

有限要素法による連立一次方程式のベクトル・並列向き反復解法

日立製作所 ソフトウェア開発本部 後 保範(Yasunori Ushiro)

1. はじめに

有限要素法で発生する大次元疎行列の連立一次方程式は、不完全LU分解付き共役勾配法系解法で高速に計算できる。しかし、この方法は一台のベクトル計算機には効率良く適合するよう工夫できるが、複数台のベクトル計算機では効率が悪い。そこで共役勾配法系の前処理方式として、不完全LU分解方式に代えて行列乗算方式を検討した。行列乗算方式を採用すると、常に次元数Nの並列度が得られて、複数台のベクトル計算機に良く適合する。しかし行列乗算方式の前処理では、収束が遅く、安定性も悪いという欠点がある。本報告では、行列乗算方式の前処理で、収束の改善と安定性の確保のため工夫した点について述べる。数値実験はBi-CGSTAB法とBCG法に前処理を付けて行なった。

2. 前処理付きBi-CGSTAB法とBCG法

(1) 前処理付きBi-CGSTAB法

(r, r)は内積を示し、前処理行列Mは行列の積の場合と(LU)⁻¹と陰形式になる場合がある。連立一次方程式Ax = bの解xを反復解法で求める。

初期値 x を与える

行列 A から前処理行列 M を作成

$$q = M \cdot b \quad , \quad b_n = (q, q)$$

$$p = r \quad r_0 = r = q - M \cdot A \cdot x$$

err ≤ 10⁻⁸ or NI 回まで反復計算

$$q = M \cdot A \cdot p$$

$$\sigma = (r_0, q) \quad , \quad \alpha = c / \sigma$$

$$e = r - \alpha q \quad , \quad v = M \cdot A \cdot e$$

$$\mu = (e, v) / (v, v)$$

$$x = x + \alpha p + \mu e$$

$$r = e - \mu v$$

$$r_n = (r, r) \quad , \quad c = (r_0, r)$$

$$err = \sqrt{r_n / b_n} \quad , \quad \beta = c / (\mu \sigma)$$

$$p = r + \beta (p - \mu q)$$

注)

$$|c| \leq r_n \times 10^{-12}$$

となると一回だけ

$$r_0 = r \quad , \quad c = r_n$$

とリセットする

2)

(2) 前処理付きBCG法

連立一次方程式 $Ax = b$ の解 x を反復解法で求める。初期値 x を与える行列 A から前処理行列 M を作成

$$p^* = p = r^* = r = M(b - Ax)$$

$$c1 = (r, r^*)$$

 $err \leq 10^{-8}$ or NI 回まで反復計算

$$q = M \cdot A \cdot p, \quad q^* = A^T \cdot M^T \cdot p^*$$

$$\alpha = c1 / (q, p^*), \quad x = x + \alpha p$$

$$r = r - \alpha q, \quad r^* = r^* - \alpha q^*$$

$$rn = (r, r), \quad c2 = (r, r^*)$$

$$err = \sqrt{rn/bn}$$

$$\beta = c2 / c1, \quad c1 = c2$$

$$p = r + \beta p, \quad p^* = r^* + \beta p^*$$

3. 提案する前処理方式

3)

不完全LU分解を使用したベクトル計算機向き前処理方式は、既に発表されているのでここでは省略する。

提案方式の係数行列とリストの持ち方は次のようにする。

 N : 連立一次方程式の次元数 ND : 対角要素を除いた係数行列 A の一行当りの最大非ゼロ要素数 $D(N)$: 対角要素 d_i ($i=1, 2, \dots, N$) $B(N)$: 右辺ベクトル b_i ($i=1, 2, \dots, N$) $A(N, ND)$: 係数行列 A の対角を除いた非ゼロ要素 $L(N, ND)$: 対角要素を除いた非ゼロ要素の列番号リスト、 $A(N, ND)$ に対応

3.1 提案する方式の目的

- (1) 共役勾配法系反復解法のベクトル・並列向き前処理に適用する。
- (2) 有限要素法で作成した行列を対象にする。
- (3) 常に次元数 N の並列度を持つこと。
- (4) 不完全LU分解を前処理とする方式とほぼ同等の収束性を持つこと。
- (5) アイソパラメトリック二次要素にも適用できること。
- (6) CAD入力でも適用できるように、節点の番号付けで収束が変化しないこと。

3.2 提案する前処理方式

- (1) スケーリングベクトルの作成

$$U(i) = \sum_{j=1}^{ND} |A(i, j)| \quad (i=1, 2, \dots, N)$$

$$W(i) = \frac{1 + \alpha}{\text{MAX}(U(i), D(i)) + \alpha D(i)}$$

今回は $\alpha = 4.0$ に固定した、但し $D(i) \geq 0$ としておく。

(2) 係数行列と右辺のスケーリング

$$D(i) = D(i) * W(i)$$

$$B(i) = B(i) * W(i) \quad (i=1, 2, \dots, N)$$

$$A(i, j) = A(i, j) * W(i) \quad (j=1, 2, \dots, ND)$$

(3) 非対角要素のソート

$$|A(i, j)| \geq |A(i, j+1)| \quad (i=1, 2, \dots, N, j=1, 2, \dots, ND-1)$$

対応して列番号リスト $L(i, j)$ も変更する。

(4) ソートした非対角行列 $A(N, ND)$ の分割

$$A1 \equiv A(N, 1) \quad , \quad A2 \equiv A(N, 2)$$

$$A3 \equiv A(N, 3) \quad , \quad A4 \equiv A(N, 4)$$

$$A5 \equiv A(N, 5) + A(N, 6)$$

$$A6 \equiv A(N, 7) + A(N, 8)$$

$$A7 \equiv A(N, 9) + A(N, 10) + A(N, 11)$$

$$A8 \equiv A(N, 12) + A(N, 13) + A(N, 14) + A(N, 15)$$

$$A9 \equiv A(N, 16) + A(N, 17) + \dots + A(N, 20)$$

$$A10 \equiv A(N, 21) + A(N, 22) + \dots + A(N, 26)$$

$$A11 \equiv A(N, 27) + A(N, 28) + \dots + A(N, 34)$$

.....

(5) 前処理行列 M の作成

$ND = 30$ の例を示す

$A1, A2, \dots, A10$ は上記で作成

$A11$ を次のように作成

$$A11 \equiv A(N, 27) + A(N, 28) + A(N, 29) + A(N, 30)$$

以上より前処理行列 M を作成する。

$$M \equiv (I - A11) \cdot \dots \cdot (I - A2)(I - A1)$$

ここで、 I は単位行列である。

ND の値の大小にかかわらず前処理行列 M は上記のように行列の乗算で表現する。

4. テスト問題

各種の問題で数値実験をするため、3次元の移流拡散問題、2次元及び3次元の固体と流体を同時に扱う問題及び曲線形状の2次元拡散問題をテスト問題に選定した。更に、一次要素と二次要素の使用によりテスト問題のケース分けをした。

ケースA : 3次元の移流拡散問題で六面体一次要素を使用

ケースB1 : 2次元の固体を含む移流拡散問題で三辺形一次要素を使用

ケースB2 : 2次元の固体を含む移流拡散問題で四辺形一次要素を使用

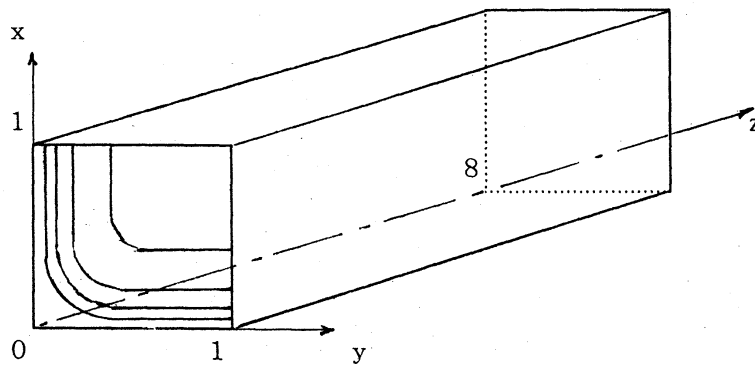
ケースC1 : 3次元の固体を含む移流拡散問題で六面体一次要素を使用

ケースC2 : 3次元の固体を含む移流拡散問題で六面体二次要素を使用

ケースD1 : 曲線形状の2次元拡散問題で四辺形一次要素を使用

ケースD2 : 曲線形状の2次元拡散問題で四辺形二次要素を使用

4.1 3次元の移流拡散問題 (ケースA)



$$-\operatorname{div}(k \cdot \nabla \phi) + V \cdot \nabla \phi = f, \quad V = (0, 0, w)$$

$$k = 1, \quad f = 100$$

$$w = w_0 \cdot \max[0, 1 - (1-x)^5 - (1-y)^5]$$

$$\text{境界条件} \begin{cases} \phi = 0 & \text{on } x=y=z=0 \text{ の } 3 \text{ 面} \\ (\nabla \phi) \cdot n = 0 & \text{on } x=y=1, z=8 \text{ の } 3 \text{ 面} \end{cases}$$

六面体一次要素 ($N=18081$) を使用, $20 \times 20 \times 40$ 分割

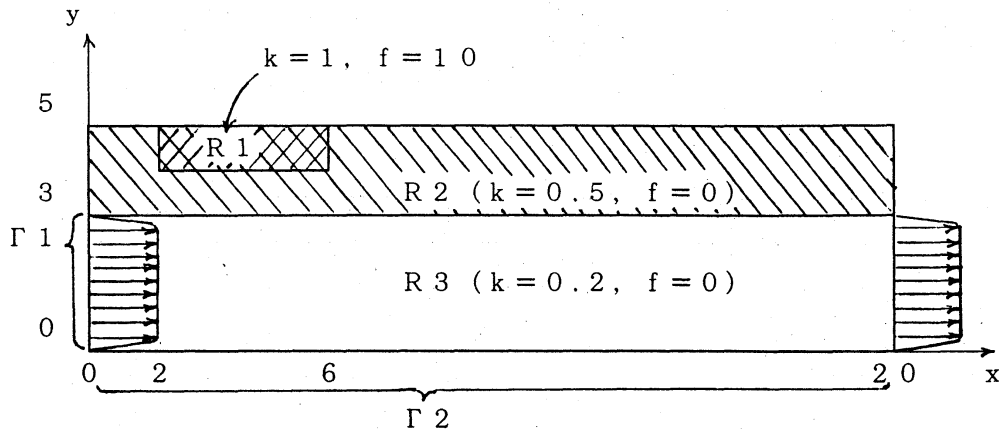
$$\text{セル・ペクレ数} = \Delta x \cdot w_0 / k = w_0 / 20$$

w_0 を変えて種々のセル・ペクレ数で数値実験した。

節点の番号付けは x, y, z の順に付けた。

本問題は村田健郎他著「大型数値シミュレーション」²⁾ p132のモデルを z の寸法だけ変更して使用した。

4.2 2次元の固体を含む移流拡散問題 (ケースB)



$$-\operatorname{div}(k \cdot \nabla \phi) + V \cdot \nabla \phi = f, \quad V = (u, 0)$$

$$u = u_0 [1.5^4 - (1.5 - y)^4] / 1.5^4$$

$$\text{境界条件} \begin{cases} \phi = 0 & \text{on } \Gamma 1 \\ (k \cdot \nabla \phi) \cdot n = -\phi & \text{on } \Gamma 2 \\ (k \cdot \nabla \phi) \cdot n = 0 & \text{on 其他} \end{cases}$$

$N = 20301$ (200 × 100 分割)

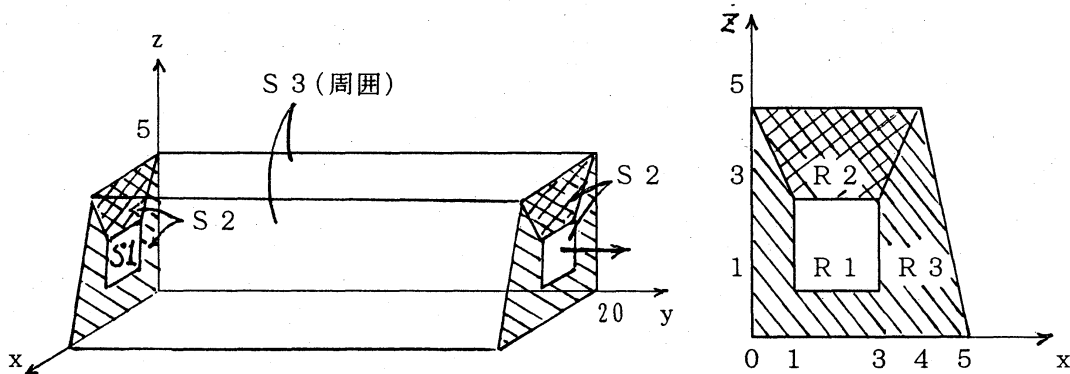
セル・ペクレ数 = $\Delta y \cdot u_0 / k = u_0 / 4$

ケースB1: 三辺形一次要素で分割

ケースB2: 四辺形一次要素で分割

節点の番号付けはCADを使用した。

4.3 3次元の固体を含む移流拡散問題 (ケースC)



$$-\operatorname{div}(k \cdot \nabla \phi) + C_p \cdot V \cdot \nabla \phi = f, \quad V = (0, v, 0)$$

$$\text{境界条件} \begin{cases} \phi = 20 & \text{on } S1 (\text{流入口}) \\ (k \cdot \nabla \phi) \cdot n = 0 & \text{on } S2 \\ (k \cdot \nabla \phi) \cdot n = -2(\phi - 20) & \text{on } S3 (\text{周囲}) \end{cases}$$

$$v = v_0 \cdot \min [1 - (x - 2)^4, 1 - (z - 2)^4]$$

領域	k	C _p	v ₀	f
R1	0.2	α	4	0
R2	0.5	0.2	0	50
R3	1.0	0.3	0	0

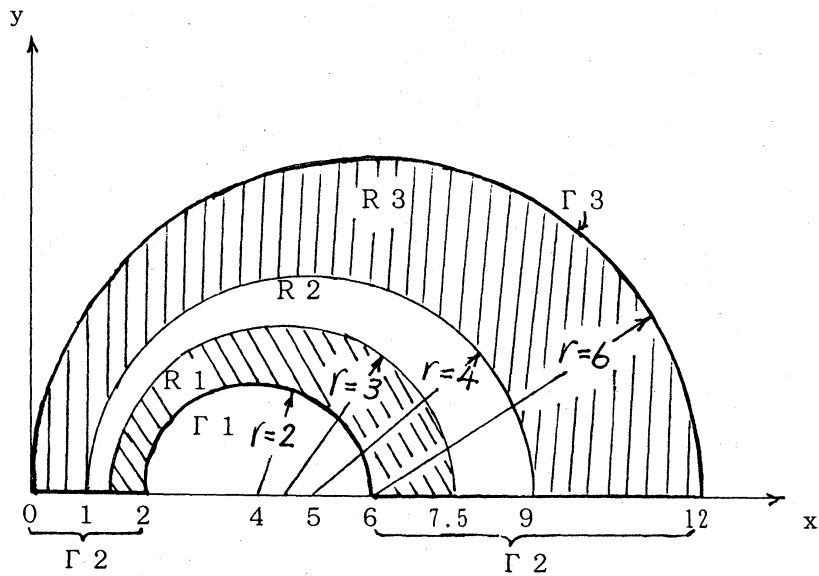
$$\text{セル・ペクレ数} = \alpha \cdot \Delta x \cdot v_0 / k = 20 \cdot \alpha \cdot \Delta x$$

ケースC1：六面体一次要素 ($\Delta x = 1/4$)

ケースC2：六面体二次要素 ($\Delta x = 1/3$)

節点の番号付けはCADを使用した。

4.4 曲線形状の2次元拡散問題 (ケースD)



$$-\operatorname{div}(k \cdot \nabla \phi) = 0$$

$$\text{境界条件} \begin{cases} \phi = 20 & \text{on } \Gamma1 \\ (k \cdot \nabla \phi) \cdot n = 0 & \text{on } \Gamma2 \\ (k \cdot \nabla \phi) \cdot n = -2(\phi - 20) & \text{on } \Gamma3 \end{cases}$$

k 領域	スカラ (1 : 2)	スカラ (1 : 100)	直交異方	一般異方
R 1	k = 1.0	k = 1.0	$k = \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix}$	$k = \begin{bmatrix} 1 & 0.3 \\ 0.2 & 0.5 \end{bmatrix}$
R 2	k = 0.5	k = 0.01	$k = \begin{bmatrix} 0.5 & 0 \\ 0 & 1 \end{bmatrix}$	$k = \begin{bmatrix} 0.5 & 0.2 \\ 0.3 & 1 \end{bmatrix}$
R 3	k = 1.0	k = 0.1	$k = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$	$k = \begin{bmatrix} 1 & 0.5 \\ 0.3 & 2 \end{bmatrix}$

ケースD1 : 四辺形一次要素分割 (N = 4636)

ケースD2 : 四辺形二次要素分割 (N = 7401)

節点の番号付けはCADを使用した。

5. 数値実験結果

数値実験に必要な連立一次方程式の作成はDEQSOL E2の有限要素版を使用した。数値計算はHITACS-820/80で行った。

5.1 比較対象とした反復解法

各テスト問題に下記5種類の反復解法を適用して、収束性及びCPU時間を測定しグラフで表示した。各図における反復解法は下記の記号で表示した。

(1) 記号：提案

提案する前処理方式付きBi-CGSTAB法。

(2) 記号：ILU

不完全LU分解付きBi-CGSTAB法。ベクトル計算機向きにプログラムしてある。

(3) 記号：I-E

係数行列AをA = I + Eと対角要素でスケーリングし、I-Eを前処理に使用したBi-CGSTAB法。

(4) 記号：I-P·E

係数行列AをA = P + P · Eと提案の方式でスケーリングし、I - P · Eを前処理に使用したBi-CGSTAB法。

(5) 記号：BCG

不完全LU分解付きBCG法。ベクトル計算機向きにプログラムしてある。

5.2 3次元の移流拡散問題の結果 (ケースA)

3次元の移流拡散問題で六面体双一次要素を使用し、セル・ペクレ数をゼロから16まで変えて収束するまでの反復回数を図1に示し、CPU時間を図2に示す。節点の番号付けは、CADに依存せずx, y, zの順に付けた。次元数N = 18081で20 × 20 × 40に分割した。解の収束判定は相対残差 ≤ 10⁻⁸で行った。一行当たりの対角を除いた非ゼロ要素数はND = 26である。

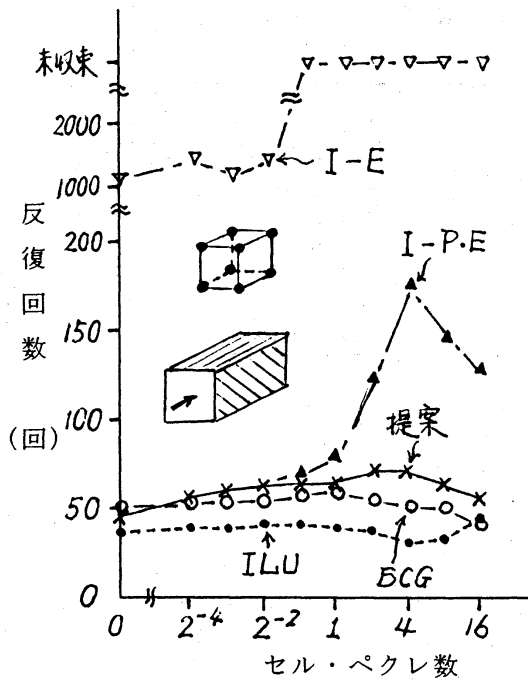


図1 FEM, ケースAの反復回数

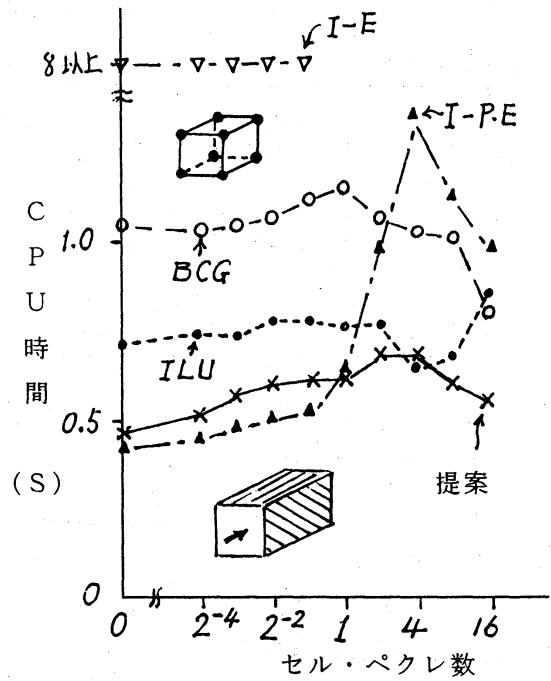


図2 FEM, ケースAのCPU時間

図1からILU(不完全LU分解付きBi-CGSTAB法)が反復回数では最も少なくて収束するのが分かる。しかし、図2からCPU時間では提案(提案する前処理方式付きBi-CGSTAB法)が最も高速であるのが分かる。本比較はHITACS-820モデル80一台で測定した値であり、並列・ベクトル計算機で測定すれば提案方式はILUに比較して更に、高速になると考えられる。

図3にセル・ペクレ数に対する各行の非対角要素の絶対値の和と対角要素の比の最大値を示した。この値が1でなく1.9以上の値のため、I-E(対角要素でスケリングしたBi-CGSTAB法)の収束が悪く、I-P-E(提案の方式でスケリングしたBi-CGSTAB法)の収束がセル・ペクレ数が大の場合を除き良くなっている。本有限要素法では風上分散化を使用していないので、セル・ペクレ数が8以上だと数値解が振動し、分散化の精度が問題になる。

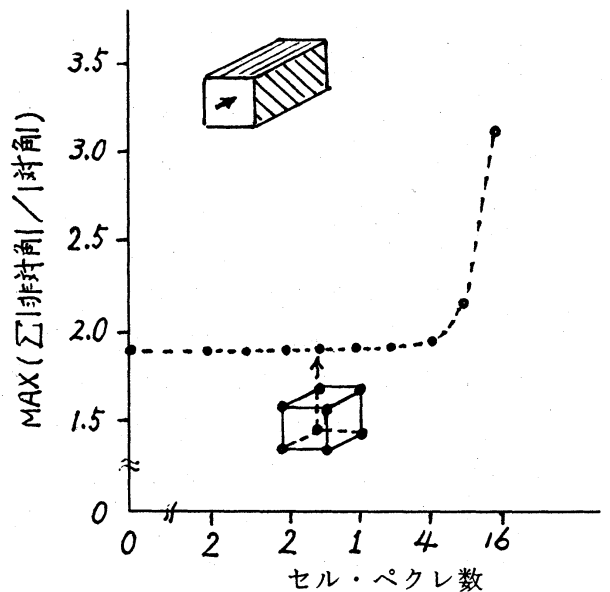


図3 FEM, ケースAの対角・非対角比

5.3 2次元の固体を含む移流拡散問題 (ケースB)

2次元の移流拡散問題で三辺形一次要素を使用し、セル・ペクレ数をゼロから8まで変えて収束するまでの反復回数を図4に示し、CPU時間を図5に示す。四辺形双一次要素を使用した反復回数を図6に示し、CPU時間を図7に示す。節点の番号付けはCADを使用した。次元数 $N=20301$ で、解の収束判定は相対残差 $\leq 10^{-8}$ で行った。一行当たりの対角を除いた非ゼロ要素数はケースB1で $ND=6$ 、ケースB2で $ND=8$ である。

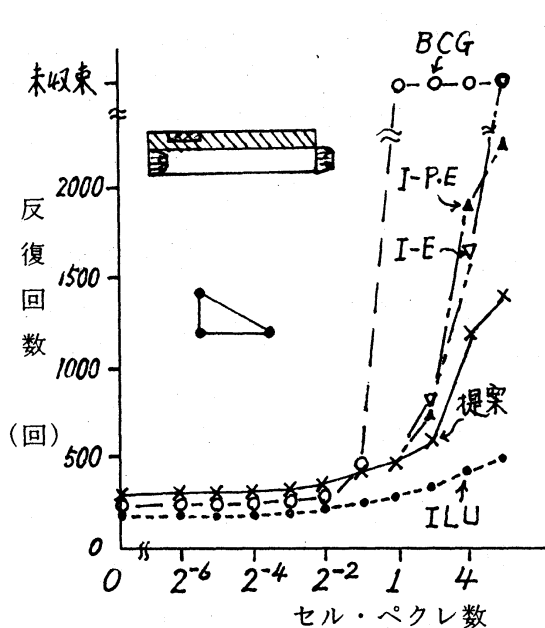


図4 FEM, ケースB1の反復回数

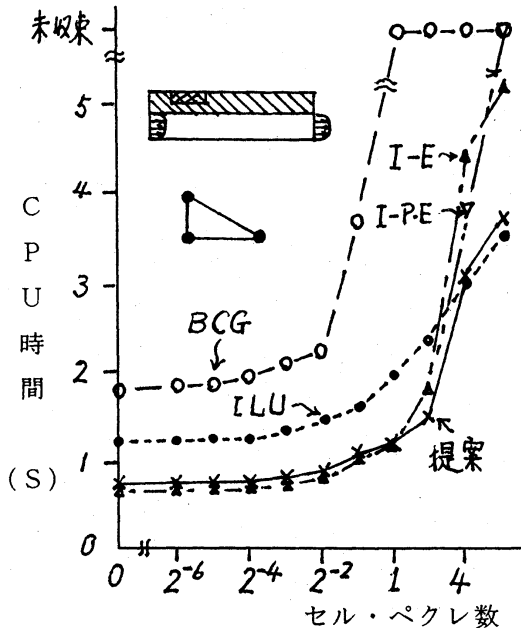


図5 FEM, ケースB1のCPU時間

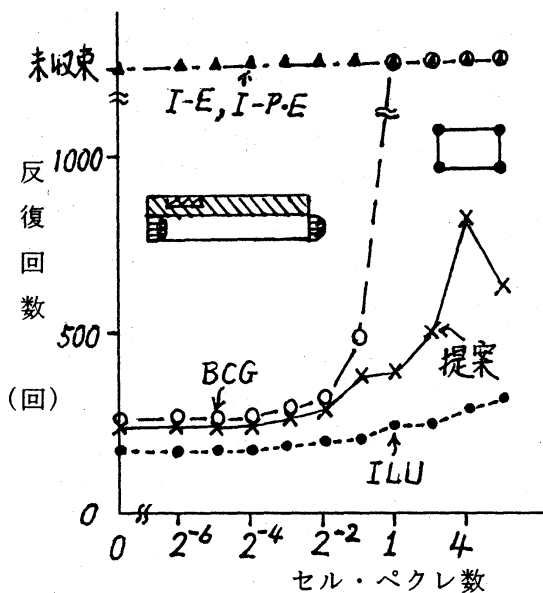


図6 FEM, ケースB2の反復回数

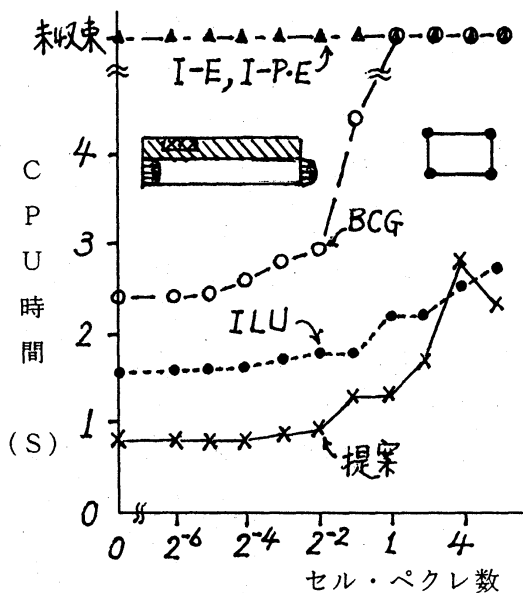


図7 FEM, ケースB2のCPU時間

図4からケースB1ではILUが最少の反復回数で収束するのが分かる。しかし、図5からCPU時間では提案が最も高速であるのが分かる。また、ケースB1ではI-E及びI-P・EもCPU時間では高速なのが見える。BCG(不完全LU分解付きBCG法)はセル・ペクレ数が1以上で収束しなくなる。これは、BCG法の直交性が崩壊するのが原因である。ケースB2(四辺形双一次要素)ではケースB1(三辺形一次要素)に比較して、I-E及びI-P・Eの収束が極端に悪くなるのが図6及び図7から分かる。その原因は対角要素対非対角要素の比の差と考えられる。セル・ペクレ数が1以下ではケースB1はその比が1であるのに対し、ケースB2はその比が1.4と大きい値となる。図6からケースB2ではILUが最少の反復回数で収束するのが分かる。しかし、図7からCPU時間では提案が最も高速であるのが分かる。ケースB2でも、BCGはセル・ペクレ数が1以上で収束しなくなる。これより、ケースB1、B2共に提案がCPU時間で高速なのが見える。本比較はHITACS-820/80一台で測定した値であり、並列・ベクトル計算機で測定すれば提案方式はILUに比較して更に、高速になると考えられる。

5.4 3次元の固体を含む移流拡散問題(ケースC)

3次元の移流拡散問題で六面体双一次要素を使用し、セル・ペクレ数を変えて収束するまでの反復回数を図8に示し、CPU時間を図9に示す。六面体アイソパメトリック二次要素を使用した反復回数を図10に示し、CPU時間を図11に示す。節点の番号付けはCADを使用した。解の収束判定は相対残差 $\leq 10^{-8}$ で行った。ケースC1(六面体双一次要素)では次元数 $N=5733$ で、一行当たりの対角を除いた非ゼロ要素数 $ND=26$ である。ケースC2(六面体アイソパメトリック二次要素)では $N=11624$ で $ND=80$ である。

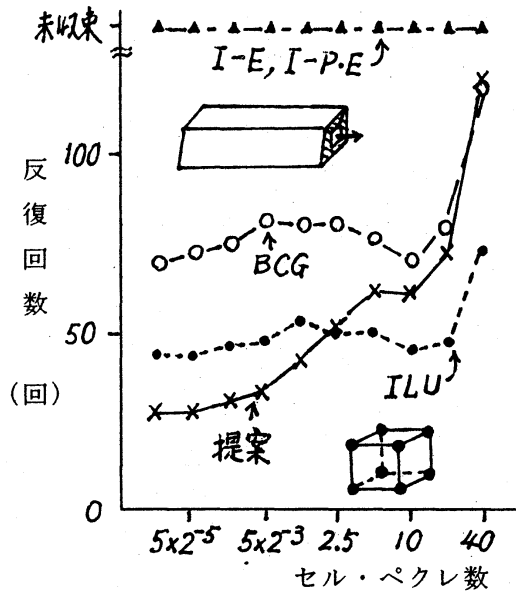


図8 FEM, ケースC1の反復回数

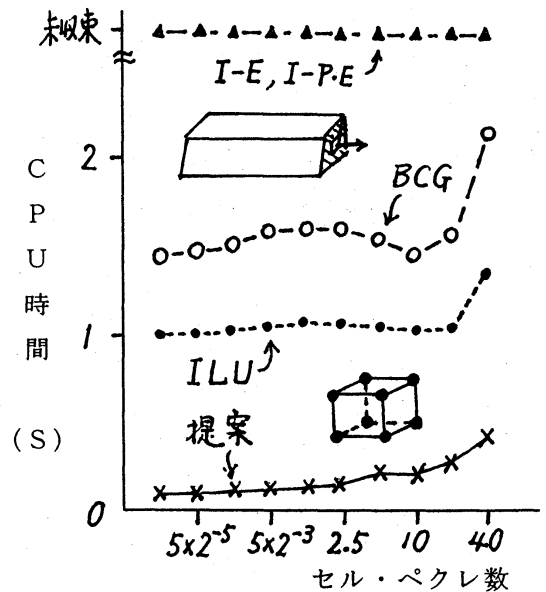


図9 FEM, ケースC1のCPU時間

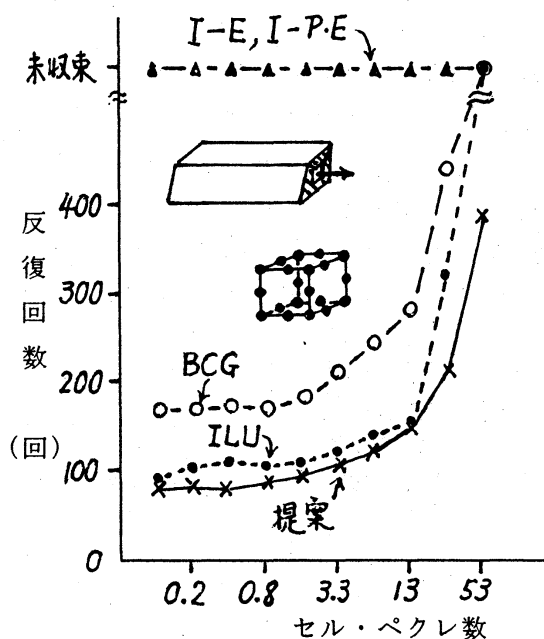


図10 FEM, ケースC2の反復回数

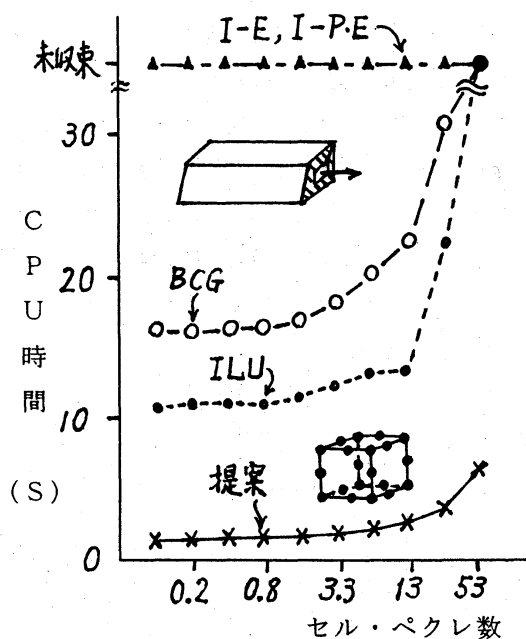


図11 FEM, ケースC2のCPU時間

図8からケースC1ではセル・ペクレ数が小さい範囲では、提案の方が反復回数が少なく、大きくなるとILUの方が反復回数が少なくて収束するのが分かる。図9からCPU時間では提案が最も高速で、次に高速なILUに比較し5~10倍も高速なのが分かる。図10からケースC2では提案が反復回数でも最少で収束するのが分かる。図11からCPU時間では提案が最も高速で、次に高速なILUに比較し約10倍も高速なのが分かる。これは、ケースC1, C2共にCADで節点の番号付けをしたため、ILU前処理付き解法の収束性が悪くなったのも一因である。これに対し、提案方式は収束性が節点の番号付けに依存せずCAD向きの解法である。

BCG(不完全LU分解付きBCG法)とILU(不完全LU分解付きBi-CGSTAB法)を比較すると、反復回数及びCPU時間共にILUの方が優れている。I-E及びI-P-Eは本テスト問題では共に収束しない。その原因は、本テスト問題では対角優位性が大幅に崩れているためと考えられる。対角要素対非対角要素の比がケースC1で3以上、ケースC2では5以上である。本比較はHITACS-820/80一台で測定した値であり、並列・ベクトル計算機で測定すれば提案方式はILUに比較し更に、高速になると考えられる。

5.5 曲線形状の2次元拡散問題 (ケースD)

2次元の拡散問題で四辺形双一次要素を使用し、種々の拡散係数に対して収束するまでの反復回数を図12に示し、CPU時間を図13に示す。四辺形アイソパラメトリック二次要素を使用した反復回数を図14に示し、CPU時間を図15に示す。節点の番号付けはCADを使用した。解の収束判定は相対残差 $\leq 10^{-8}$ で行った。ケースD1(四辺形双一次要素)では次元数 $N=4636$ で、非ゼロ要素数 $ND=8$ である。ケースD2(四辺形アイソパラメトリック二次要素)では $N=7401$ で、 $ND=20$ である。

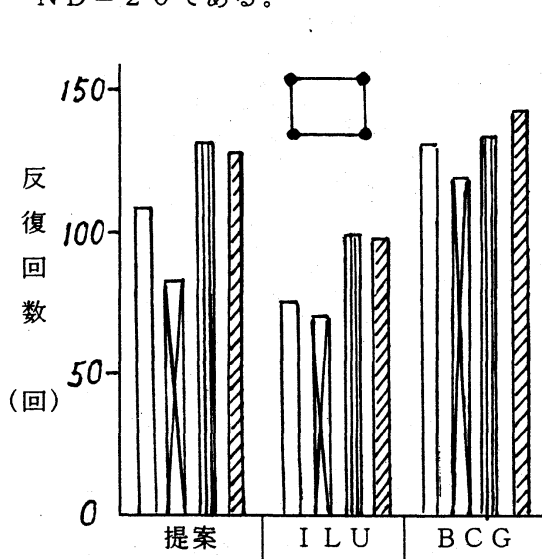


図12 FEM, ケースD1の反復回数

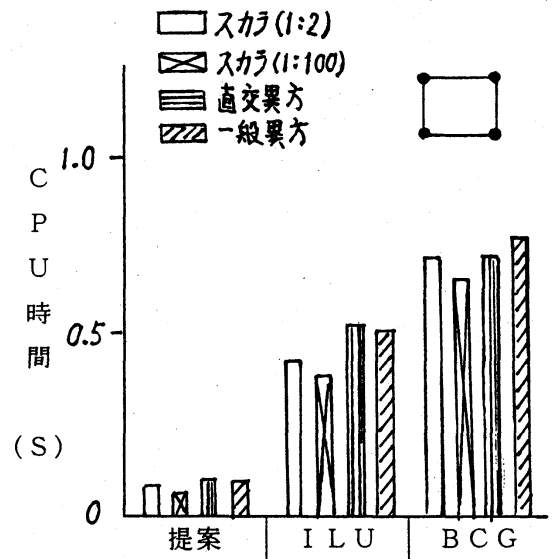


図13 FEM, ケースD1のCPU時間

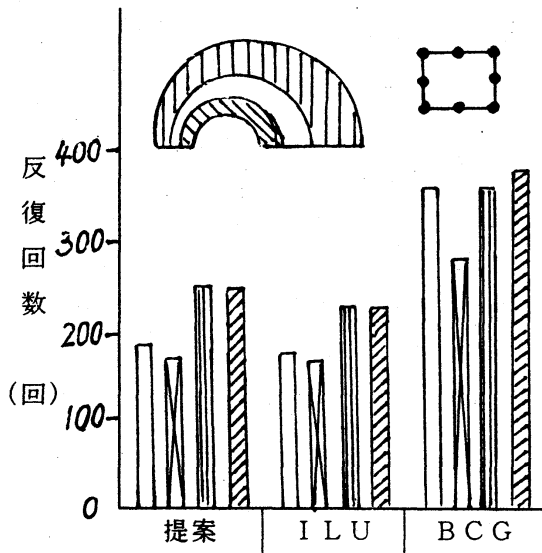


図14 FEM, ケースD2の反復回数

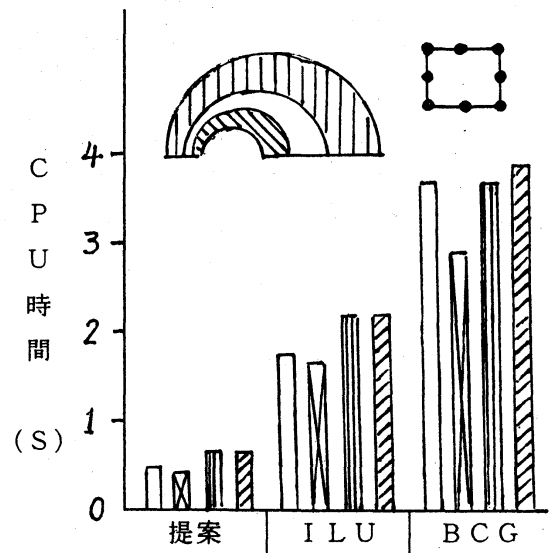


図15 FEM, ケースD2のCPU時間

図12からケースD1ではILUが最小の反復回数で収束するのが分かる。しかし図13からCPU時間では提案が最も高速であるのが分かる。図14からケースD2ではILUと提案がほぼ同じ反復回数で収束する。図15からCPU時間では提案が最も高速であると分かる。ケースD1, D2共に拡散係数をスカラで1:2、スカラで1:100、直交異方性及び一般異方性と変えてテストした。異方性の方がスカラより収束が、いずれの解法でも少し遅いが大差は無い。BCG(不完全LU分解付きBCG法)はILU(不完全LU分解付きBi-CGSTAB法)に比較して、収束性でもCPU時間でも悪い。図12から図15には記載していないが、I-E及びI-P-Eは共に収束しない。その原因は対角優位性が大幅に崩れているためである。本比較はHITAC M-820/80一台で測定した値であり、提案方式はILUよりCPU時間で3~5倍高速である。並列・ベクトル計算機では提案方式が更に高速になると考えられる。

6. おわりに

提案する前処理付きBi-CGSTAB法は、有限要素法で離散化された連立一時方程式をベクトル計算機で高速かつ安定に数値解を求めることができる。本解法はベクトル・並列向き反復解法と言える。

- (1) テストした7ケースの問題で、提案する前処理付きBi-CGSTAB法は不完全LU分解付きBi-CGSTAB法より、HITAC M-820/80のCPU時間で1~10倍高速である。ベクトル・並列計算機で比較すると提案方式は更に高速になると考えられる。
- (2) 不完全LU分解付きBi-CGSTAB法と同じ前処理付きBCG法は、比較した全ての問題でBi-CGSTAB法の方が、収束するまでの反復回数もCPU時間も共に優れている。
- (3) 提案する前処理付きBi-CGSTAB法は不完全LU分解付きBi-CGSTAB法に比較しアイソパラメトリック二次要素では、S-820/80のCPU時間で3~10倍高速である。これより提案方式は二次要素にも十分適用できる。
- (4) 提案する前処理方式は不完全LU分解を使用せず、行列乗算を前処理に使用するため、節点の番号付けで収束が変化しない。従って、CAD向きの方式である。

参考文献

- 1) H.A. Van Der Vorst, Bi-CGSTAB; a Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems, Preprint Nr.633, University Utrecht, Dept. Math. (1990) (Submitted to the SIAM J. Sci. Stat. Compt.)
- 2) 村田健郎, 名取 亮, 唐木幸比古; 大型数値シミュレーション, 岩波書店 (1990)
- 3) 後 保範; ベクトル計算機向きICCG法, 京大数理研講究録514, pp.110~134 (1984)

- 4) H.A.Van Der Vorst ; ICCG and Related Methods for 3D Problems on Vector Computers, Computer Physics Comm.53, pp.223~235 (1989)
- 5) J.A.Meijerink, H.A.Van Der Vorst ; Iterative Solution Method for Linear Systems of Which the Coefficient Matrix is a Symmetric M-Matrix, Math. Comp., Vol.31, pp.148~162 (1977)
- 6) O.Axelsson ; Solution of Linear Systems of Equations, Lecture Note in Math. NO.572, Springer Verlag, pp.1~51 (1977)
- 7) 村田健郎, 小国 力, 唐木幸比古 ; スーパーコンピューター、科学技術計算への適用, 丸善 (1985)
- 8) 藤野清次, 森 正武, 武内敏己 ; 超平面法のベクトル計算機上の性能, 数値解析と科学計算, 京大数理研講究録 7 4 6, pp.11~21 (1991)
- 9) 戸川隼人 ; 共役勾配法, 教育出版 (1976)
- 10) 後 保範 ; 大型疎行列に対する P C G, P C R 法 ; 最近の計算技術, コンピュートロール NO.12, コロナ社, pp.16~22 (1985)
- 11) 張 紹良, 藤野清次 ; 反復解法の収束特性と計算効率, 情報処理学会研究報告 9 1 - N A - 3 7, pp.91~98 (1991)