

ICCG/MICCG 法の、多段メッシュによる加速について

許 夏 鎮
中国延辺大学

村 田 健 郎
図書館情報大学

1. はじめに

移流項の無い線形問題: $\text{div}(-k(x) \text{grad } u) + c(x) = f(x)$

あるいは弱い非線形問題に対しては、三次元問題まで含めて多くの場合に ICCG 法 (Incomplete Cholesky Conjugate Gradient 法 — 以下略して ICCG 法)^{1),2)} や MICCG 法 (Modified Incomplete Cholesky Conjugate Gradient 法 — 以下略して MICCG 法)³⁾ だけで満足な結果が得られるようになった。

しかし、強い非線形問題になると、それだけで押し通すのでは難儀な問題が多いし、なおもうひと工夫が要る。

最近、不完全コレスキー分解による PCG 法 (Preconditioned Conjugate Gradient 法)、特に ICCG 法の差分法と、それに Gustafsson による修正を行った MICCG 法の比較及びそれぞれに Multigrid 法併用の初期 preconditioning を施す手法を試みているので、以下そのことについて述べる。それは「粗いあみ目による ICCG/MICCG 法の解から精いあみ目の ICCG/MICCG 法のための初期解を作る」という、初期解に対する一種の preconditioning を施す方法で、ある意味で言えば、いわゆる ICCG/MICCG 法と Multigrid 法との組合せと言う風にも見られようが、ここで採用している Multigrid とは、「粗いあみ目から精いあみ目に向けての、一方向的に計算を進めて行く」多段メッシュによる ICCG/MICCG 法である。

2. ICCG/MICCG 法についての簡述

叙述の便宜のために、まず ICCG/MICCG 法について概略的に説明する。

よく知られている CG 法 (Conjugate Gradient 法) は 1952 年に発表されたが、1970 年代に入っては PCG 法の形に見直され、現在では三次元の場の問題を解くための最も有力な方法と認められている。

ICCG 法とは、対称正定値行列を係数行列とする連立一次方程式:

$$Ax = b \quad (1)$$

を解くために、係数行列 A に対して不完全コレスキー分解:

$$A = U^T U - R, \quad (U^T U \geq 0, R \geq 0) \quad (2)$$

を行い、方程式:

$$U^T A U^{-1} \tilde{x} = U^T b, \quad \text{但し } U^{-1} \tilde{x} = x \quad (3)$$

に対して CG 法を適用するものである。ここで、 $U^T A U^{-1}$ は対称正定値になる。その考え方で、 \tilde{x} についての CG 法の算法を作った後、同じことを x について施すという具合に書き直したものは次のようになる:

$$\begin{array}{l}
 \text{100 } \left\{ \begin{array}{l}
 \text{X。 (初期解) を用意; } \quad \gamma = b - AX \quad ; \quad p = \gamma \\
 C = (p, r) \quad ; \quad \text{eps} = \text{eps} * \|b\| \\
 \text{continue} \\
 q = Ap \quad ; \quad \alpha = C / (p, q) \\
 x = x + \alpha p \quad ; \quad r = r - \alpha q \\
 r' = (U^T U)^{-1} r \quad ; \quad d = (r', r) \quad ; \quad \beta = d / C \\
 p = r' + \beta p \quad ; \quad C = d \\
 \text{if } \|r\| > \text{eps go to 100}
 \end{array} \right. \quad (4)
 \end{array}$$

周知の通り、CG法は次の事実に基づいている：

「方程式 (1) の解 x は函数：

$$f(x) = (Ax, x) - 2(b, x)$$

を最小にする x である。即ち、残差 $r = b - Ax$ に対して

$$\begin{aligned}
 (r, A^{-1}r) &= (b - Ax, A^{-1}(b - Ax)) \\
 &= (Ax, x) - 2(b, x) + (b, A^{-1}b)
 \end{aligned} \quad (5)$$

を最小化することである。また (1) の解 x の誤差について考えると、 x の誤差を $S = A^{-1}b - x$ とおくと、 $S = A^{-1}b - x = A^{-1}r$ ゆえ、(5) の最小化問題は、

$$(S, AS) = (A^{-1}r, r) = (Ax, x) - 2(b, x) + (b, A^{-1}b) \quad (6)$$

を最小化することと同じことである。いま A の固有値と固有ベクトルを

$$\begin{aligned}
 \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n \quad ; \\
 v_1, v_2, \dots, v_n \quad ; \quad \text{但し } v_i^T v_j = \delta_{ij}
 \end{aligned}$$

とすると、 $(r, A^{-1}r)$ と (S, AS) は次のように示される：

$$(r, A^{-1}r) = \sum_{i=1}^n \frac{r_i^2}{\lambda_i} v_i \quad (7)$$

$$(S, AS) = \sum_{i=1}^n \lambda_i S_i^2 v_i \quad (8)$$

(7)、(8) から見るように、方程式 (1) に対する CG法は、残差 r に関しては、係数行列 A の固有値の低次のモードを強く意識して最小化をはかり、誤差 S に関してはその高次のモードを強く意識して最小化をはかるのである。

また、CG法が早く収束するための条件は、係数行列 A の固有値 $\lambda(A)$ に重根固有値、あるいは密集固有値があると CG法の収束が速い。

$\lambda(A)$ に重根固有値があるとき、例えば $\lambda_1 = \lambda_2$ の場合、(1) の残差ベクトル $\{r_k\}$; $k = 1, 2, \dots$ を次の式：

$$\begin{cases}
 r_0 = \sum_{i=1}^n C_i v_i & \text{(初期残差)} \\
 r_{k+1} = \sum_{i=1}^n C_i R_k(\lambda_i) v_i
 \end{cases} \quad (9)$$

で表示したとき、 r_k 中の v_1 及び v_2 の成分はいつも $C_1 : C_2 (C_1 R_k(\lambda_1) : C_2 R_k(\lambda_2)) = C_1 : C_2$ になるから r_k が許される空間の次元数は n にくらべて $n-1$ に落ちる。

密集団有値の場合、例えば、 $\lambda_1 = \lambda_2 = \lambda_3$ のときは T_1 が殆んどゼロになる。それは、CG法の算法から

$$\alpha_0 = \frac{(r_0, r_0)}{(r_0, Ar_0)} = \frac{\sum C_j^2}{\sum C_j^2 \lambda_j}$$

即ち、 α_0 は λ_j の重み C_j^2 付き平均の逆数ゆえ、 λ_j は殆んど α_0^{-1} に等しい。ここで

$$\lambda_j = \left(\frac{1}{\alpha_0}\right)(1 + \epsilon_j)$$

とおくと、

$$r_1 = r_0 - \alpha_0 Ar_0 = \sum C_j v_j (1 - \alpha_0 \lambda_j) = \sum C_j v_j \epsilon_j$$

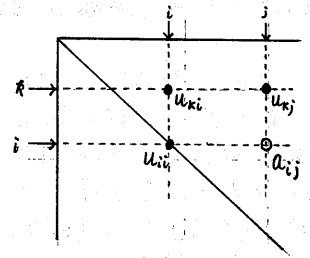
となり、殆んどゼロになる。

また、 $\lambda_1 = \lambda_2 < \lambda_3$ のとき、 T_2 が殆んどゼロになることも証明できる。これからCG法の収束の加速も明らかである。PCG法の誕生もこれに基づいている。

Meijerink 等の用いた不完全 $U^T U$ 分解は、 $d_i = u_{ii}^{-1}$ となるような次のような $U^T D U$ タイプのものである。

要素 (i, j) ; $i, j = 1, 2, \dots, n$ の部分集合 G を考え、 G 上の A の不完全コレスキー分解: $A = U^T D U - R$ を次のように定義する:

$$\begin{array}{l} \text{do } i = 1, n \\ \quad u_{ii} = a_{ii} - \sum_{k=1}^{i-1} u_{ki}^2 d_k ; d_i = u_{ii}^{-1} \\ \quad \text{do } j = i+1, n \\ \quad \quad \text{if } (i, j) \in G \text{ then} \\ \quad \quad \quad u_{ij} = a_{ij} - \sum_{k=1}^{i-1} u_{ki} u_{kj} d_k \\ \quad \quad \quad \text{else} \\ \quad \quad \quad u_{ij} = 0 \end{array}$$



(10)

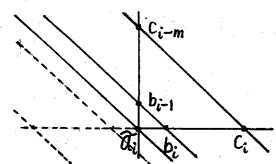
(図 1)

G が全集合のとき、これは周知のコレスキー分解になる。 G としては A の非ゼロ集合: $\{(i, j); a_{ij} \neq 0\}$ を含むものが使われる。特に $G = \{(i, j); a_{ij} \neq 0\}$ としたものが好んで使われる。

以下、二次元問題に対する五点差分用の不完全 $U^T D U$ 分解の例を示す。

a) ICCG(1,1) 用: $G = \{(i, j); a_{ij} \neq 0\}$ にとつた最も簡単なもの。

このとき、 $A = U^T D U - R$ 中、 U の非対角部の要素は、もとの A の非対角部そのままを使ってよいことになる。 D^{-1} と U の対角部が等しいため、結局 U の対角要素 \tilde{a}_i と D の要素 \tilde{d}_i を次の式によって計算すれば十分である。



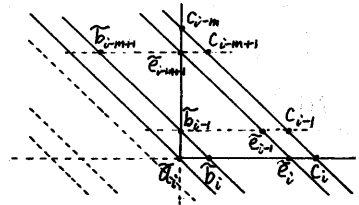
(図 2)

$$\begin{cases} \tilde{a}_i = a_i - b_{i-1}^2 \tilde{a}_{i-1} - c_{i-m}^2 \tilde{a}_{i-m} \\ \tilde{a}_i = \tilde{a}_i^{-1} \end{cases} \quad (11)$$

ここで、 a_i, b_i, c_i は、もとの行列 A の対角及び非対角部の第 i 行要素， m は A の半帯幅である。

b) ICCG (1,2) 用:

$$\begin{cases} \tilde{a}_i = a_i - \tilde{b}_{i-1}^2 \tilde{a}_{i-1} - c_{i-m}^2 \tilde{a}_{i-m} - \tilde{e}_{i-m+1}^2 \tilde{a}_{i-m+1} \\ \tilde{a}_i = \tilde{a}_i^{-1} \\ \tilde{b}_i = b_i - c_{i-m+1} \tilde{e}_{i-m+1} \tilde{a}_{i-m+1} \\ \tilde{e}_i = -c_{i-1} \tilde{b}_{i-1} \tilde{a}_{i-1} \end{cases} \quad (12)$$



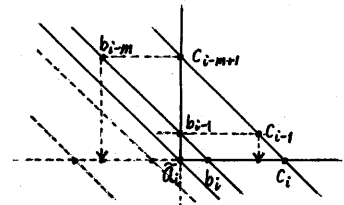
(図 3)

他に、ICCG (1,3)、ICCG (2,4) 等がある。二次元問題に対しては、メモリの容量さえ許せば ICCG (2,4) あたりが一番よいといわれる²⁾。しかし、三次元問題について考えると ICCG (1,1) 相当の ICCG (1,1,1) と ICCG (1,2) 相当のものぐらいまでが当面興味の対象になっている。

ma) MICCG (1,1) 用:

原論文³⁾では不完全 $U^T U$ 分解の形で記述してあるが、ここでは ICCG (1,1) 用のプログラムを改造する便宜を考えて $U^T D U$ 形、但し $d_i = U_{ii}^{-1}$ の形で記述する (後記の MICCG (1,2) についても同様)。

$\sigma = 1 + \epsilon$ ($\epsilon > 0$) にとり、 a_i, b_i, c_i を A のためのライン、 $\tilde{a}_i, \tilde{b}_i, \tilde{c}_i$ を U のためのライン、 \tilde{d}_i を D のためのラインとしたとき MICCG (1,1) は次のようになる:



(図 2)

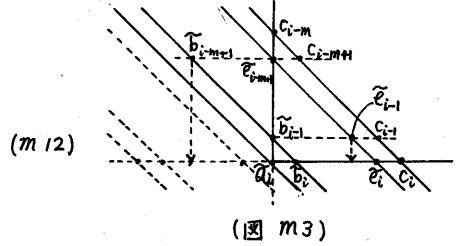
$$\begin{cases} \tilde{a}_i = \sigma a_i - b_{i-1}^2 \tilde{a}_{i-1} - c_{i-m}^2 \tilde{a}_{i-m} - \underline{c_{i-1} b_{i-1} \tilde{a}_{i-1}} - \underline{c_{i-m} b_{i-m} \tilde{a}_{i-m}} \\ \tilde{a}_i = \tilde{a}_i^{-1} \end{cases} \quad (m11)$$

ここに、 $\sigma = 1 + \epsilon$ 中 ϵ 値としては、内点方程式については $O(\epsilon^2)$ 、ノイマン境界上の方程式については $O(\epsilon)$ の程度で与ればよい (ϵ はあみ目の間隔)。下線部 $\underline{\quad}$ をした項が modify 項である。これを追加したとき、それだけでは、 $A_M = U_M^T D_M U_M - R_M$ が Varga の所謂正則分離に陥る危険がある (\tilde{a}_i が負になる危険がある)。それを避けるために、 a_i に ϵ ぐらいの補正を行う。それで $\tilde{a}_i > 0$ 、 $R_M \geq 0$ を保持することができる。

$\sigma = 1$ 、且つ下線部 $\underline{\quad}$ をした項 (modify 項) を消したら ICCG (1,1) になる。MICCG (1,1) の反復過程は ICCG (1,1) と全く同じである (以下 MICCG (1,2) に対しても同様)。

mb] MICCG (1, 2) 用:

$$\begin{cases} \tilde{a}_i = \sigma a_i - \tilde{b}_{i-1}^2 \tilde{a}_{i-1} - \tilde{e}_{i-m+1}^2 \tilde{a}_{i-m+1} - c_{i-m}^2 \tilde{a}_{i-m} \\ \quad - \tilde{b}_{i-1} \tilde{e}_{i-1} \tilde{a}_{i-1} - \tilde{b}_{i-m+1} \tilde{e}_{i-m+1} \tilde{a}_{i-m+1} \\ \tilde{a}_i = \tilde{a}_i^{-1} \\ \tilde{b}_i = b_i - c_{i-m+1} \tilde{e}_{i-m+1} \tilde{a}_{i-m+1} \\ \tilde{e}_i = -c_{i-1} \tilde{e}_{i-1} \tilde{a}_{i-1} \end{cases}$$



MICCG (1, 3)、MICCG (2, 4) 等になると補正項が益々増加するので厄介であるからここでは省略する。

3. ICCG/MICCG 法の動作特性と、初期残差の preconditioning

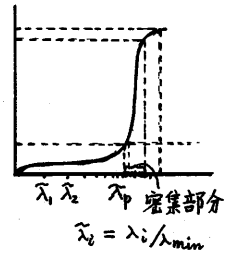
以下、多段メッシュによる ICCG/MICCG 法を ICGMG/MICGMG 法と命名する。この方法の考え方の焦点は、次の三点に基づいている。

a) $U^T A U^{-1}$ の固有値は、(図4) のように密集する。

b) $B = U^T A U^{-1}$ に対して CG 法を適用する。その際生成される残差ベクトルを $r_1, r_2, \dots, r_k, \dots$ とすると、前節の (9) 式:

$$\begin{cases} r_0 = \sum C_j V_j \\ r_{k+1} = \sum C_j R_k(\lambda_j) V_j \end{cases}$$

から見るように、(図4) で見る密集していない離散的な固有値 $\lambda_1 < \lambda_2 < \dots < \lambda_p$ に対応する C_1, C_2, \dots, C_p が $\neq 0$ になるような r_0 を選べば CG 法の収束が速くなる。



(図4)

c) このような初期残差 r_0 は、粗いあみ目 G_c による方程式:

$$A^c x^c = b^c \quad (c \text{ は coarse の } c) \quad (13)$$

の解 x^c から prolongation (interpolation) によって、精いあみ目 G_f 上の初期解 x^f を作り、そのときの初期残差を

$$r^f = b^f - A^f x^f \quad (f \text{ は fine の } f) \quad (14)$$

とすれば予想の目的が達成できるのである。

まず a) について考えると、(1) の係数行列 A に近い M 行列 M を使って $M^{-1} A x = M^{-1} b$ を作ると、 $M^{-1} A$ の固有値は密集する。しかし、これでは M を対称正定値にとっても、 $M^{-1} A$ が対称にもならないから CG 法の適用は不可能である。そこで、 A を (2) のように不完全 $U^T U$ 分解: $A = U^T U - R$ にしたあと $U^T U$ を M として起用すると、 $U^T A U^{-1}$ は対称正定値になり、またそれを方向ベクトル p_k に施す演算: $q_k = U^T A U^{-1} p_k$ も楽になるので、CG 法の適用も比較的煩調

に行なうことができる。

b) で、 $B = U^{-T}AU^{-1}$ に対して CG 法を適用する際、各反復で “ $(Y_{k+1}, B^{-1}Y_{k+1})$ ” を

アフィン空間： $\alpha_0 + \text{span}(r_0, r_1, \dots, r_k)$

の上で最小にならしめるよう多項式 $R_k(B)$ を生成しながら進行する。”

ところで、

$$\begin{aligned} (Y_{k+1}, A^{-1}Y_{k+1}) &= (Y_k - \alpha_k AP_k, A^{-1}Y_k - \alpha_k P_k) \\ &= (Y_k, A^{-1}Y_k) - 2\alpha_k (Y_k, P_k) + \alpha_k^2 (P_k, AP_k) \\ &= (Y_k, A^{-1}Y_k) - \frac{(Y_k, P_k)^2}{(P_k, AP_k)} \quad (\because \alpha_k = \frac{(Y_k, P_k)}{(P_k, AP_k)}) \end{aligned}$$

しかし、周知のとおり $(Y_k, P_k) = (Y_k, Y_k)$ 。また $(P_k, AP_k) \leq (Y_k, AY_k)$ 。それは次のように示される：

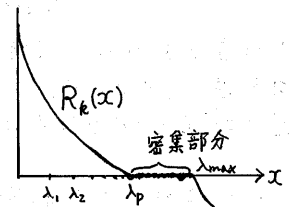
$$\begin{aligned} (P_k, AP_k) &= (Y_k + \beta_{k-1} P_{k-1}, AY_k + \beta_{k-1} AP_{k-1}) \\ &= (Y_k, AY_k) + 2\beta_{k-1} (P_{k-1}, AY_k) + \beta_{k-1}^2 (P_{k-1}, AP_{k-1}) \\ &= (P_{k-1}, AP_{k-1}) \left[\beta_{k-1}^2 + 2 \frac{(P_{k-1}, AY_k)}{(P_{k-1}, AP_{k-1})} \beta_{k-1} + \frac{(Y_k, AY_k)}{(P_{k-1}, AP_{k-1})} \right] \\ &= (P_{k-1}, AP_{k-1}) \left[\left(\beta_{k-1} + \frac{(P_{k-1}, AY_k)}{(P_{k-1}, AP_{k-1})} \right)^2 + \frac{(Y_k, AY_k)}{(P_{k-1}, AP_{k-1})} - \left(\frac{(P_{k-1}, AY_k)}{(P_{k-1}, AP_{k-1})} \right)^2 \right] \end{aligned}$$

β_{k-1} は上式を最小にするよう決めためえ

$$\beta_{k-1} = - \frac{(P_{k-1}, AY_k)}{(P_{k-1}, AP_{k-1})}$$

$$\begin{aligned} \therefore (P_k, AP_k) &= (P_{k-1}, AP_{k-1}) \left[\frac{(Y_k, AY_k)}{(P_{k-1}, AP_{k-1})} - \left(\frac{(P_{k-1}, AY_k)}{(P_{k-1}, AP_{k-1})} \right)^2 \right] \\ &= (Y_k, AY_k) - \frac{(P_{k-1}, AY_k)^2}{(P_{k-1}, AP_{k-1})} \end{aligned}$$

$$\therefore (P_k, AP_k) \leq (Y_k, AY_k)$$



(図 5)

従って、前の式は

$$\begin{aligned} \frac{(Y_{k+1}, A^{-1}Y_{k+1})}{(Y_k, A^{-1}Y_k)} &= 1 - \frac{(Y_k, Y_k)}{(Y_k, A^{-1}Y_k)} \cdot \frac{(Y_k, Y_k)}{(P_k, AP_k)} \\ &\leq 1 - \frac{(Y_k, Y_k)}{(Y_k, A^{-1}Y_k)} \cdot \frac{(Y_k, Y_k)}{(Y_k, AY_k)} = 1 - \frac{\sum c_j^2}{\sum c_j^2 \lambda_j^{-1}} \cdot \frac{\sum c_j^2}{\sum c_j \lambda_j} \\ &\leq 1 - \frac{\lambda_p}{\lambda_{\max}} \end{aligned}$$

以上の事実から見られるように、 λ_{p+1}, \dots の近くを x 軸と交わる (図 5) のような多項式 $R_k(x)$ が生成されていく。そこで、 c_1, c_2, \dots, c_p があるときとくらべてそれがゼロに近いような λ を選んでうまくやって行けばその分だけ少い反復回数で速く収束するはずである。

c) で、粗いあみ目 G_C 上の解 x^C から精いあみ目 G_f 上の初期解 x^f を作る時、Prolongation 作用素 P_k を如何に取るかは問題に存っているが、ここで

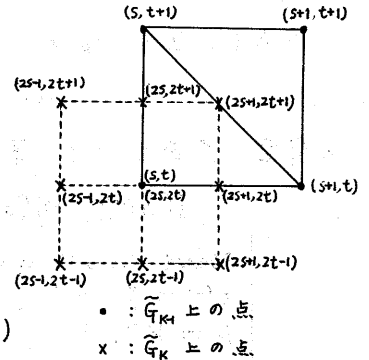
は二次元の九点差分を使った。それは(図6)による次のようなものである。

粗いあみ目 G_{k-1} 上の解ベクトル $x^{(k-1)}$ の集合を \tilde{G}_{k-1} とし、Prolongation 作用素を

$$P_k: \tilde{G}_{k-1} \rightarrow \tilde{G}_k$$

としたとき、 q -point prolongation は:

$$\begin{cases} x_{2s,2t}^{(k)} = (P_k x^{(k-1)})_{2s,2t} = x_{s,t}^{(k-1)} \\ x_{2s+1,2t}^{(k)} = (P_k x^{(k-1)})_{2s+1,2t} = \frac{1}{2}(x_{s,t}^{(k-1)} + x_{s+1,t}^{(k-1)}) \\ x_{2s,2t+1}^{(k)} = (P_k x^{(k-1)})_{2s,2t+1} = \frac{1}{2}(x_{s,t}^{(k-1)} + x_{s,t+1}^{(k-1)}) \\ x_{2s+1,2t+1}^{(k)} = (P_k x^{(k-1)})_{2s+1,2t+1} = \frac{1}{4}(x_{s,t}^{(k-1)} + x_{s+1,t}^{(k-1)} + x_{s,t+1}^{(k-1)} + x_{s+1,t+1}^{(k-1)}) \end{cases} \quad (15)$$



(図6)

慣例の Multigrid 法の場合、Prolongation 作用素 P_k は、その逆作用素 Restriction 作用素

$$R_k: \tilde{G}_k \rightarrow \tilde{G}_{k-1}$$

と適用しなければならぬ。近頃一般的には、Prolongation P_k ($k=2,3,\dots,l$) が決めたら、あとは自動的に

$$R_k = (P_k)^T; \quad A_{k-1} = R_k A_k P_k \quad (k=l, l-1, \dots, 2)$$

とするのがよいという意見が強くなっている。但し A_{k-1} の方はそれぞれのあみ目 G_{k-1} から直接離散化することにし、 $R_k = (P_k)^T$ だけ守ることもあるという。

q -point prolongation に対応する q -point restriction R_k は:

$$\begin{aligned} x_{s,t}^{(k-1)} = (R_k x^{(k)})_{s,t} &= \frac{1}{4}x_{2s,2t}^{(k)} + \frac{1}{8}(x_{2s+1,2t}^{(k)} + x_{2s-1,2t}^{(k)} + x_{2s,2t+1}^{(k)} + x_{2s,2t-1}^{(k)}) \\ &+ \frac{1}{16}(x_{2s+1,2t+1}^{(k)} + x_{2s-1,2t+1}^{(k)} + x_{2s+1,2t-1}^{(k)} + x_{2s-1,2t-1}^{(k)}) \end{aligned} \quad (16)$$

(15)と(16)は互いに適合する。但し、固定境界の隣り及びその他特別な場合にはそれなりの注意が要る。今度の場合は、慣例の Multigrid とはちがって、“粗”から“精”への一方向的な進み方ゆえ、さらに“粗”へ戻る操作は不必要だから Restriction 作用素 R_k は使っていない。

4. テスト用プログラムと数値例

次に、三段メッシュによる ICGMG/MICGMG 法の基本的手順とその算法及び数値例の結果に対して概要的に述べる。

まず、あみ目の系列: G_1, G_2, G_3 (G_1 :粗; G_2 :中間; G_3 :精) を用意し、各あみ目 G_k ($k=1, 2, 3$) 上の方程式を

$$A_k x^{(k)} = f^{(k)}, \quad k=1, 2, 3$$

とする。 G_k 上の解ベクトル $x^{(k)}$ の集合を \tilde{G}_k とし、Prolongation 作用素 P_k

$$P_k: \tilde{G}_{k-1} \mapsto \tilde{G}_k$$

を用意し、また G_k 上、 A_k の不完全 $U_k^T D_k U_k$ 分解:

$$A_k = K_k - R_k; \quad K_k = U_k^T D_k U_k$$

を用意したあと、次の算法によってプログラムを作成する:

ICGMG/MICGMG 算法

```

solve  $A_1 x^{(1)} = f^{(1)}$ 
 $x^{(2)} = P_2 x^{(1)}$ ;  $r^{(2)} = f^{(2)} - A_2 x^{(2)}$ ;  $res\ 20 = \|r^{(2)}\| / \|f^{(2)}\|$ 
 $p = K_2^{-1} r^{(2)}$ ;  $c = (p, r^{(2)})$ 
write res 20
do nc = 1, n2
     $q = A_2 p$ ;  $a = c / (p, q)$ 
     $x^{(2)} = x^{(2)} + ap$ ;  $r^{(2)} = r^{(2)} - aq$ 
     $r = K_2^{-1} r^{(2)}$ ;  $d = (r, r^{(2)})$ 
     $b = d / c$ ;  $c = d$ 
     $p = r + bp$ 
res 2 =  $\|r^{(2)}\| / \|f^{(2)}\|$ 
write res 2
 $x^{(3)} = P_3 x^{(2)}$ ;  $r^{(3)} = f^{(3)} - A_3 x^{(3)}$ ;  $res\ 30 = \|r^{(3)}\| / \|f^{(3)}\|$ 
write res 30
 $p = K_3^{-1} r^{(3)}$ ;  $c = (p, r^{(3)})$ ;  $nc = 0$ ;  $eps\ 3 = eps * \|f^{(3)}\|$ 
100 Continue
     $q = A_3 p$ ;  $a = c / (p, q)$ 
     $x^{(3)} = x^{(3)} + ap$ ;  $r^{(3)} = r^{(3)} - aq$ 
     $r = K_3^{-1} r^{(3)}$ ;  $d = (r, r^{(3)})$ 
     $b = d / c$ ;  $c = d$ 
     $p = r + bp$ ;  $nc = nc + 1$ 
    if  $\|r^{(3)}\| > eps\ 3$  goto 100
res 3 =  $\|r^{(3)}\| / \|f^{(3)}\|$ 
write nc, res 3
write  $x^{(3)}$ 

```

ここで、 $n2$ は固定してやっているが、これは適当にとってよい。しかし、あんまり大きく取る必要はない。普通、一番粗いあみ目 G_1 のサイズの平方根ぐらいの程度を基準として取るのがよいと思う。

数値テスト例.

数値テストは(図7)のよう反四つのタイプのモデルに対して、異なる丸値の次の問題について試みて見た:

$$\begin{cases} \operatorname{div}(-k(x)\operatorname{grad} u) = f(x) \\ u|_{\Gamma_1} = 0; (k \cdot \operatorname{grad} u) \cdot n|_{\Gamma_2} = 0 \end{cases}$$

あみ目のサイズは、 $G_3 = 40 \times 40, 80 \times 80$ に取ってやってみた。テストの結果を見ると反復回数はいつも

$$\text{MICCG} < \text{ICCG}$$

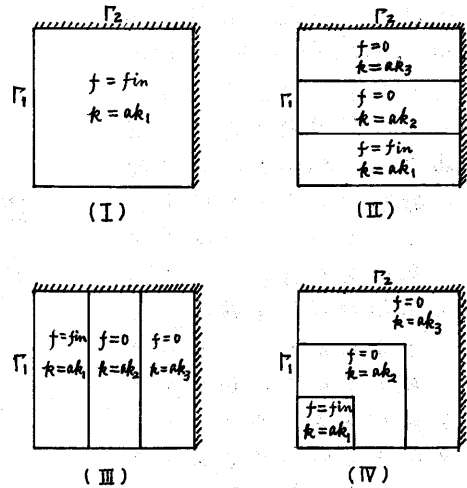
となっている。ICGMG/MICGMG と比べて見ると反復回数は常に

$$\text{ICGMG} < \text{ICCG}; \text{MICGMG} < \text{MICCG}$$

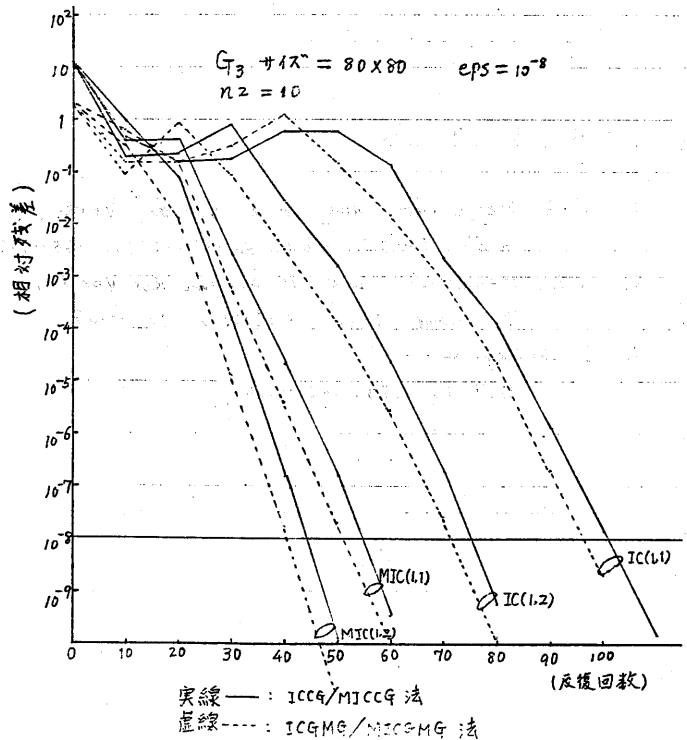
を保持している。

その他、各レベルについて調べてみると、 $\text{ICGMG}(1,x)/\text{MICGMG}(1,x)$ の傾向は、もとの $\text{ICCG}(1,x)/\text{MICCG}(1,x)$ の傾向に追随されている。各タイプのモデルに対して考察すると、タイプ(I)が一番収束がよい。タイプ(II)とタイプ(III)は殆んど同様である。タイプ(IV)の場合が少し振動があらわれるが、これらは皆固有値の分布と関係があるようにみえる。

(図8)に、タイプ(IV)の部分的结果をグラフで示した。(図9)に ICCG/MICCG の、タイプ(I)の情况についてあらわした。図で見ると、タイプ(IV)とくらべてみれば、その収束は割りによくなっている



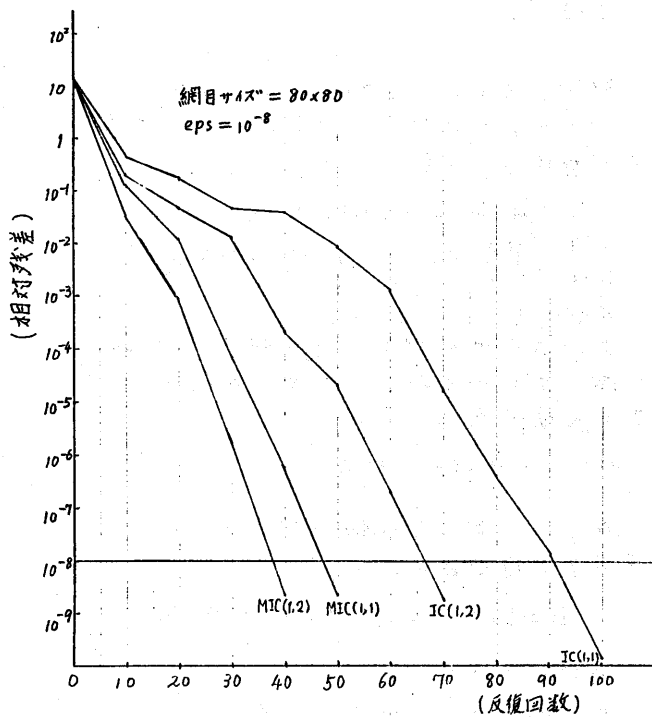
(図7)



(図8)

5. おわりに

はじめに言ったように、移流項のない線形問題の場合は、ICCG/MICCG法だけでもかなりよい結果が得られるが、移流項のある問題の場合は、粗いあみ目で普通に離散化したのではM行列にならないうことが起るから、粗いあみ目に対しては‘人工粘性項’を粗さに応じて十分利かせて、セルベクレ数を1以下に抑えてやる必要がある。G₁としては直接法で解いても主メモリに入れる程度の粗さのあみ目を取ればうまく行けると思う。



(図9)

参 考 文 献

1. J. A. Meijerink and H. A. van der Vorst, *Math. Comp.*, vol 31 No. 137 (1977), 148-162.
2. J. A. Meijerink and H. A. van der Vorst, *J. Comp. Phys.* 44(1981), 134-155
3. I. Gustafsson, *BIT* 18(1978) 142-156