3, i=1, \dots, n$) は同一の点であるから、上の方法によって求めた

$$u(P_{2k_i}) = \sum_{l=0}^7 \sum_{j=1}^n g(Q_{l_j}) u(P_{2k_i}, Q_{l_j}) \quad (29)$$

$$u(P_{2k+1_i}) = \sum_{l=0}^7 \sum_{j=1}^n g(Q_{l_j}) u(P_{2k+1_i}, Q_{l_j}) \quad (30)$$

は同一の解を与えることになる。そこで、(29), (30) 式の平均をその点の解とするならば、これは $2N$ 回の試行を行ったことに相当する。逆に、必要な試行回数が N ならば、これらの点については試行回数を予め $N/2$ に設定してよい。

このようなことから、格子点 P_{0i} ($i=1, \dots, n(n-1)/2$) の試行回数を各々 N 回、 R_{0i} ($i=1, \dots, n$) の試行回数を $N/2$ 回とすると、全試行回数は

$$\frac{n(n-1)}{2} \times N + n \times \frac{N}{2} = \frac{n^2 N}{2}$$

となる。一方、この方法を用いないならば、全格子点から酔歩を出発させなければならないから、全試行回数は $4n^2 N$ となる。すなわち、境界形状の対称性を利用した解法を用いることによって、全試行回数を $1/8$ に減らすことができる。

D_0 内の $n(n+1)/2$ 個の代表格子点の解はそれぞれ独立に求まる。従って、3 節で述べた方法で並列演算を行なうことができる。このとき、推定、割当ては 3.2 (ii), 3.1 の方法を用いればよい。

ここでは、境界形状の対称性を利用した解法を、正方形領域の場合を例にとって説明した。こ

の方法は境界形状が対称性を有する全ての場合に用いられる。また、Poisson 方程式及び熱伝導方程式の第1種境界値問題に適用することが可能である。

5. おわりに

本論文では、MIMD タイプの並列演算装置を用いてモンテカルロ法により偏微分方程式を解く手法を提案した。

プロセッサの台数は与えられたものとし、格子点数がそれより多い一般的な場合を扱った、この場合、割当てに任意性があり、従って何らかの最適な割当てを求める問題が生じる。この割当て問題解決の手法として、最適化問題の準最適な解法とそのための平均ステップ数の推定方法を提案した。数値実験の結果、提案された手法は有効であることが示された。

多くの数値解法では、領域の形状と境界条件が共に対称な場合に計算時間が減少されるのに対して、モンテカルロ法では、「境界形状の対称性を利用した解法」によって、領域の形状のみが対称な場合でも計算時間を減少できる。

謝 辞

本学工学部在職中、本研究及び種々にわたりお世話下さいました旭川工業高等専門学校長三浦良一先生に深謝の意を表します。

参 考 文 献

- 1) 宮武 修・中山 隆：モンテカルロ法，(1969)，日刊工業新聞社。
- 2) Shreider, A. Y.: The Monte Carlo Method, (1966), p. 35-55, Pergamon.
- 3) Bhavsar, V. C. and Kanetker, V. V.: Advances in Computer Methods for Partial Differential Equations-II, (1977), p. 205-213.