数理解析研究所講究録
第453巻 1982年 102-126

ベクトル計算機向き不完全三角分解

日立 ソフト　後 保範

1 はじめに

疎な対称正定値行列を係数とする連立1次方程式の計算において，不完全三角分解とCG法を組み合わせた解法は古典的なCG法やSOR法より収束が格段に良いことが最近知られてきた。[1) 2)] 一方高速計算の必要性からベクトル計算機が出現してきている。ベクトル計算機は処理するベクトルデータの中に相互に関連するデータがある場合はベクトル化できず逐次的な計算を行わざるを得なくなる。

不完全三角分解とCG法を組み合わせた計算方法をここではILUCG法と呼ぶ。CG法自体はベクトル計算機向き解法であるが，不完全三角分解と組み合わせたILUCG法はベクトルデータの中に相互に関連するデータが発生し通常ベクトル計算に不適となる。しかし偏微分方程式を差分近似することにより得られる連立1次方程式などの場合は，差分の

格子点の番号順を変更することによりベクトル計算機向きにすることが可能である。ここでは格子点の番号順を変更して得られるベクトル計算機向き行列とした場合の連立1次方程式の解の収束に対する数値実験結果を報告する。数値実験に使用した行列は2次元矩形領域における拡散方程式を5点中央差分公式で100×100に離散化した行列を使用した。また参考のため格子点の番号順の変更とSOR法の収束に関する数値実験を行ったので同時に報告する。ただしSOR法の収束実験には30×30に離散化した行列を使用した。

2. ILUCG法と他解法の収束の速さの比較

ベクトル計算機向き解法の収束の評価を行う前に，不完全三角分解とCG法を組み合わせたILUCG法と古典的CG法，SOR法及びSLOR法の収束の速さの比較を行った。その結果を以下に示す。

収束の速さの比較には以下に示す3次元拡散方程式を中央差分により離散化して得られる行列を使用した。計算対象領域は直方体領域の左下奥1/8領域である。全体領域の表面は0.0に固定された境界（Γ_1）とし，1/8領域への切断面は熱移動のない境界（Γ_2）とする。拡散係数と形状の収束性への影響を考慮して計算対象モデルには図2.1に示すケース1からケース4までの4種類のモデルを選定した。各モデルは直

方体領域の 1/8 領域に対応するものであり境界条件はすべて同一とする。

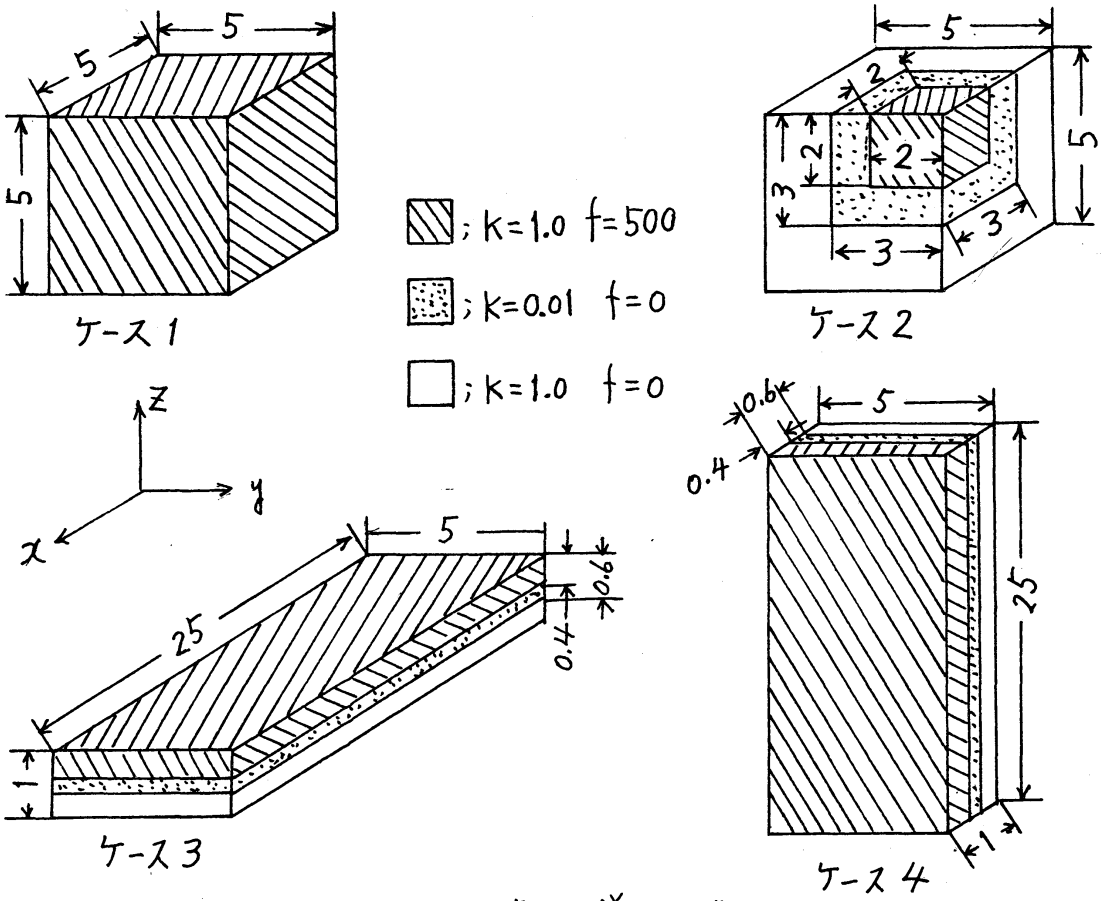

図 2.1　3次元熱伝導モデル

各モデルとも下記の3次元拡散方程式及び境界条件に従う。

$$-\left\{\frac{\partial}{\partial x}\left(K\frac{\partial \phi}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial \phi}{\partial y}\right)+\frac{\partial}{\partial z}\left(K\frac{\partial \phi}{\partial z}\right)\right\}=f \qquad (2.1)$$

$$\left.\begin{aligned} \phi &= 0 && \text{on } \Gamma_1 \\ \left(K\frac{\partial \phi}{\partial x}, k\frac{\partial \phi}{\partial y}, K\frac{\partial \phi}{\partial z}\right)^T \cdot n &= 0 && \text{on } \Gamma_2 \end{aligned}\right\} \text{境界条件} \qquad (2.2)$$

各モデルとも $20\times20\times20$ に等分割し，中央差分公式を使用して次元数8000の次のような連立1次方程式を作成する。

$$A\boldsymbol{x} = \boldsymbol{b} \qquad (2.3)$$

未知数となる節点の番号はx方向，y方向，z方向の順に付けた。$\boldsymbol{x}^{\nu}$ を ν 回反復計算時の解ベクトルとするとき，解の収束を判定する基準として次式で表現される2乗ノルムによる相対残差を使用した。

$$\text{相対残差} = \frac{\|A\boldsymbol{x}^{\nu} - \boldsymbol{b}\|_2}{\|\boldsymbol{b}\|_2} \qquad (2.4)$$

反復計算の初期値 $\boldsymbol{x}^0$ は $(0,0,\cdots,0)^T$ なるゼロベクトルとした。ケース1のモデルにおけるILUCG法，SOR法，SLOR法及びCG法の反復回数に対する収束の状態を図2.2に示す。同様にケース2，ケース3及びケース4に対するものをそれぞれ図2.3，図2.4及び図2.5に示す。CG法には加速係数が存在しない。ILUCG法も加速係数は与えないで計算できる。一方SOR法及びSLOR法には加速係数 α が必要であり，その値はYoung-Frankel[3]の理論により次式で計算した。

$$\left.\begin{aligned} \alpha &= 1 + (1-\sqrt{1-\lambda})/(1+\sqrt{1-\lambda}) \\ \omega_i &= \|A\boldsymbol{x}^i - \boldsymbol{b}\|_2 / \|\boldsymbol{b}\|_2 \quad (i=\ell-3,\ell-2,\ell-1,\ell) \\ \lambda_i &= \omega_i / \omega_{i-1} \quad (i=\ell-2,\ell-1,\ell) \\ r &= (\lambda_\ell - \lambda_{\ell-1})/(\lambda_{\ell-1} - \lambda_{\ell-2}) \end{aligned}\right\} \qquad (2.5)$$

$$\lambda = \lambda_{\ell} + (\lambda_{\ell} - \lambda_{\ell-1}) \cdot r/(1-r)$$

ℓはここでは25とした。

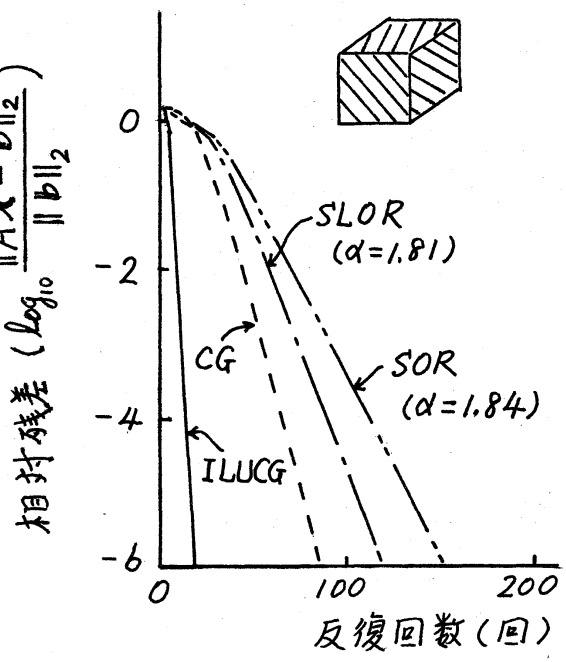

図2.2　ケース1の結果

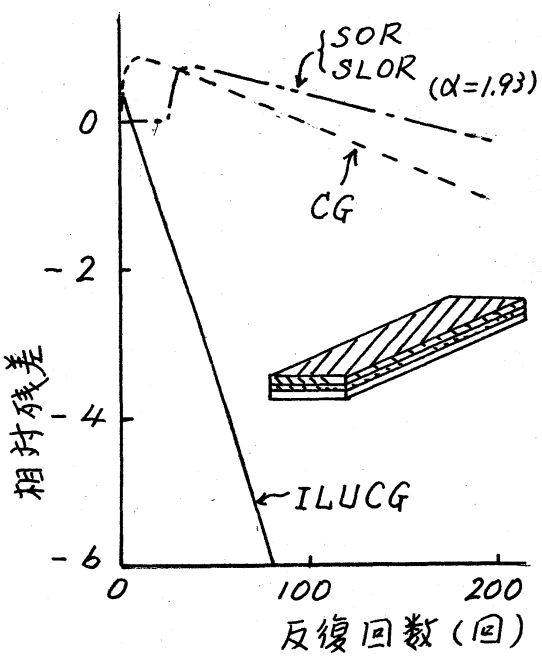

図2.4　ケース3の結果

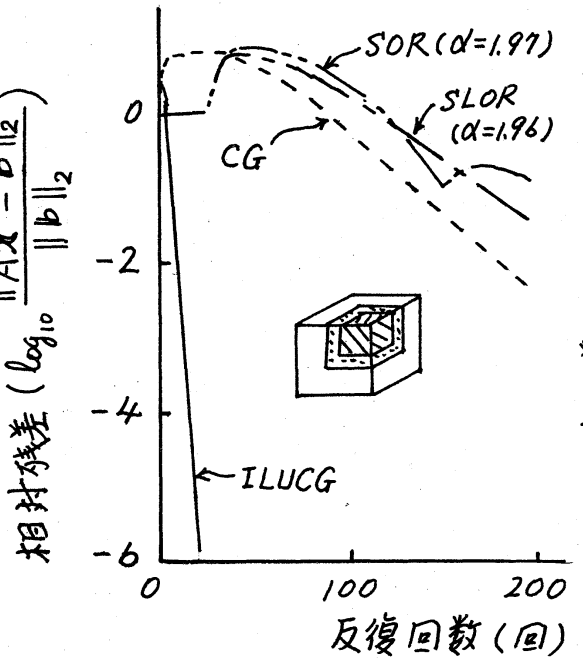

図2.3　ケース2の結果

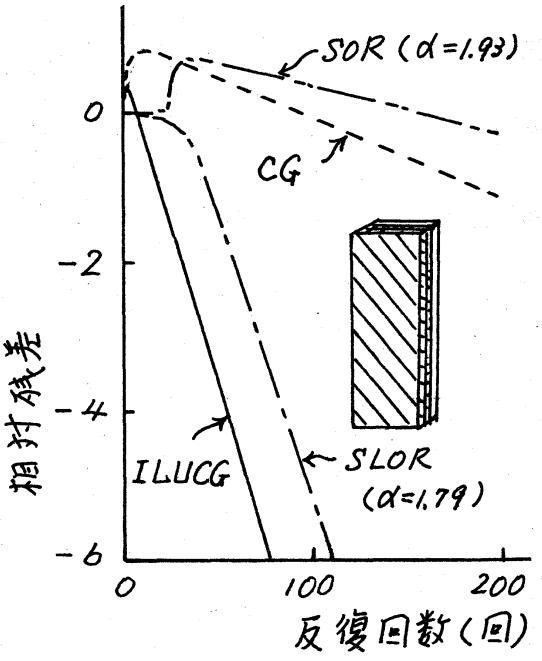

図2.5　ケース4の結果

図2.2から図2.5におけるSOR法とSLOR法の加速係数αはα=1.0でℓ回（ここでは25回）計算を行い，それ以後(2.5)式で算出したαを使用した。ここで算出した加速係数αがどの程度最適加速係数に近い値であるか調査するため，算出した加速係数αの近傍の加速係数を使用して一定回数反復計算時の相対残差を求めた。図2.6にSOR法の加速係数を変化させた場合の相対残差のグラフを示す。同様に図2.7にSLOR法に関するグラフを示す。両図ともαは0.02単位に変化させたものである。これからSOR法及びSLOR法ともに(2.5)式で計算した加速係数αが最適加速係数に近いものであることがわかる。

図2.2から図2.5よりILUCG法は4種類のモデルのすべてでSOR法，SLOR法及びCG法より収束が速くかつ安定した解法であることがわかる。ILUCG法とほぼ同等の性能を示しているのは図2.5におけるケース4のモデルに適用した場合のSLOR法のみである。特に図2.3及び図2.4に示すケース2及びケース3の2モデルの場合はILUCG法は他の3つの反復解法に比較して圧倒的に収束が速い解法であることがわかる。図2.4と図2.5は同一のモデルに対し未知数となる格子点の番号付けを変えたものである。ILUCG法，SOR法及びCG法は図2.4と図2.5で同一の収束性を示して

いる。一方SLOR法は図2.4ではSOR法と同一の収束性を，図2.5ではILUCG法に近い収束性を示している。これはSLOR法が方向性に大きく依存すること及びSLOR法が図2.5のモデルに適した解法であることを示している。

100回反復計算するまでのCPU時間をHITAC M-200Hで測定した結果，ILUCG法は14.5秒，SOR法は7.5秒，SLOR法は8.2秒，CG法は8.6秒であった。本測定は反復計算5回に1回の割り合いで(2.4)式に示す相対残差を計算して出力する方式を採用して行った。CPU時間を基準にしてもILUCG法は他の3解法に比較して収束が速い。ただし図2.5のようにSLOR法に最も適したモデルの場合にはSLOR法の方がわずかにILUCG法より収束が速い場合がある。

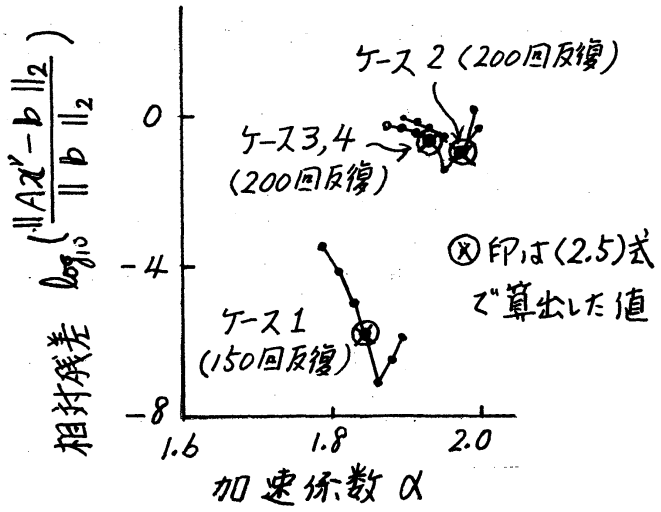

図2.6 SOR法の結果

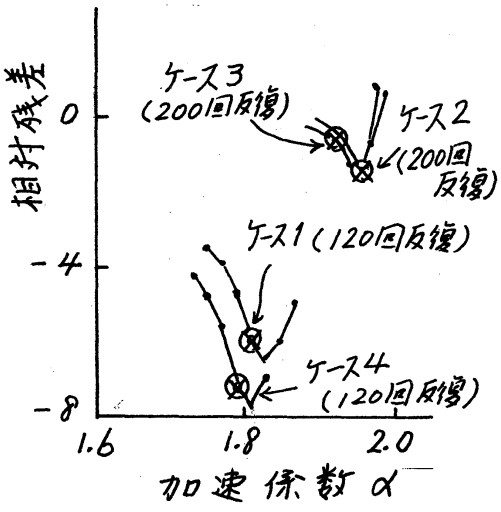

図2.7 SLOR法の結果

3 計算方法

まず最初に不完全三角分解とCG法を組み合わせたILUCG法の計算方法を述べる。ILUCG法はIncomplete LU Decomposition and Conjugate Gradient Methodの略である。疎な対称正定値行列Aを係数とする連立1次方程式を$Ax=b$とし，Aの不完全三角分解をLDL^Tと表すとILUCG法は図3.1で示す通りである。

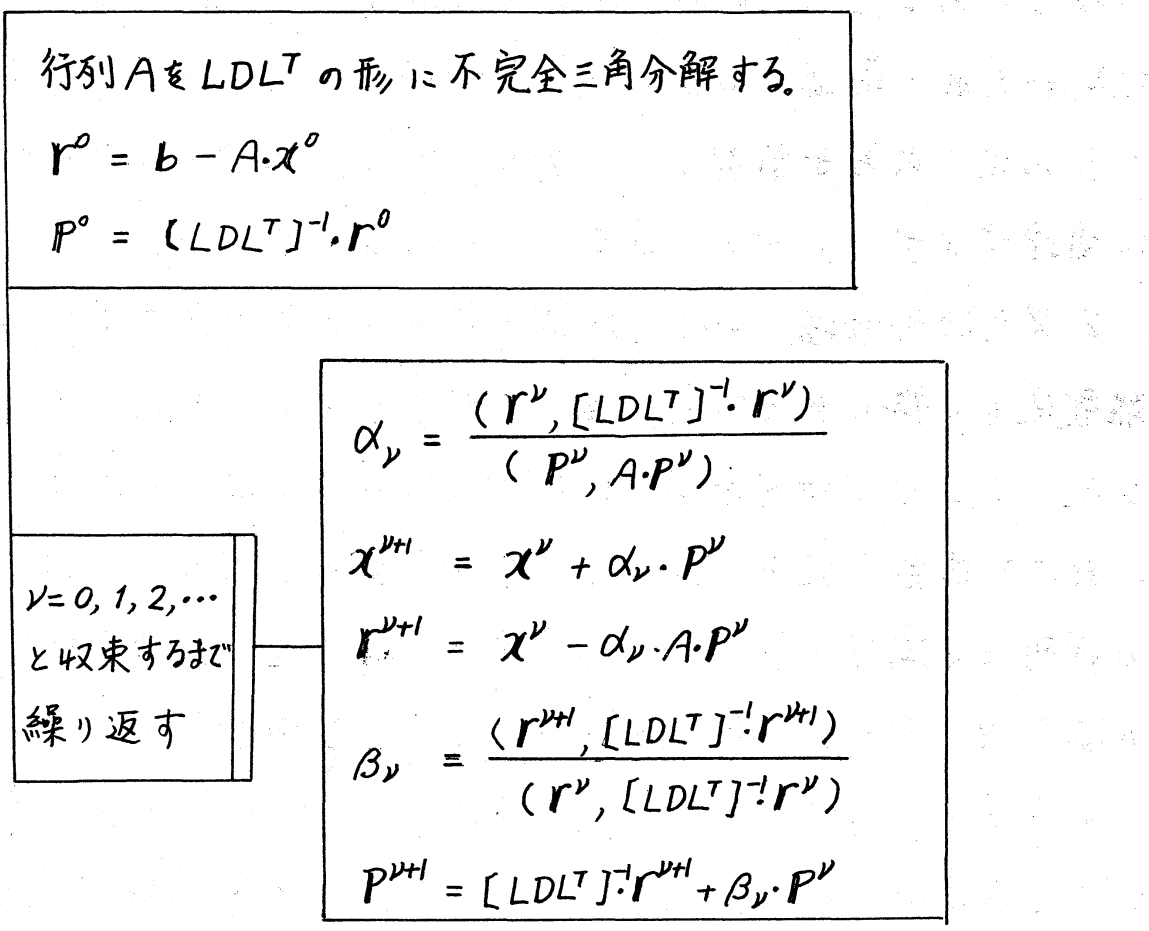

図3.1 ILUCG法の計算方法

図3.1において（r, p）は内積を示し，$[LDL^T]^{-1}r$ は行列Aを不完全三角分解したLDL^T行列を使用して$LDL^T z = r$なる連立1次方程式の解zを前進及び後退代入で計算することを意味する。図3.1の反復計算1回当り$[LDL^T]^{-1}r^{\nu}$及び$[LDL^T]^{-1}r^{\nu+1}$が合計4回，$A \cdot p^{\nu}$が2回出現しているが計算はともに1回でよい。

ベクトル計算機で図3.1に示す計算を行うとき問題となるのは行列AをLDL^Tの形に不完全三角分解する部分と$[LDL^T]^{-1}r$で表わされる前進，後退代入計算部分である。他の部分はベクトル化されるが前記2つの部分はベクトルデータ中に相互に関連するデータがあり通常ベクトル化できない。

2次元矩形領域における拡散方程式を5点中央差分公式で離散化して得られる行列に関してベクトル化の可能性を検討した。ここでの検討事項はほぼ同じ方法で3次元直方体領域における拡散方程式を7点中央差分公式で離散化して得られる行列にも適用できる。ここで検討した方法は差分の格子点の番号順を変えた3種類の方法である。3種類の方法をそれぞれ方式1，方式2，方式3と名付ける。以下にそれぞれの格子点の番号付け，行列Aの形及び不完全三角分解と前進後退代入部分の計算式を示す。方式1は通常の番号付けでベクトル化できない。方式2と方式3は共にベクトル化可能であ

る。領域の分割数を $n_x \times n_y$ とすると方式2のベクトル長は $n_x \times n_y / 2$ であり，方式3のベクトル長は $\min(n_x, n_y)$ となる。

方式1は横方向又は縦方向に格子点の番号を連続して付けたものである。格子点の番号付けの例を図3.2に示す。係数行列Aを図式的に示すと図3.3のようになる。係数行列Aの次元数を n，帯の半幅を m とする。Aの対角要素を a_i，対角の1つ下の要素を b_i，対角より m 離れた要素を c_i とする。

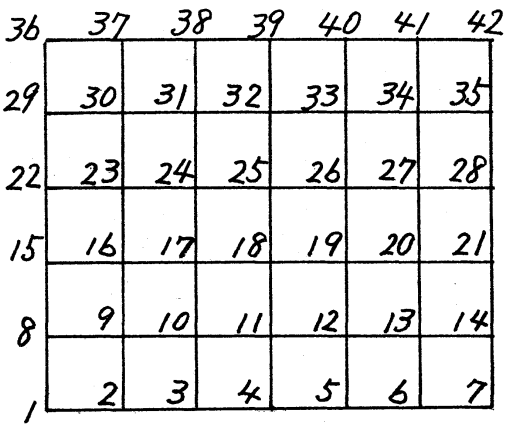

図3.2 方式1の番号付け

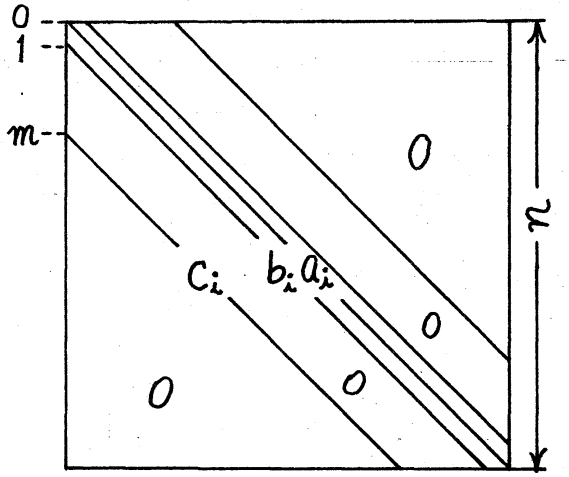

図3.3 方式1の行列Aの形

図3.3の形の行列に対し完全な三角分解を行うと帯内全体が非ゼロ要素となるが，c_i 及び b_i より離れるに従って通常非ゼロ要素の値は急激に小さくなる。不完全三角分解とは急激に小さくなる要素は無視し0として三角分解する方法である。不完全三角分解は分解後の行列が持つ非ゼロ要素の数を増減

させることにより各種作成できる。不完全三角分解した下三角行列をL，対角行列をDと表す。ここではLの非ゼロ要素の位置が元の行列Aの下三角行列の非ゼロ要素の位置と同一となるような不完全三角分解を採用する。行列Lの形を図3.4で表す。対角行列Dの要素をd_iで表す。行列Lは対角要素を1でなく$1/d_i$とすると対角以外の要素は元の行列Aの下三角行列と値まで同一とすることができる。図3.5にLDL^Tの計算を行った結果の行列の形を示す。行列AとLDL^Tとの差は図3.5に示すようにe_i及びその対称部分にのみ発生する。

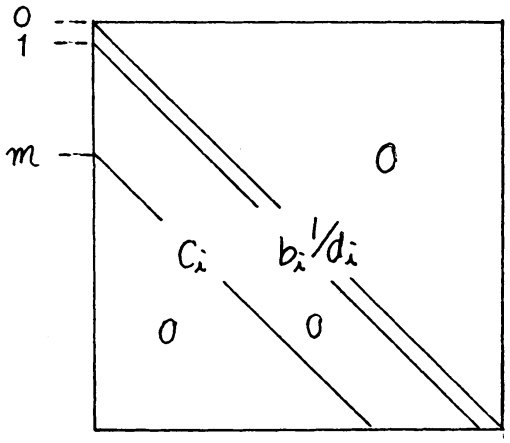

図3.4 方式1の行列Lの形

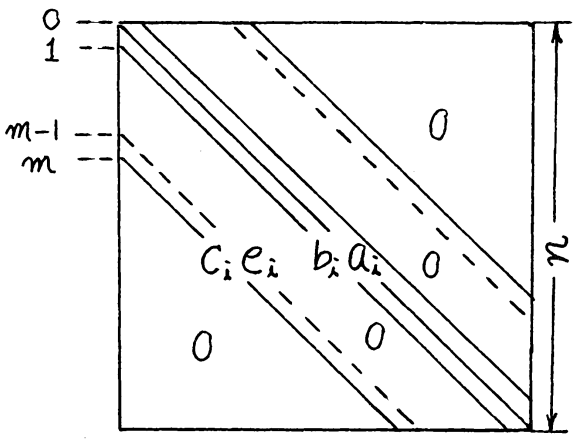

図3.5 方式1の行列LDL^Tの形

方式1の不完全三角分解と前進後退代入の計算方法を以下に示す。不完全三角分解においてはd_iの値のみ計算すればよい。前進後退代入は$LDL^T x = r$の形の連立1次方程式を解くことである。

方式1の不完全三角分解（$LDL^T \longleftarrow A$）

$$d_i \longleftarrow \frac{1}{a_i - b_i^2 \times d_{i-1} - c_i^2 \times d_{i-m}} \qquad (3.1)$$

$$(i=1,2,\cdots,n)$$

方式1の前進後退代入（$x \longleftarrow [LDL^T]^{-1} \cdot r$）

$$x_i \longleftarrow (r_i - b_i \times x_{i-1} - c_i \times x_{i-m}) \times d_i \qquad (3.2)$$

$$(i=1,2,\cdots,n)$$

$$x_i \longleftarrow x_i - (b_{i+1} \times x_{i+1} + c_{i+m} \times x_{i+m}) \times d_i \qquad (3.3)$$

$$(i=n,n-1,\cdots,1)$$

(3.1)式はd_iの計算に1つ前で計算したd_{i-1}の値が必要なためベクトル化されない。同様な理由で(3.2)式及び(3.3)式もベクトル化されない。一部のベクトル計算機はFirst order Iteration命令（$x_i \leftarrow a_i + b_i \times x_{i-1}$）を持っており(3.2)式及び(3.3)式は形式上はベクトル化することができる。しかしFirst Order Iteration命令は本質的な先行計算ができないという意味で他のベクトル命令と本質的に異なり，ベクトル計算機による高速化には限界がある。ここではFirst Order Iteration命令を含むものはベクトル化できない方に位置づけした。

<u>方式2</u>は格子点i，jで$i+j$＝奇数のものを先に番号付けし，その後で$i+j$＝偶数のものに番号を付けたものである。

格子点の番号付けの例を図3.6に示す。係数行列Aを図式的に示すと図3.7のようになる。ただし図3.6で横方向の格子点の数が偶数の場合はその方向に1つ余分な格子点を追加して格子点の番号付けを行う。追加した格子点に対応する行列は対角要素を1とし，その他の要素はすべて0とする。
次元数をnとし，先に番号付けした格子点の数をn_1，後で番号付けした格子点の数をn_2とする。nが偶数のときは$n_1 = n_2 = n/2$であり，奇数のときは$n_1 = n_2 + 1 = (n+1)/2$となる。

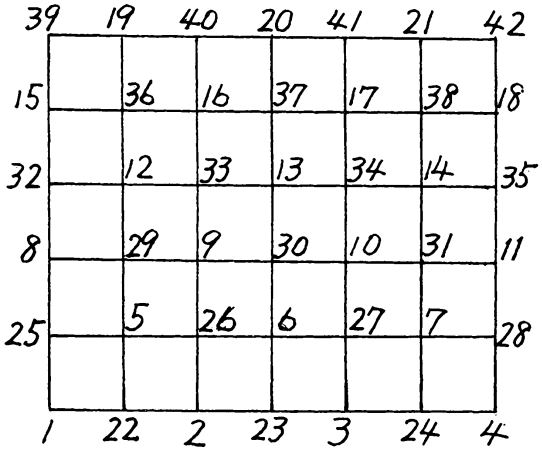

図3.6 方式2の番号付け

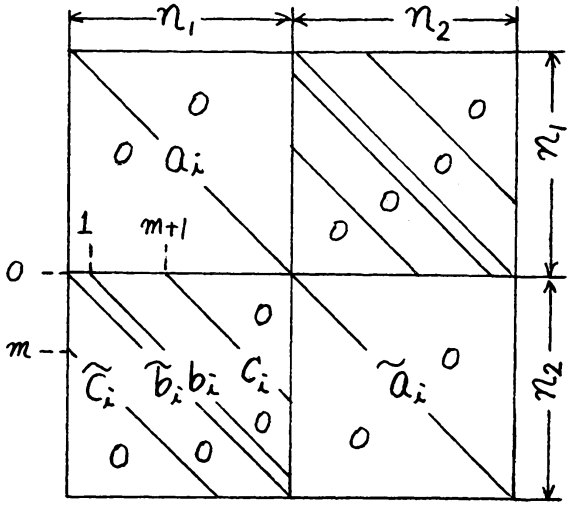

図3.7 方式2の行列Aの形

不完全三角分解した下三角行列をL，対角行列をDと表す。不完全三角分解の方法は方式1と同様に元の行列Aの下三角行列にのみ非ゼロ要素が発生するものとする。行列Lの形を図3.8で表す。対角行列Dの要素はn_1次元までd_iでそれ以後

はα_iで表す。行列Lは対角要素を除き元の行列Aの下三角行列と値まで同一のものとすることができる。図3.9にLDL^{T}の計算を行った結果の行列の形を示す。ただし図3.7の右下$n_2 \times n_2$部分のみ示す。他の部分は元の行列Aと同一である。

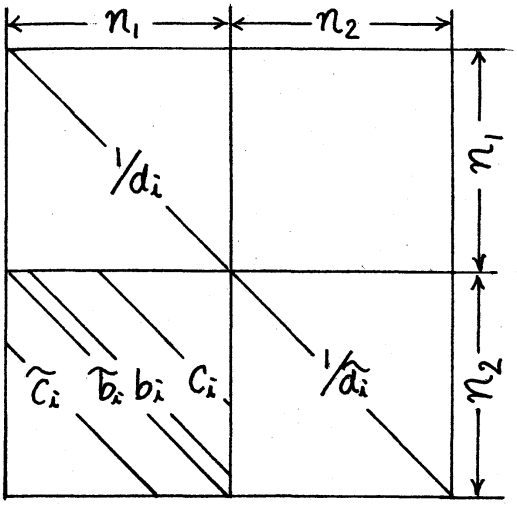

図3.8 方式2の行列Lの形

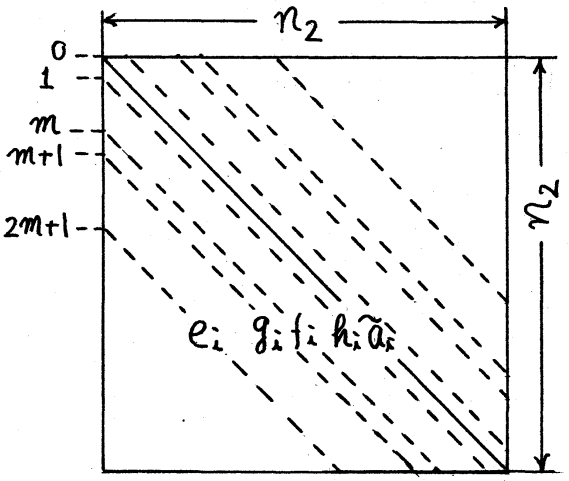

図3.9 方式2の行列LDL^{T}の形
(右下1/4)

方式2の不完全三角分解 (LDL^T ← A)

$$d_i \longleftarrow \frac{1}{a_i} \qquad (i=1,2,\cdots,n_1) \qquad (3.4)$$

$$\tilde{d}_i \longleftarrow \frac{1}{\tilde{a}_i - \tilde{c}_i^2 \times d_{i-m} - \tilde{b}_i^2 \times d_i - b_i^2 \times d_{i+1} - c_i^2 \times d_{i+m+1}} \qquad (3.5)$$

$$(i=1,2,\cdots,n_2)$$

方式2の前進・後退代入 ($x \longleftarrow [LDL^T]^{-1} r$)

$x = (x_1, x_2, \cdots x_{n_1}, \tilde{x}_1, \tilde{x}_2, \cdots \tilde{x}_{n_2})^T$ 及び

$r = (r_1, r_2, \cdots r_{n_1}, \tilde{r}_1, \tilde{r}_2, \cdots \tilde{r}_{n_2})^T$ とする。

$$x_i \longleftarrow r_i \times d_i \qquad (i=1,2,\cdots,n_1) \qquad (3.6)$$

$$\tilde{x}_i \longleftarrow (\tilde{r}_i - \tilde{c}_i \times x_{i-m} - \tilde{b}_i \times x_i - b_i \times x_{i+1} - c_i \times x_{i+m+1}) \times d_i \qquad (3.7)$$

$$(i = 1, 2, \cdots, n_2)$$

$$x_i \longleftarrow x_i - (c_{i-m-1} \times \tilde{x}_{i-m-1} + b_{i-1} \times \tilde{x}_{i-1} + \tilde{b}_i \times \tilde{x}_i + \tilde{c}_{i+m} \times \tilde{x}_{i+m}) \times d_i \qquad (3.8)$$

$$(i = n_1, n_1-1, \cdots, 1)$$

(3.4)式から(3.8)式はすべてベクトルデータ中に相互に関連するデータがないためベクトル化される。ベクトル長は次元数nの約半分となる。

方式3は図3.10に示すように対角線方向に格子点の番号を付けたものである。左上の三角形領域の格子点はそのまま右に平行移動して点線の位置にあると仮定して帯幅が一定となるように番号を付ける。係数行列Aを図式的に示すと図3.11のようになる。矩形領域を$n_x \times n_y$の格子点で分割し，$m = \min(n_x, n_y)$，$l = \max(n_x, n_y)$とする。

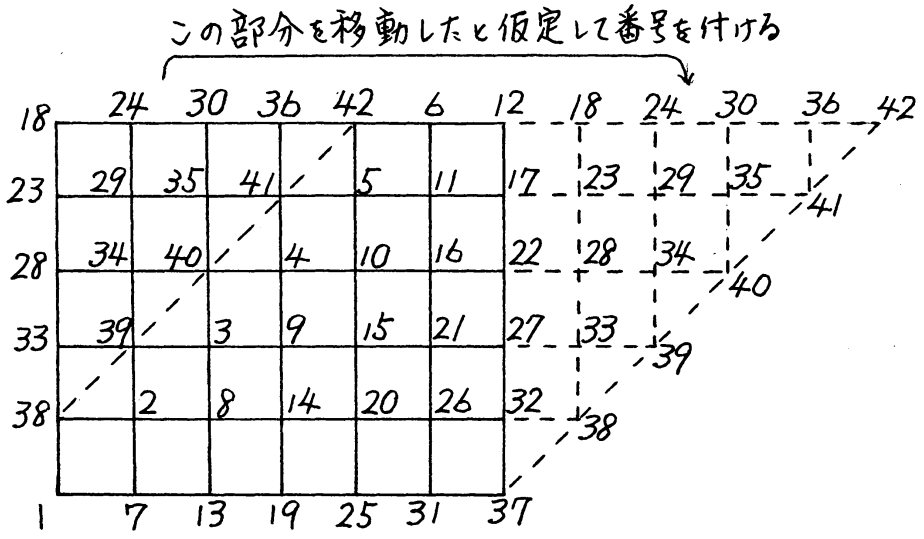

図3.10　方式3の番号付け

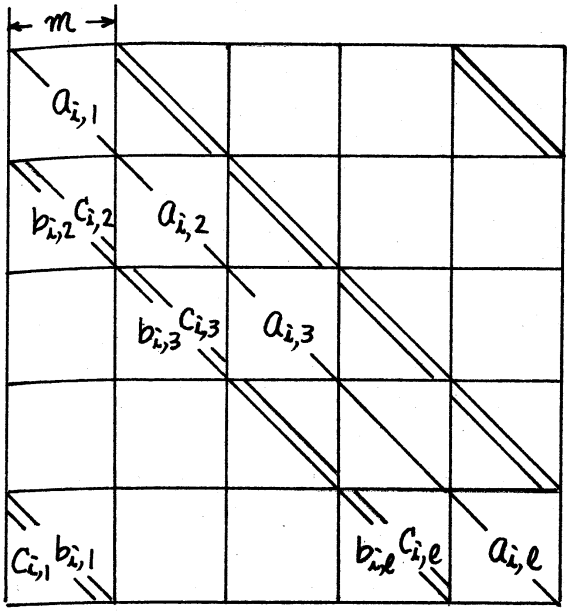

図3.11 方式3の行列Aの形

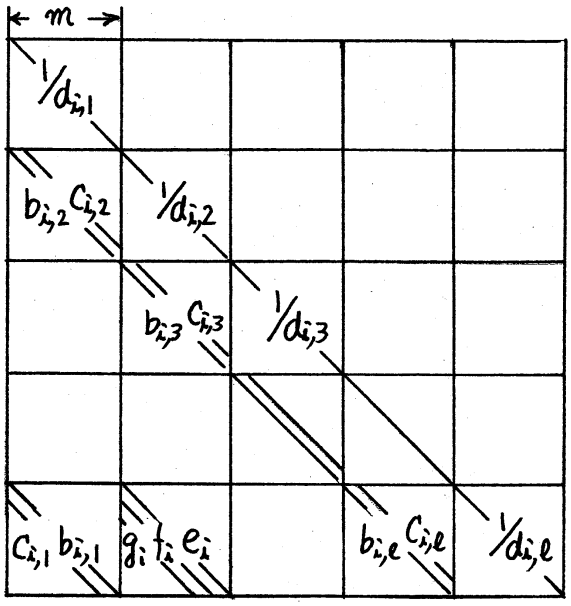

図3.12 方式3の行列Lの形

不完全三角分解した下三角行列をL，対角行列をDと表す．不完全三角分解後の行列Lの形を図3.12に示す。行列Lは元の行列Aの下三角行列の非ゼロ要素よりe_i, f_i, g_iだけ非ゼロ要素を追加した。この追加により反復計算1回当りの計算量は30x30分割で2%，100x100分割で0.6%増加するのみである。e_i, f_i, g_iの部分に非ゼロ要素を追加したのは元の行列AとLDL^Tとの相異がこの部分で特に大きくなる可能性を除くためである。対角行列Dの要素を$d_{i,j}$とする。行列Aの要素$b_{i,j}$及び$c_{i,j}$は行列Lでも値がそのまま受け継がれる。

方式3の不完全三角分解（$LDL^T \leftarrow A$）

$$d_{i,1} \leftarrow \frac{1}{a_{i,1}} \qquad (i=1,2,\cdots,m) \qquad (3.9)$$

$$d_{i,j} \longleftarrow \frac{1}{a_{i,j} - b_{i,j}^2 \times d_{i,j-1} - c_{i,j}^2 \times d_{i+1,j-1}} \qquad (3.10)$$

$$(i = 1, 2, \cdots, m \ , \ j = 2, 3, \cdots, \ell - 1)$$

$$\left.\begin{aligned} e_i &\longleftarrow -b_{i,1} \times b_{i,2} \times d_{i,1} \\ f_i &\longleftarrow -c_{i,1} \times b_{i-1,2} \times d_{i-1,1} - b_{i,1} \times c_{i-1,2} \times d_{i,1} \\ g_i &\longleftarrow -c_{i,1} \times c_{i-2,2} \times d_{i-1,1} \end{aligned}\right\} \qquad (3.11)$$

$$(i = 1, 2, \cdots, m)$$

$$\left.\begin{aligned} d_{i,\ell} &\longleftarrow \frac{1}{h} \\ h &= a_{i,\ell} - b_{i,\ell}^2 \times d_{i,\ell-1} - c_{i,\ell}^2 \times d_{i+1,\ell-1} - b_{i,1}^2 \times d_{i,1} \\ &\quad - c_{i,1} \times d_{i-1,1} - e_i^2 \times d_{i,2} - f_i^2 \times d_{i-1,2} - g_i^2 \times d_{i-2,2} \end{aligned}\right\} \qquad (3.12)$$

$$(i = 1, 2, \cdots, m)$$

方式3の前進・後退代入（$x \longleftarrow [LDL^T]^{-1} \cdot r$）

$$x_{i,1} \longleftarrow r_{i,1} \times d_{i,1} \qquad (i = 1, 2, \cdots, m) \qquad (3.13)$$

$$x_{i,j} \longleftarrow (r_{i,j} - b_{i,j} \times x_{i,j-1} - c_{i,j} \times x_{i+1,j-1}) \times d_{i,j} \qquad (3.14)$$

$$(i = 1, 2, \cdots, m \ , \ j = 2, 3, \cdots, \ell - 1)$$

$$x_{i,\ell} \longleftarrow (r_{i,\ell} - b_{i,\ell} \times x_{i,\ell-1} - c_{i,\ell} \times x_{i+1,\ell-1} - b_{i,1} \times x_{i,1} - c_{i,1} \times x_{i-1,1}$$
$$- e_i \times x_{i,2} - f_i \times x_{i-1,2} - g_i \times x_{i-2,2}) \times d_{i,\ell} \qquad (3.15)$$

$$(i = 1, 2, \cdots, m)$$

$$x_{i,j} \longleftarrow x_{i,j} - (b_{i,j+1} \times x_{i,j+1} + c_{i-1,j+1} \times x_{i-1,j+1}) \times d_{i,j} \qquad (3.16)$$

$$(i = m, m-1, \cdots, 1 \ , \ j = \ell - 1, \ell - 2, \cdots, 3)$$

$$x_{i,2} \longleftarrow x_{i,2} - (b_{i,3} \times x_{i,3} + c_{i-1,3} \times x_{i-1,3} + e_i \times x_{i,\ell} + f_{i+1} \times x_{i+1,\ell}$$
$$+ g_{i+2} \times x_{i+2,\ell}) \times d_{i,2} \qquad (i = m, m-1, \cdots 1) \qquad (3.17)$$

$$x_{i,1} \longleftarrow x_{i,1} - (b_{i,2} \times x_{i,2} + c_{i-1,2} \times x_{i-1,2} + b_{i,1} \times x_{i,e} + c_{i+1,1} \times x_{i+1,e}) \times d_{i,1} \quad (i = m, m-1, \cdots 1) \quad (3.18)$$

(3.9)式から(3.18)式はすべて添字 i を変化させる範囲ではベクトル化される。このためベクトル長は $\min(n_x, n_y)$ となる。

4 数値実験結果

方式1，方式2，方式3の収束の速さを比較するために数値実験を行った。その結果を以下に示す。

収束の速さの比較には以下に示す2次元拡散方程式を中央差分により離散化して得られる行列を使用した。計算対象モデルは図4.1に示すケースAからケースCまで3種類のモデルを選定した。

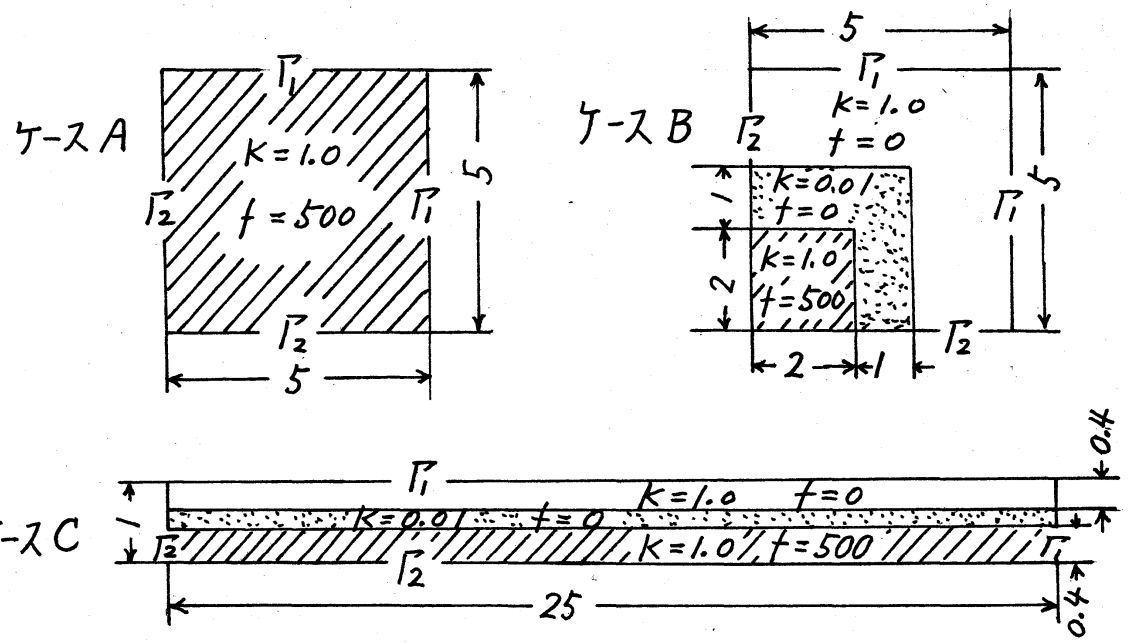

図4.1 2次元熱伝導モデル

各モデルとも下記の2次元拡散方程式及び境界条件に従う。

$$-\left\{\frac{\partial}{\partial x}\left(k\frac{\partial \phi}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial \phi}{\partial y}\right)\right\} = f \qquad (4.1)$$

$$\left.\begin{aligned} \phi &= 0 && \text{on } \Gamma_1 \\ \left(k\frac{\partial \phi}{\partial x}, k\frac{\partial \phi}{\partial y}\right)^T \cdot n &= 0 && \text{on } \Gamma_2 \end{aligned}\right\} \text{境界条件} \qquad (4.2)$$

各モデルとも100×100に等分割し，中央差分公式を使用して次元数10000（ただし方式2ではダミーの格子点を追加しているため11000）の次のような連立1次方程式を作成する。

$$A x = b \qquad (4.3)$$

x^ν を ν 回反復計算時の解ベクトルとするとき解の収束を判定する基準として次式で表現される2乗ノルムによる相対残差を使用した。

$$\text{相対残差} = \frac{\|Ax^\nu - b\|_2}{\|b\|_2} \qquad (4.4)$$

反復計算の初期値 x^0 は $(0,0,\cdots,0)^T$ なるゼロベクトルとした。ケースAのモデルにおける方式1，方式2，方式3の3種類のILUCG法の反復回数に対する収束の状態を図4.2に示す。ケースBのモデルにおける収束の状態を図4.3に，ケースCのモデルにおける収束の状態を図4.4に示す。

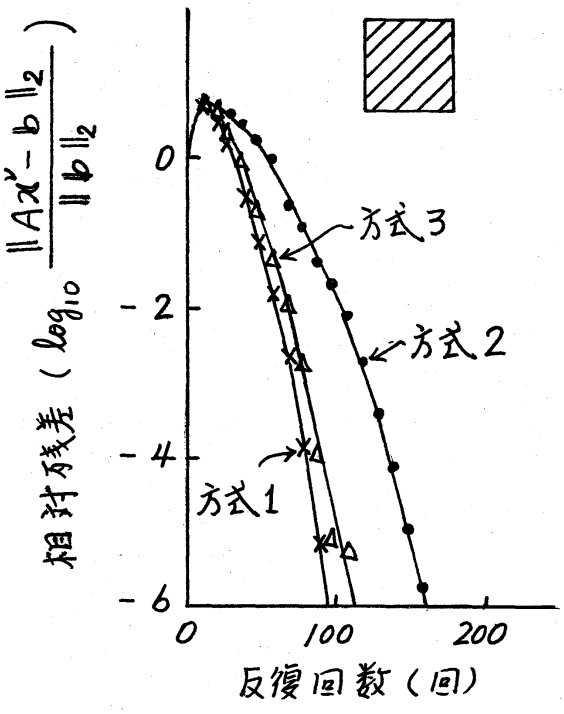

図4.2 ケースAの結果

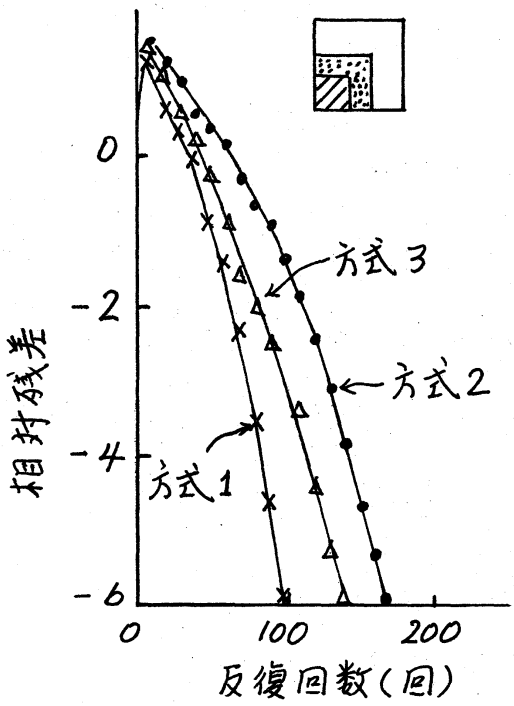

図4.3 ケースBの結果

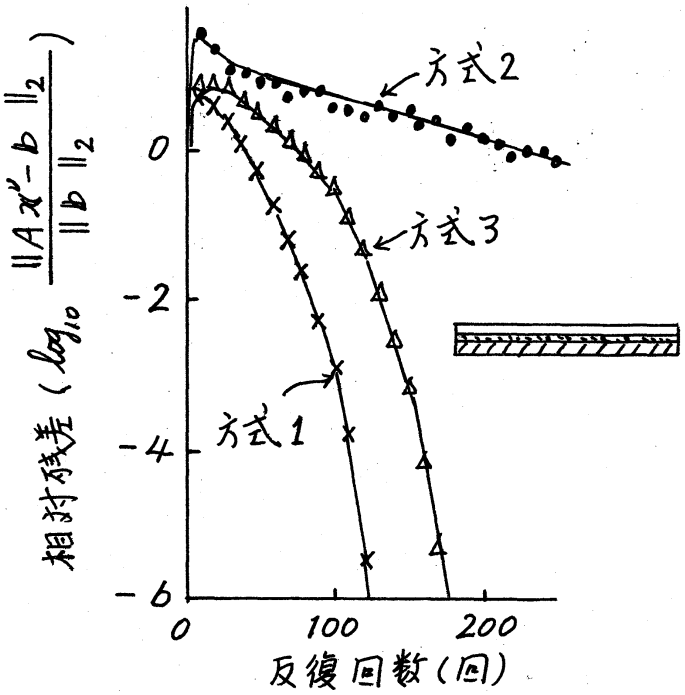

図4.4 ケースCの結果

いずれも

100×100 分割

ケースA，ケースB，ケースCのいずれのモデルにおいても方式1が反復1回当りの収束は最も速く，次いで方式3の収束が速い。方式2はいずれのケースでも収束が最も悪い。特に図4.4よりケースCのモデルでは方式2の収束が他の2つの方式に比較して極度に悪い。一方ベクトル計算の立場より見ると方式2が最も都合が良く，ベクトル計算機の能力を十分に発揮できる。方式3はベクトル化されるがベクトル長が短かいため方式2よりは少しばかりベクトル計算機の効率は低下する。方式1は不完全三角分解及び毎回の反復計算中の前進後退代入部分がベクトル化されずベクトル計算機の特性が生かされない。本数値実験結果より不完全三角分解とCG法を組み合わせたILUCG法において，ベクトル計算機向き修正を行うと反復1回当りの収束率は方式1に比較して悪化することがわかる。しかし方式3のような工夫をすれば収束率をあまり悪化させないでベクトル計算機向き修正がある程度は可能である。

ケースCで方式2の収束が悪いのは図3.9のe_iの項に大きな値が発生するためである。e_iの項は図3.8の$\tilde{b}_i$の項とb_iの項により発生する。方式1及び方式3ではこれらの項に対応する部分は0となるように三角分解している。

ILUCG法はベクトル計算機向き修正を行うと反復1回当

りの収束率が悪化した。SOR法ではどのようになるかを検討するため方式1，方式2及び方式3で作成したのと同じ方法で作成した行列でSOR法の収束の速さを調べた。SOR法の数値実験はモデルの分割数を小さくして30x30分割で行った。SOR法の数値実験では加速係数が問題となる。ここではケースAには0.01きざみ，ケースBとケースCには0.002きざみの加速係数を使用して収束を調べ，最も収束が速い加速係数を選んだ。いずれのケースも最適な加速係数は格子点の番号付けには依存せず，ケースAでは1.91，ケースBでは1.978，ケースCでは1.980となった。ケースAのモデルにおけるSOR法の反復回数に対する収束の状態を図4.5に示す。ケースBのモデルにおける収束の状態を図4.6に，ケースCのモデルにおける収束の状態を図4.7に示す。いずれも方式1，方式2及び方式3で作成した行列Aを使用したSOR法の収束が示してある。参考のためにILUCG法では収束が最も遅い方式2のILUCG法も示した。これらの図よりSOR法においては格子点の番号付けにより反復1回当りの収束率はほとんど変化しないことがわかる。ただし最初の反復計算での相対残差は異なる。SOR法の場合はILUCG法の場合とは様子が異なり方式3が同一反復回数では最も小さい相対残差を示す。

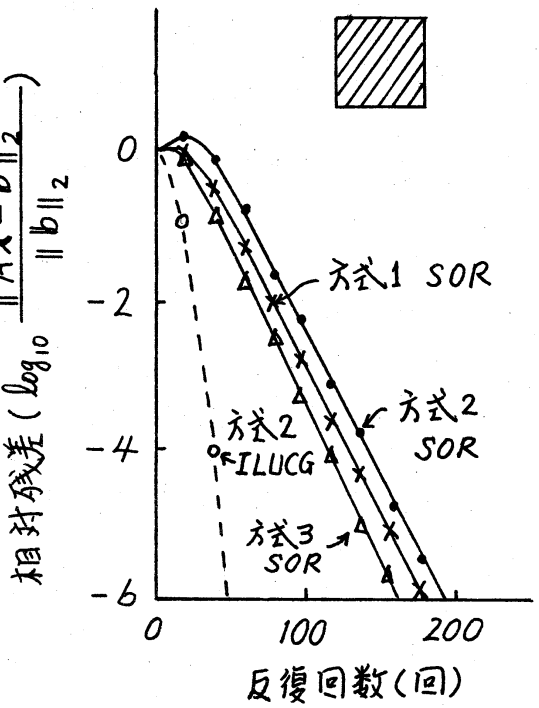

図4.5　ケースAの結果

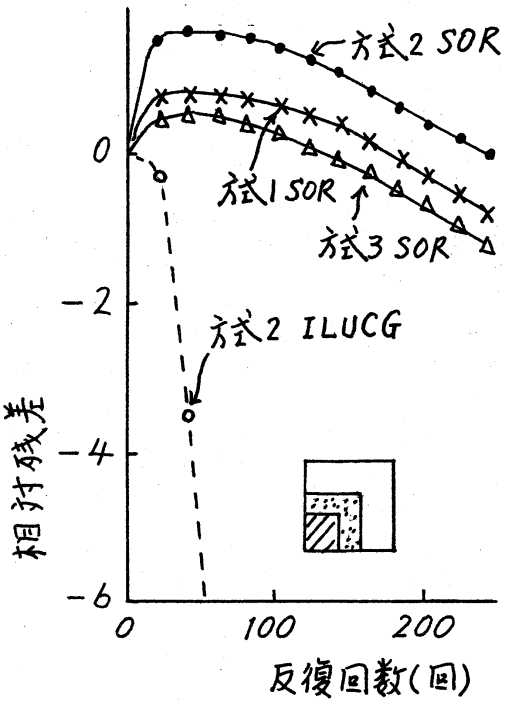

図4.6　ケースBの結果

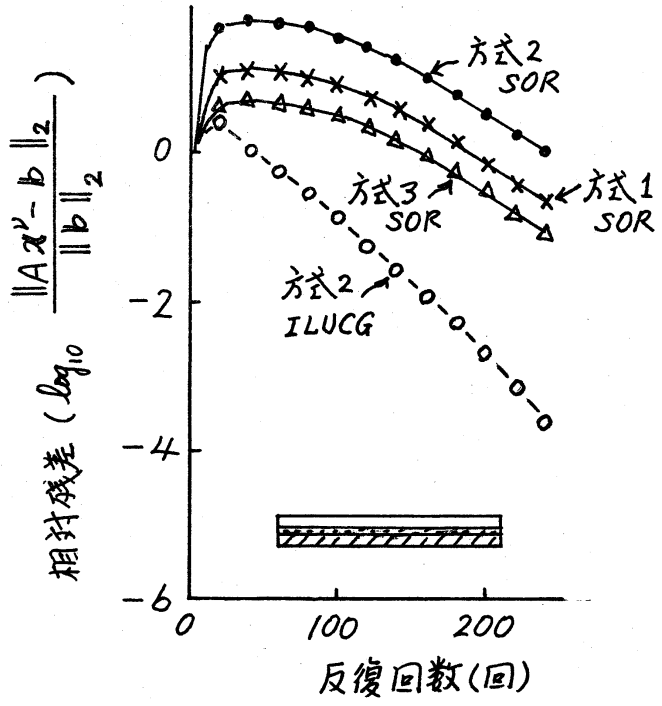

図4.7　ケースCの結果

いずれも
30×30 分割

5. おわりに

拡散方程式より作成される疎な対称正定値行列を係数とする連立1次方程式において，不完全三角分解とCG法を組み合わせた解法はCG法，SOR法，SLOR法等に比較すると収束が格段に良い。しかし一般にはこの解法はベクトル計算機向きではない。拡散方程式を差分により離散化するとき格子点の番号付けを工夫するとSOR法等と同じようにベクトル計算向けに修正することができる。しかしベクトル計算機向け修正を行った場合は一般的な格子点の番号付け方法に比較して収束が悪くなる。収束の悪化を小さくする方法として3章の方式3で示す格子点の番号付けと不完全三角分解を提案した。この方法はまだ応用範囲が限定されており，一般的な要素分割を行った有限要素法等より作成される行列への適用が今後の課題である。

謝辞　御指導いただいた筑波大学 名取 亮助教授及び東京都立大学 小野寺 嘉孝助教授に感謝します。

参考文献

1) J.A. Meijerink and H.A. Van der Vorst ; An Iterative Solution Method for Linear Systems of which the Coefficient Matrix is a Symmetric M-Matrix ; Math. Comp. Vol.31 , N137, PP 148～162 (1977)

2) T.A. Manteuffel ; An Incomplete Factorization Technique for Positive Definite Linear Systems ; Math Comp Vol. 34 N150, PP 473～497 (1980)

3) フォーサイス, ワソー , 藤野精一 ; 偏微分方程式の差分法による近似解法 ; 第3章, 吉岡書店 (1976)

4) R.S. バーガ , 渋谷政昭 ; 計算機による大型行列の反復解法 ; サイエンス社 (1972)

5) 戸川隼人 ; 共役勾配法 ; 教育出版 (1977)

6) 山田博 ; コンピューター・アーキテクチャ ; 産業図書 (1976)

7) 星野力 ; 並列計算機のもたらす技術計算へのインパクト ; 日本原子力学会誌 , Vol. 22 , N.4 , PP 204～212 (1980)