数理解析研究所講究録
第 880 巻 1994 年 86-95

京大 数理研 研究会「数値計算アルゴリズムの現状と展望」 '93.10.26

差分スキームの再考によるベクトル計算機向き 不完全 LU 分解について

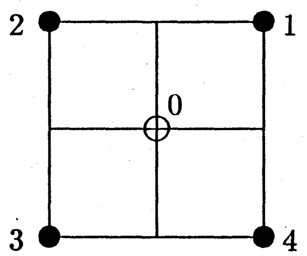
藤野 清次 (計算流体力学研究所) (Seiji Fujino)
竹内 敏己 (花王) (Toshimi Takeuchi)

1 はじめに

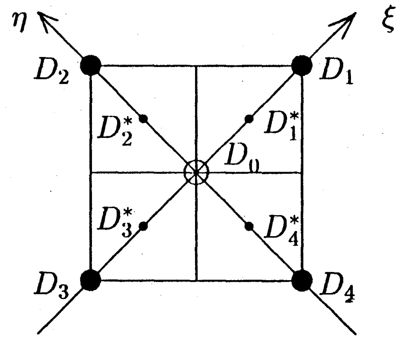
本研究では、変形 5 点差分 [1] による不完全 LU 分解のベクトル計算機上の効率評価および精度の検証をおこなう。取り上げる問題は、拡散係数が場所ごとに違うポアソン方程式とし、この方程式を (M)ICCG 法で解く。また、変形差分の導出は、座標系の回転を利用して行なう。

2 座標系の回転による変形差分の導出と離散化

解くべきポアソン方程式を $\nabla(D(x,y)\nabla u(x,y)) = -f$ とする。ここで、拡散係数 $D(x,y)$ は場所の関数とし、差分格子の縦横の比はすべて等しい (=h) とする。各格子内の斜め方向に位置する 4 つの格子点を使った変形 5 点差分格子を Fig.1(a) に示す。



(a)A variant of difference scheme



(b)ξ-η axes

Fig. 1. A variant of 5-point difference scheme and ξ-η axes

このとき、拡散係数 D の ξ 方向および η 方向の平均値をそれぞれ $D_1^* = \frac{1}{2}(D_0 + D_1)$, $D_2^* = \frac{1}{2}(D_0+D_2)$, $D_3^* = \frac{1}{2}(D_0+D_3)$, $D_4^* = \frac{1}{2}(D_0+D_4)$ とおく (Fig.1(b) 参照)。このとき、

$$
\begin{aligned}
D_1^* u_\xi &= D_1^*(u(i+1,j+1) - u(i,j))/\sqrt{2}h, \\
D_2^* u_\eta &= D_2^*(u(i-1,j+1) - u(i,j))/\sqrt{2}h
\end{aligned}
$$

と近似できる。また、演算子$\Delta = \nabla(\nabla u)$は回転に対し不変であるので[4]、$x$-$y$座標系での方程式$\nabla(D(x,y)\nabla u(x,y))$は、$\xi$-$\eta$座標系の$\nabla(D(\xi,\eta)\nabla u(\xi,\eta))$と等しい。したがって、

$$\begin{aligned}\nabla(D(x,y)\nabla u(x,y)) &= \frac{1}{\sqrt{2}h}\{D_1^*\frac{u(i+1,j+1)-u(i,j)}{\sqrt{2}h} - D_3^*\frac{u(i,j)-u(i-1,j-1)}{\sqrt{2}h}\} \\ &+ \frac{1}{\sqrt{2}h}\{D_2^*\frac{u(i-1,j+1)-u(i,j)}{\sqrt{2}h} - D_4^*\frac{u(i,j)-u(i+1,j-1)}{\sqrt{2}h}\}\end{aligned}$$

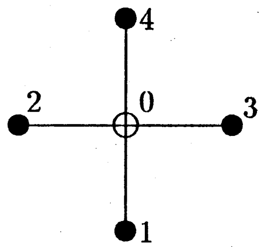
のように、斜め方向の4つの格子点を使った変形差分で方程式は離散化できる。

3 ベクトル化

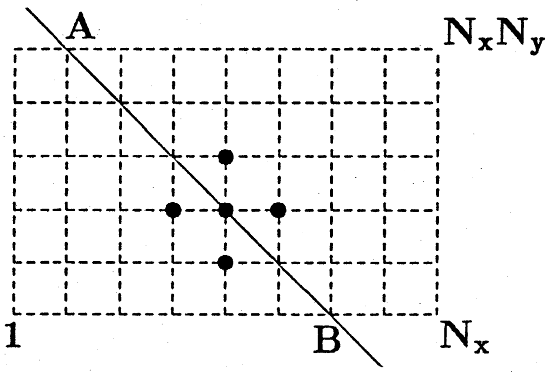
以下では、格子点の順番付けは辞書式(自然な順番付け)で行なうものとし、差分格子は縦横の格子幅が等しくかつ一様な等方性格子に限るものとする。

3.1 通常の5点差分と超平面法

Fig.2(a)に通常の5点差分で用いられる5個の格子点を示す。またFig.2(b)に不完全分解のベクトル化で用いられる超平面の一つの例を示す。



(a)Standard 5-point difference scheme



(b)An example of hyperplane

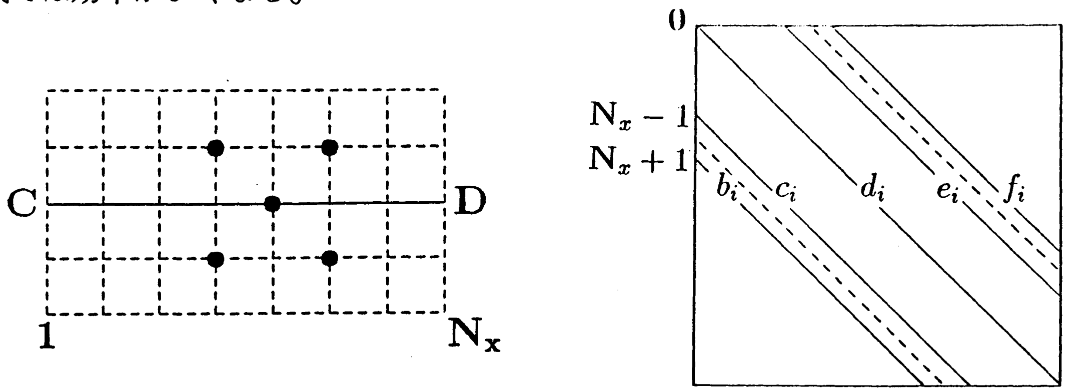
Fig. 2. Standard 5-point difference scheme and its vectorization

ベクトル計算機では、Fig.2(b)中の斜め線(点Aと点Bを結んだ線)上の点に対して、前進/後退代入の計算を同時に行なうことができる。しかし、水平方向の格子点数をN_xとすると、超平面上の格子点番号の間隔は$N_x - 1$になり、メモリへ連続的にアクセスできない。そのため、計算速度はその最高演算速度に比べてかなり低い。また、平均ベクトル長もオーダ$O(\frac{N_x}{2})$と短くその大きさも一定でないため、この方法でベクトル計算機の高速性を十分に引き出すのは難しい。

Fig.2(b)からわかるように、点Cと点Dを結んだ水平線上の格子点を同時に更新できれば、メモリへ連続的にアクセスでき効率向上が期待できる。

3.2 変形 5 点差分のベクトル化

一方、Fig.1(a) に示した変形 5 点差分の場合には、Fig.3(a) に示すように格子点 (i_0, j_0) が位置する C と点 D を結んだ水平線上の格子点ごとにベクトル化ができ、ベクトル計算機では効率がよくなる。



(a)Vectorization　　(b)Nonzero elements of matrix A

Fig. 3. Vectorization of a variant of difference scheme and nonzero elements

4 行列要素の対応

ここでは、元の係数行列 A の要素と変形 5 点差分を使ったときの不完全 LDU 分解後の行列要素との対応について記述する。いま x 方向の格子点数を N_xとする。元の係数行列 A の対角要素を d_iとし, 対角要素の下 $N_x - 1, N_x + 1$ だけ離れた要素をそれぞれ c_i, b_i とする。同様に、対角要素の上 $N_x - 1, N_x + 1$ だけ離れた要素をそれぞれ e_i, f_iとする。Fig.3(b) に行列 A の非零要素を示す。不完全 LDU 分解における非零要素は元の行列 A の非零要素と同じ位置のものだけとする。このとき、下 (上) 三角行列 L(U) の非零要素は元の行列 A の対応する要素と等しくなる。 また、対角行列 D の要素 dd_iは、

$$dd_i^{-1} = d_i - b_i f_{i-N_x-1} d_{i-N_x-1} - c_i e_{i-N_x+1} d_{i-N_x+1} \tag{4.1}$$

で表される。Gustafsson 流 [3],[5] の修正不完全 LDU 分解をしたときの対角項 dd_iは、

$$\begin{aligned} dd_i^{-1} &= d_i - b_i f_{i-N_x-1} d_{i-N_x-1} - c_i e_{i-N_x+1} d_{i-N_x+1} \\ &\quad -\alpha(b_i e_{i-N_x-1} d_{i-N_x-1} + c_i f_{i-N_x+1} d_{i-N_x+1}) \end{aligned} \tag{4.2}$$

で表される。ここでαはパラメータ $(0 \le \alpha \le 1.0)$ である。

5. 拡散係数場所依存の問題に対する収束性

以下の数値実験は、ベクトル計算機 VP-200 を使い、計算はすべて倍精度演算で行なった。そして、単位正方形領域で定義されたポアソン方程式 $\nabla(\mathrm{D}(x,y)\nabla u(x,y)) = -f$を(M)ICCG 法で解き、拡散係数をいろいろ変えてその収束性を調べた。

5.1 拡散係数の分布が連続のとき

ここでは, 拡散係数 $\mathrm{D}(x,y)$ が (i)$\mathrm{D}_a(x,y) = 0.01+x^2+y^2$ と (ii)$\mathrm{D}_b(x,y) = 20e^{3.5(x^2-y^2)}$ について収束性を調べた。ポアソン方程式の右辺項は解がすべて 1.0 となるように定める。Fig.4 に拡散係数 $\mathrm{D}_a(x,y)$ と $\mathrm{D}_b(x,y)$ の分布の透視図を示す。

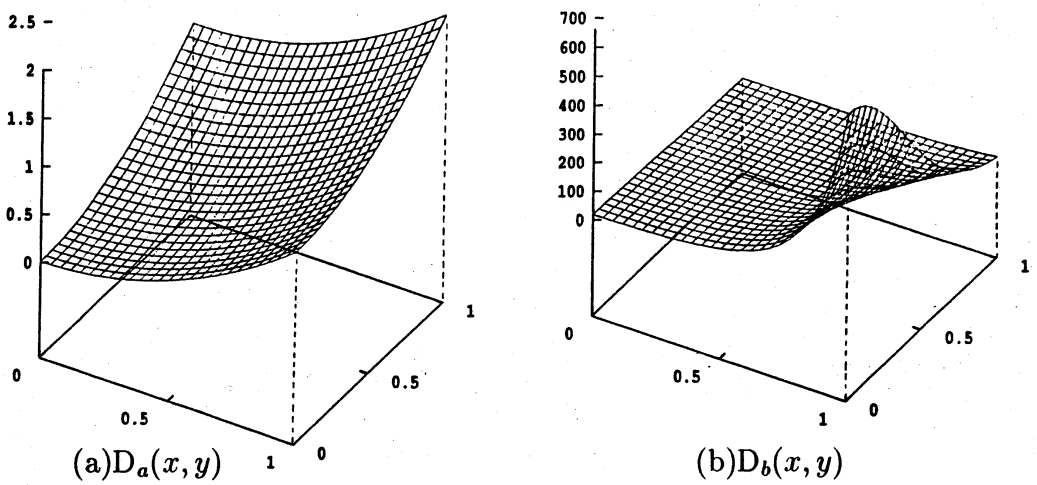


Fig.4. Perspective view of diffusion parameters $\mathrm{D}_a(x,y)$ and $\mathrm{D}_b(x,y)$

Table 1 にポアソン方程式を ICCG 法で解いたときの反復回数, CPU 時間, 最大誤差, L_2ノルム誤差を示す。括弧内は MICCG 法の結果である。(M)ICCG 法の収束条件は相対残差 L_2ノルムで 10^{-12}とし、格子点数は 250×250 とした。通常の 5 点差分に対する (M)IC 分解のベクトル化は超平面法で行なった。メモリ衝突の影響ができるだけ少ないように、x 方向の格子点数を 250 にとって実験を行なった (このとき、メモリへのアクセス間隔は奇数: 249 になる [2])。この表から、変形 5 点差分の方が、通常の 5 点差分より、CPU 時間が半分 [(a) のとき 52%, 55%, (b) のとき 48%, 49%] で済み、誤差も少ないことがわかる。

Table 1. Numerical results for diffusion parameters $\mathrm{D}_a(x,y)$ and $\mathrm{D}_b(x,y)$

$\mathrm{D}_a(x,y)$	通常 5 点	変形 5 点
反復回数	270 (134)	218 (114)
CPU [sec]	2.47 (1.22)	1.28 (0.67)
Max	1.67×10^{-11}	1.11×10^{-11}
L_2 norm	3.94×10^{-12}	3.07×10^{-12}

$\mathrm{D}_b(x,y)$	通常 5 点	変形 5 点
反復回数	278 (144)	209 (110)
CPU [sec]	2.54 (1.32)	1.23 (0.65)
Max	8.81×10^{-11}	3.60×10^{-11}
L_2 norm	2.69×10^{-11}	1.04×10^{-11}

Fig.5 は、拡散係数が $D_b(x,y)$ のとき、MICCG 法における MIC 分解のパラメータ α の値を変化させて、収束までの反復回数を調べたものである。変形差分のときも通常の差分のときと同様に、パラメータαの値を 1 に近づける程反復回数が単調に減少したことがわかる。α=0.95 のとき、反復回数で 144 回が 110 回 (76%) に、CPU 時間で 1.32 秒が 0.65 秒 (49%) まで減少し、変形差分の方が効率がよいことがわかる。

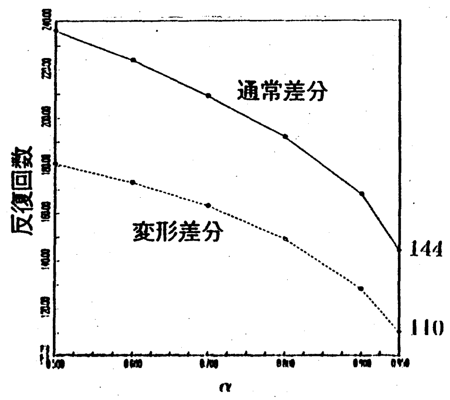


Fig. 5. Convergence property of MICCG method for parameter α

5.2 拡散係数の分布が不連続のとき

ここでは、拡散係数 $D(x,y)$ が不連続な分布で与えられるポアソン方程式を (M)ICCG 法で解き収束性を調べた。ポアソン方程式の右辺項は解がすべて 1.0 になるように定め、格子点数や (M)ICCG 法の収束条件などの計算条件は前節と同じとした。

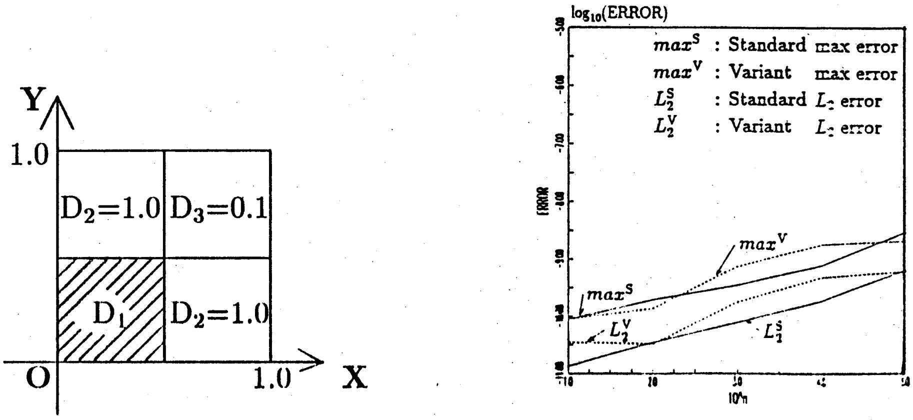
5.2.1 問題 1: 水平と垂直方向の境界が混在するとき

Fig.6(a) に問題 1 の拡散係数 D_i $(i=1,2,3)$ の分布を示す。ここでは、拡散係数 D_2, D_3の値は固定し、D_1のみ値を変化させ (M)ICCG 法の収束性を調べた。Table 2 に ICCG 法で解いたときの反復回数と CPU 時間を示す。この表から、変形 5 点差分を使うと、通常の 5 点差分のときと比べて、反復回数は 85% ~95%に、CPU 時間は 55% ~61%に減少することがわかる。

Table 2. Iterations and CPU time of the ICCG method

5点		D_1				
差分		10^0	10^1	10^2	10^3	10^4
通常	反復回数	254	253	249	244	238
変形		**241**	**215**	**223**	**218**	**215**
通常	CPU [sec]	2.32	2.32	2.28	2.23	2.20
変形		**1.42**	**1.27**	**1.31**	**1.28**	**1.26**

Fig.6(b) に、通常の 5 点差分と変形 5 点差分をそれぞれ使用したときの誤差の拡散係数 D_1依存性を示す。ここで、横軸は D_1の値, 縦軸は誤差で、いずれも対数目盛である。この図から、二つの差分スキームの誤差は同程度の大きさであることがわかる。



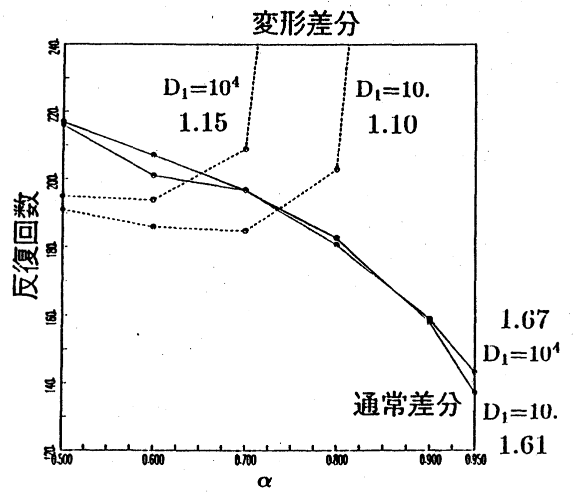
(a)$D_i(i=1,2,3)$　　　　(b)Convergence property

Fig. 6. Distribution of D_i and dependency of convergence for diffusion coefficient D_1

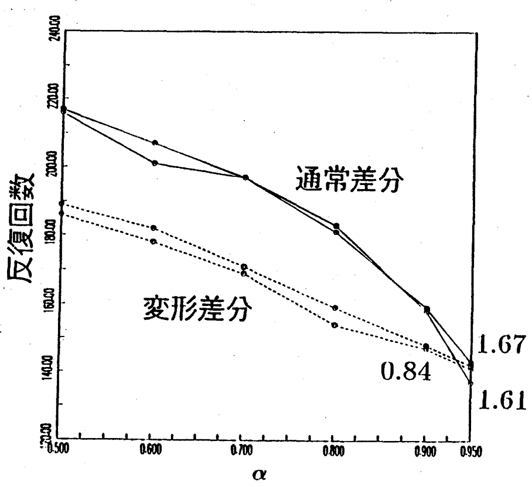
一方、MICCG 法でポアソン方程式を解いた場合の収束性を Fig.7(a),(b),(c) に示す。Fig.7(a) は、MIC 分解のパラメータαを変化させたときの収束までの反復回数をプロットしたものである。通常の 5 点差分では、MIC 分解のパラメータ αの値は解析領域全体で一定値である。一方、変形差分の場合には、すべての格子点でαの値を 0.7 以上にしたところ反復回数が急激に増加し収束性が悪化したことがわかる。

そこで、パラメータαの設定を 2 つの場合に分けて行なった。まず、(i) 境界の近傍の格子点ではαの値を 0.0 とおいた。即ち、垂直方向の境界の指標を (i_m, j) とし、水平方向の境界の指標を (i, j_n) すると、指標 i_m-1, i_m, i_m+1 および指標 j_n-1, j_n, j_n+1 となる格子点のαは 0.0 とした。(ii) 領域内のその他の格子点については$\alpha=0.95$ にして収束性を調べた。このときの結果を Fig.7(b) に示す。このように境界の形に合わせてαの値を設定すれば、パラメータα が 1 に近づいても収束性は落ちないことがわかる。

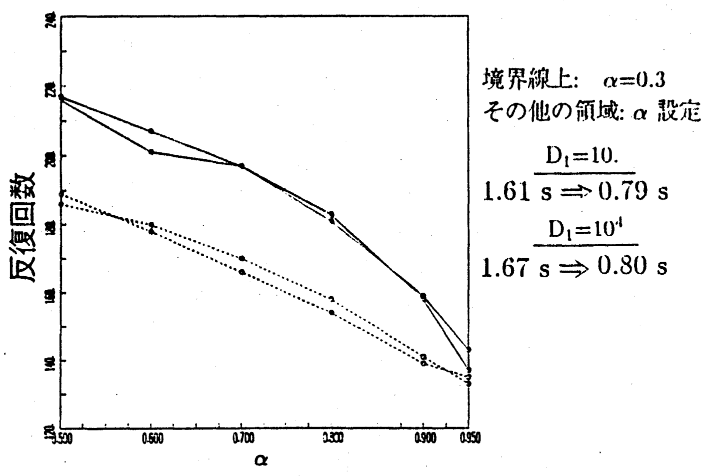
次に、境界近傍の格子点に対してα=0.3 としたときの結果を Fig.7(c) に示す。このようにαを境界に合わせて決めれば、変形差分を使ったときの CPU 時間は、通常の差分を使ったときに比べて、拡散係数 $D_1=10.$ のとき 1.61 秒が 0.79 秒 (49%) に、同じく $D_1=10^4$ のときは 1.67 秒が 0.80 秒 (48%) に減少し、変形差分の有用性がわかる。次に、このような境界に対する収束不安定性の原因を探るため、垂直方向の境界のみある場合と、水平方向の境界のみある場合、の二つの場合に分けて MIC 分解の収束性を調べた。



(a)Constant α=0.95 on whole region



(b)α=0.0 on boundaries only

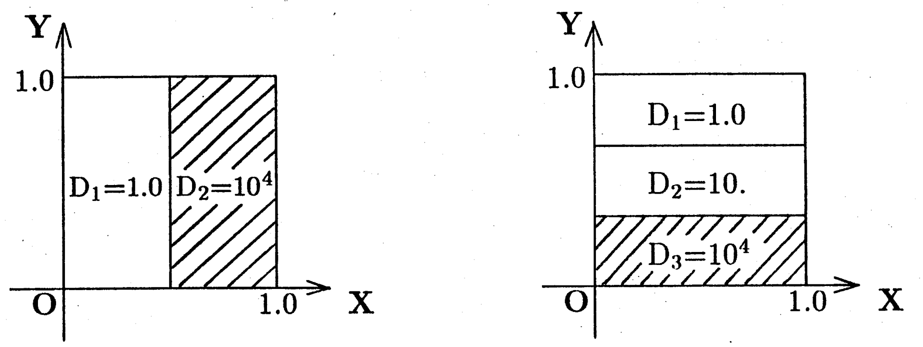


(c)α=0.3 on boundaries only

Fig. 7. Convergence property of MICCG method using a variant of difference scheme

5.2.2 問題 2: 水平あるいは垂直方向のみの境界

Fig.8(A) に問題 2 の拡散係数 (A)$D_4(x,y)$ の分布を示す。このときの MICCG 法の収束性を Fig.9(A) に示す。横軸はパラメータαの値、縦軸は収束までの反復回数を示している。変形差分を使ったとき、パラメータαの値が 1 に近づくと、MICCG 法の反復回数が急激に増えることがわかる。

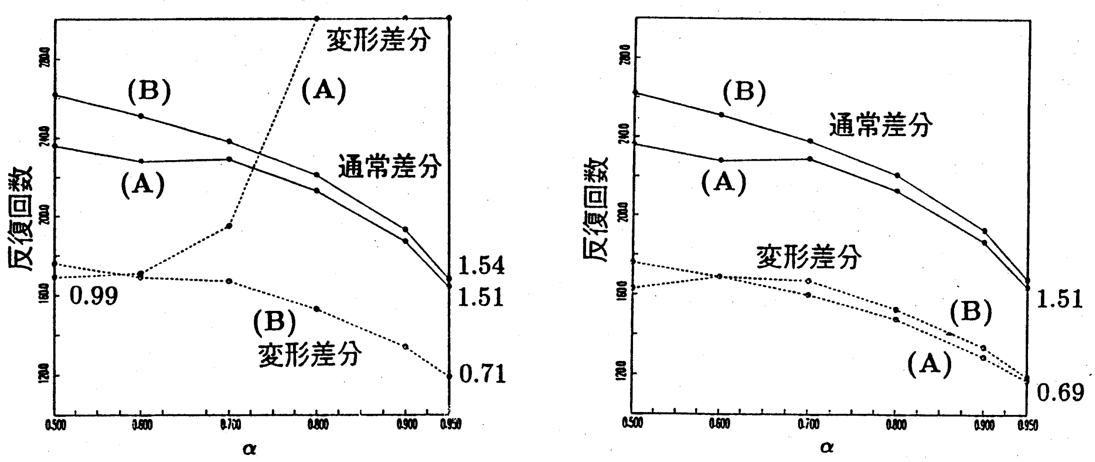


(A)$D_4(x,y)$　　(B)$D_5(x,y)$

Fig. 8. Distribution of diffusion coefficient $D_4(x,y)$ and $D_5(x,y)$

一方、拡散係数 (B)$D_5(x,y)$ のとき (Fig.8(B) 参照) は、Fig.9(A) からわかるように、通常の 5 点差分と同様の収束性を示した。Fig.9 中の数字は CPU 時間である。

そこで、(A)$D_4(x,y)$ に対して、**問題 1** と同様に MIC 分解において、境界近傍の格子点に対してα=0.3、その他の格子点に対してα=0.95 を与えた。その結果を Fig.9(B) に示す。この図から、CPU 時間は (A) のケースでは 1.54 秒が 0.71 秒 (46%) に、そして (B) のケースでは 1.51 秒が 0.69 秒 (46%) に短縮され、変形差分の方が MICCG 法でも効率がよいことがわかる。



(A) 全領域 α: 一定　　(B) 変形差分: 境界上α=0.3

Fig. 9. Convergence property for diffusion coefficient $D_4(x,y)$ and $D_5(x,y)$

この結果、この変形差分は垂直方向の境界で拡散係数が大きく変わるとき、MIC 分解の収束性に悪影響を及ぼすが、パラメータαを境界の形状に合わせて決めてやれば、通常の差分を使うより効率的であることがわかった。

5.2.3 問題 3: 境界が斜め方向にあるとき

斜め方向の境界で拡散係数の値が大きく違う拡散係数 $D_6(x,y)$ と $D_7(x,y)$ について、(M)ICCG 法の収束性を調べた。分布を Fig.10(a),(b) に示す。

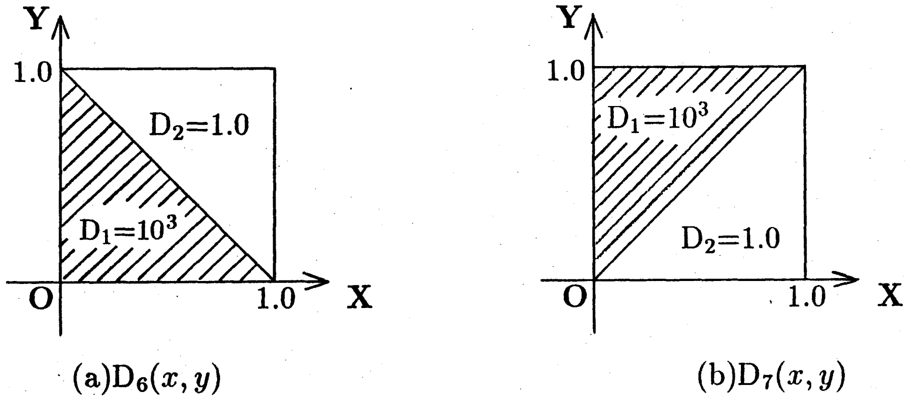


Fig. 10. Distribution of diffusion coefficient $D_6(x,y)$ and $D_7(x,y)$

Table 3 に、拡散係数が $D_6(x,y)$ および $D_7(x,y)$ のポアソン方程式を (M)ICCG 法で解いたときの実験結果を示す。括弧内の数字が MICCG 法の結果である。MICCG 法の MIC 分解のパラメータαは、通常の 5 点差分のときは $\alpha = 0.95$ に固定した。一方、変形 5 点差分のときは、境界近傍の格子点に対しては $\alpha = 0.5$ を与え、領域内のこれ以外の格子点に対しては $\alpha = 0.95$ を与えた。

また、拡散係数が Fig.10 のように斜め方向の境界で大きく変わるとき、変形 5 点差分を使った MICCG 法の (i) 反復回数は通常の 5 点差分のときより少し増加し、(ii)CPU 時間は ICCG 法で 58%(63%)、MICCG 法では 67%(70%) に短縮されたが、(iii) 誤差は逆に少し大きくなったこと (1.76 倍 ∼ 3.24 倍) などが、この表からわかる。

以上のことから、この変形 5 点差分は、境界が複雑、特に斜め方向に拡散係数が大きく変化する場合には、通常の 5 点差分のときのように MICCG 法は収束性がよくならないことがわかった。

Table 3. Numerical results for diffusion parameters $D_6(x,y)$ and $D_7(x,y)$

$D_6(x,y)$	通常 5 点	変形 5 点
反復回数	245 (145)	241 (151)
CPU [sec]	2.24 (1.32)	1.42 (0.89)
Max	5.78×10^{-11}	1.87×10^{-10}
L_2 norm	2.90×10^{-11}	5.46×10^{-11}

$D_7(x,y)$	通常 5 点	変形 5 点
反復回数	266 (139)	241 (151)
CPU [sec]	2.43 (1.27)	1.42 (0.89)
Max	6.23×10^{-11}	1.86×10^{-10}
L_2 norm	3.11×10^{-11}	5.46×10^{-11}

6 おわりに

縦横の格子幅が等しい格子系に対して、通常の x-y 座標系ではなく、それを 45 度回転した ξ-η 座標系下で方程式の離散化を行なった。このような斜め方向の 4 つの格子点を使う変形差分スキームは、拡散係数が連続的な分布のとき、通常の差分スキームで離散化したときより (M)ICCG 法の効率がよかった。また、拡散係数が不連続のとき、格子点が境界近傍の点であるかどうかによって、MIC 分解でパラメータαの値を決めれば、変形差分は同様に高い計算効率が得られる。この点は従来と異なる。

一方、拡散係数の値が大きく違う境界が斜め方向にある場合にはMICCG 法の収束性が低下する。この原因追求と収束性改善は今後の課題である。

参考文献

[1] 藤野, 竹内, 差分スキームの再考による並列計算機向き不完全 LU 分解について, 情報処理学会, HPC 研究会資料 No.49-4, 1993.10.

[2] 藤野, 長谷川, ベクトル計算機における Fill-in 付き (M)ICCG 法の性能評価, 日本応用数理学会 論文誌, **2** (1992), 105-118.

[3] Gustafsson I., A class of first order factorization method, BIT, **18**(1978), 142-156.

[4] 岩波 数学公式 I (微分積分, 平面曲線), 森口, 宇田川, 一松共著, 岩波書店, 1988, 17-18.

[5] 後 保範, ベクトル計算機向き ICCG 法, 京都大学数理解析研究所 講究録 **514**(1984), 110-134.