

SOR 法のベクトル計算機向き書換えによる効率の低下†

藤野清次** 杉原正顕***
小柳義夫† 森正武**

偏微分方程式を差分法で離散化した連立1次方程式をSOR法を使ってベクトル計算機上で解く場合に、ベクトル計算機向き書換えが逆に効率の低下を招く可能性があることが理論的に知られている。そこで、本論文の目的は、その理論的事実を実際にスーパーコンピュータ上で実証することにある。また、理論的な考察が行われている規則的な直交座標系のみでなく、一般曲線座標系の問題においても、その理論的事実がそのまま適合することを実験的に明らかにする。一般にベクトル計算機向き書換えには色々な技法が知られているが、本論文ではSOR的方法あるいは擬似SOR法(付録1参照)と呼んでいる方法を取り上げる。この方法はSOR法が本来持つ収束率の低下を犠牲にする代わりにできるだけ長いベクトル長を確保しようという考えに基づいている。一方、Hyperplane法と呼ばれるベクトル化技法が広く使用されている。この方法は再帰的關係を取り除くように計算順序を並び替えることによって、SOR法本来の収束率を保持しようという考えに基づく。本論文では、以上二つの書換え、すなわちベクトル長の増大を優先的に考える方法：擬似SOR法と、収束率の保持を優先的に考える方法：Hyperplane法によるSOR法のどちらが実際に効率的なベクトル計算機向き書換えであるか比較を行う。テスト問題としてDirichlet条件を含む境界値問題とNeumann条件を含む混合型境界値問題を取り上げ、前者については理論の検証に重点を置き、後者については固有値解析を中心に考察を行った。さらに座標系に関しても、直交座標系の問題に加えて応用面でよく使われる一般曲線座標系の問題も取り扱った。その結果、擬似SOR法は、使用したすべてのベクトル計算機においてHyperplane法によるSOR法と比較して効率が大幅に低下し、ベクトル計算機の高速度を引き出す書換えになっていないことが実験的に確かめられた。

1. はじめに

一般にベクトル計算機における計算効率を上げるための有効なベクトル化技法が数多く知られている。それらに共通して有効な考え方は、再帰的な計算をできるだけ少なくすること、そしてベクトル長をできるだけ長くすること、この二つを挙げることができる。

例えば2次元のPoisson方程式などを差分法で離散化して得られる連立1次方程式を解くとき、中心差分近似を使用すると直交座標系の場合で5点差分、一般曲線座標系の場合では9点差分になる。この連立1次方程式をベクトル計算機上でSOR法を使って解くとき、更新された最新のデータをできるだけ多く使うようにプログラムを書くと、収束するまでの反復回数は少なくて済む。その代わりにベクトル化が十分に行われ

ないため反復1回当りの計算時間は大きくなる。一方、更新前の古いデータだけを使うようにプログラムを書くと、ベクトル化はされるが、逆に反復回数が多くなる。

SOR法のベクトル化率を向上させるためにこれまでに様々なベクトル化技法が提案されてきた。例えば、再帰的演算を回避するために有効な方法として知られるHyperplane法(超平面法)もそのうちの一つの方法である^{1),2)}。この方法では格子の番号を並べ換えることによって計算順序を変更し、周囲の点と再帰的關係になるのを防いでいる。この場合平均ベクトル長はオーダー $O(N/2)$ (N は1辺の格子点数とする)のように見積られる。これはベクトル計算機の高速度を発揮させるに十分なベクトル長さであるとは必ずしも言えないが、このHyperplane法の利点は本来のSOR法の収束率を少しも低下させないことにある。

一方、これと反対に収束率を犠牲にして長いベクトル長を得ることにベクトル計算機向き書換えの主眼を置く方法もある。この考えに基づく典型的な方法がわれわれが擬似SOR法と呼ぶものであり、この方法によればベクトル長はオーダー $O(N)$ と長くなる[付録1参照]。またこの擬似SOR法は配列への格納も連続的になり、不規則な番地のデータを参照・格納することによる効率の低下(機種によってその影響の度合は

† Deterioration of the Efficiency of the SOR Method due to a Modification for Vector Computers by SEIJI FUJINO (Institute of Computational Fluid Dynamics), MASAAKI SUGIHARA (Department of Economics, Hitotsubashi University), YOSHIO OYANAGI (Institute of Information Science and Electronics, The University of Tsukuba) and MASATAKE MORI (Department of Physical Engineering, Faculty of Engineering, University of Tokyo)

** 計算流体力学研究所(株)

*** 一橋大学経済学部

† 筑波大学電子・情報工学系

** 東京大学工学部物理工学科

差がある)を防止することができるため、一見ベクトル計算機向き書換えのように思われる。しかしながら理論的に明らかにされているように、擬似SOR法の漸近収束率はオーダ $O(1/N^2)$ であり、本来のSOR法のオーダ $O(1/N)$ に比べると大幅に低下する。

本論文では、理論的な考察の正当性の裏付けをするために、上記二つの解法をいくつかの例題についてベクトル計算機上で実際に走らせ、その効率を評価した。

最初に文献3)で理論的に与えられた擬似SOR法の最適加速係数を調べる。さらに、直交座標系および一般曲線座標系で記述されたDirichlet条件のラプラス問題を解いて、擬似SOR法とHyperplane法によるSOR法の実行効率を比較する。最後に、より実際的な問題として一般曲線座標系でのNeumann条件を含む混合型境界値問題を取り上げ、収束状況を調べる。解析領域は偏心2重円で、そこで定義されたポアソン方程式の解を求める。またSOR法の反復行列の固有

値を数値的に求め、格子の変形度合や密集度が収束性へどのように影響するかを考察する。そして擬似SOR法の加速係数が本来のSOR法とは違った振舞いをすることも併せて報告する。

本論文では、擬似SOR法と比較する方法としてHyperplane法によるSOR法を採用した。しかしこの方法よりももっと効率的な方法が文献4)、5)に見られるように幾つも存在する。しかし本論文の主旨は最も効率的な解法を見いだすことにあるのではなく、(もちろんそれも非常に大切であるが)あくまでもベクトル計算機利用者がそのプログラムのベクトル化書換えに際して知っておいた方が良い実行効率の低下の顕著な例を明示することにある。

2. SOR法と擬似SOR法の理論解析の検証

解析領域を単位正方形とし、境界条件が以下のよう
に与えられるラプラス方程式のDirichlet問題:

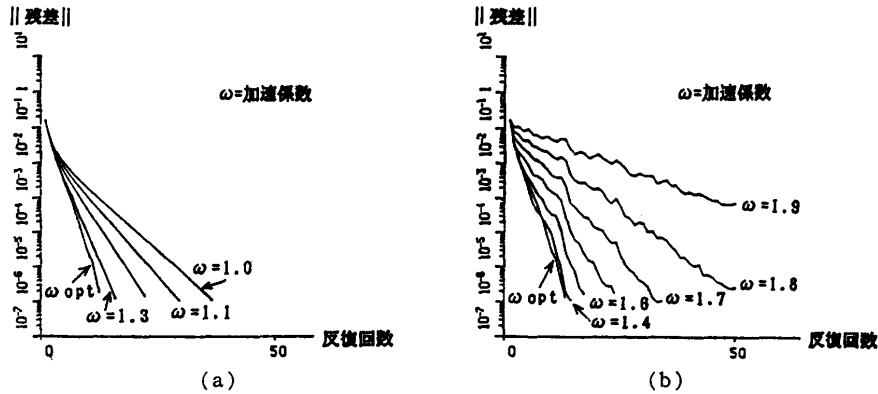


図1 5点差分の場合のSOR法の収束状況
Fig. 1 Convergence behavior of the SOR method with 5 points FDM.

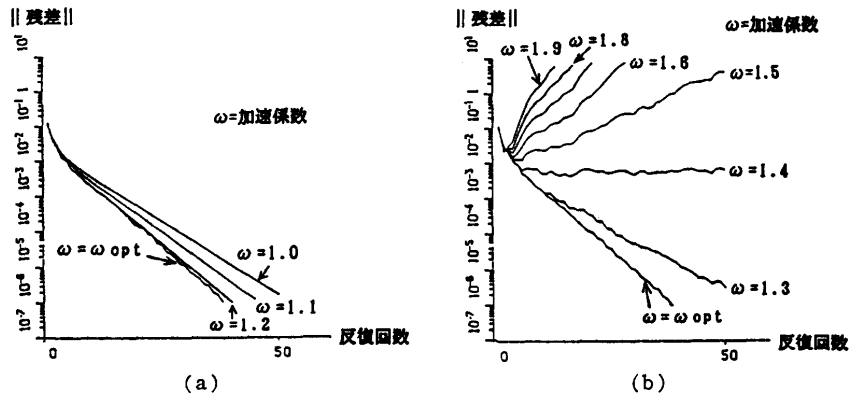


図2 5点差分の場合の擬似SOR法の収束状況
Fig. 2 Convergence behavior of the pseudo SOR method with 5 points FDM.

$$U(0, y)=0, U(1, y)=0, \quad (0 < y < 1) \quad (1)$$

$$U(x, 0)=0, U(x, 1)=0.5 - |x - 0.5| \quad (0 < x < 1) \quad (2)$$

を考える。この正方領域の一边を N 等分して、中心差分近似で離散化し、得られた連立1次方程式を Hyperplane SOR 法と擬似 SOR 法で解いた。そのときの残差の Maximum norm の変化を $N=6$ の場合について図1から図4に示す。

5点差分の場合について、Hyperplane SOR 法における残差の減少状況を図1(a), (b)に、擬似 SOR 法の残差の減少状況を図2(a), (b)に示す。図を(a), (b)に分けたのは、加速係数 ω による変化を見やすくするためである。また9点差分の場合について、Hyperplane SOR 法の残差の減少状況を図3(a), (b)に、擬似 SOR 法の残差の減少状況を図4(a), (b)に示す(初期値はいずれの場合も0とし、計算はすべて倍精度演算で行った)。文献3)で理論的に得た $N=6$ の場合の最適加速係数は次のとおりで

ある。

(1) 5点差分の場合

$$\text{SOR 法: } \omega_{\text{opt}} = 2/(1 + \sin(\pi/N)) = 1.33$$

擬似 SOR 法: $\omega_{\text{opt}} = 1.23$

(2) 9点差分の場合

$$\text{SOR 法: } \omega_{\text{opt}} = 1.31$$

擬似 SOR 法: $\omega_{\text{opt}} = 1.26$

以上の図から明らかのように、文献3)で得られた擬似 SOR 法と SOR 法の最適加速係数の理論値(上記に示す)は、実際の計算例においても最適な加速係数になっており、理論解析が正しいことが分かる。

3. ベクトル計算機上での実行効率の比較

ここでは一边の分割数 N をさらに大きくとり、擬似 SOR 法と Hyperplane 法による SOR 法について、そのベクトル計算機上での実行効率を調べた。使用したベクトル計算機とその最大性能値(カタログ性能)は以下のとおりである。

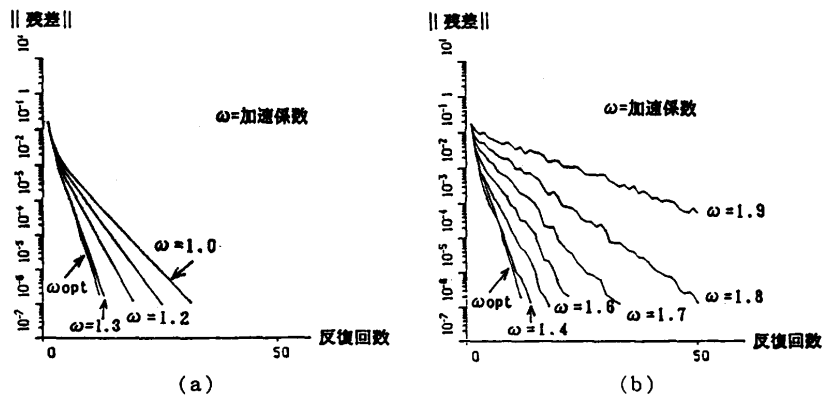


図3 9点差分の場合のSOR法の収束状況
Fig. 3 Convergence behavior of the SOR method.

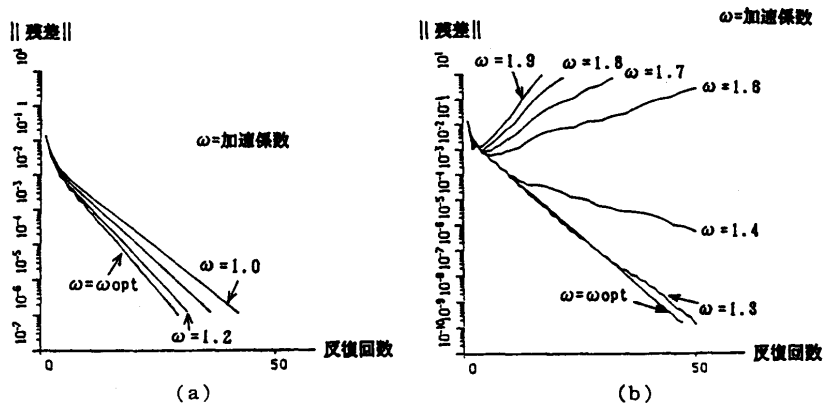


図4 9点差分の場合の擬似SOR法の収束状況
Fig. 4 Convergence behavior of the pseudo SOR method with 9 points FDM.

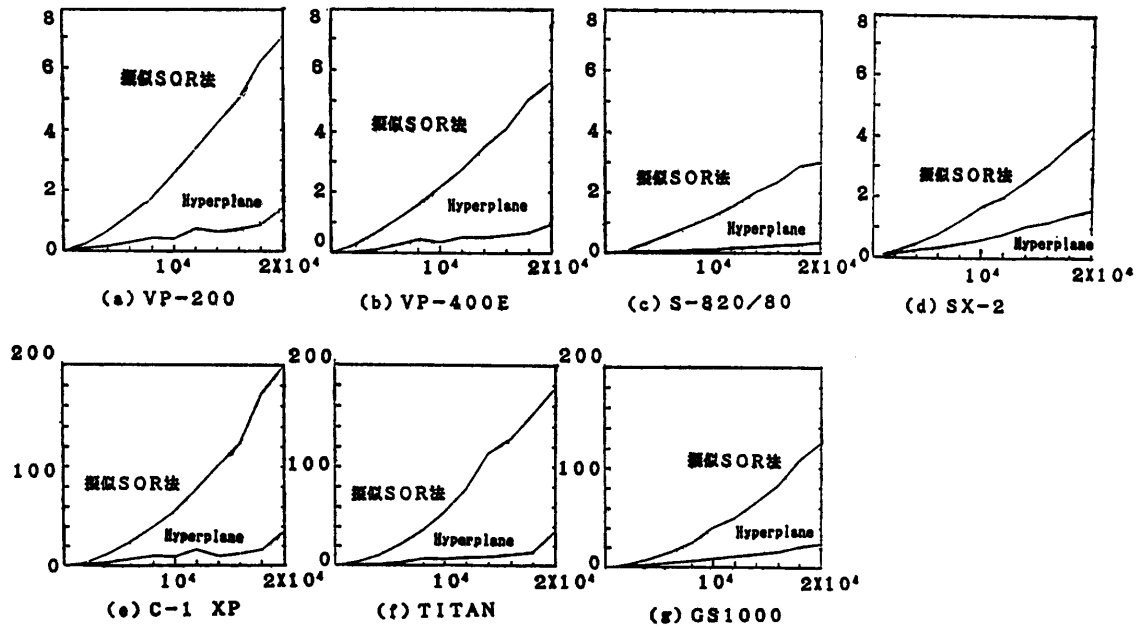


図5 5点差分の場合のCPU時間の比較

横軸は格子数, 縦軸はCPU時間(単位: 秒).

Fig. 5 Comparison of CPU time on several vector computers (in the case of 5 points FDM).

- (a) 富士通 VP-200: 570 MFLOPS
- (b) 富士通 VP-400E: 1700 MFLOPS
- (c) 日立 S-820/80: 2000 MFLOPS
- (d) 日本電気 SX-2: 1300 MFLOPS
- (e) CONVEX C-1 XP: 20 MFLOPS
- (f) ARDENT TITAN: 16 MFLOPS
- (g) STELLAR GS1000: 40 MFLOPS

3.1 直交座標系(5点差分)のディリクレ問題

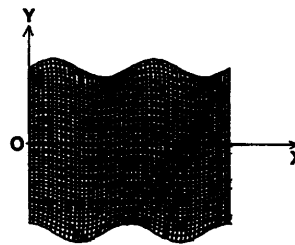
解析条件を(i)から(vii)に列挙する.

- (i) 方程式: ラプラス方程式
- (ii) 解析領域: 単位正方形
- (iii) 境界条件: 第2章の(1), (2)式使用
- (iv) 格子数: 約2000点から約20000点まで10ケース (45², 63², 77², 89², 99², 109², 119², 127², 135², 141²)
- (v) SOR法の初期値: すべて0.0
- (vi) 収束判定値: L₂ノルムで10⁻⁶
- (vii) 使用した加速係数: 第2章の理論値を採用

この条件で数値実験を行い両解法のCPU時間を図5(a)~(g)に示す. 使用したすべての機種においてHyperplane SOR法の方が擬似SOR法よりはるかに効率的であることが分かった.

3.2 一般曲線座標系⁶⁾のディリクレ問題

一般曲線座標系では, 座標変換により交差微分項が新たに現れる. そのために直交座標系でのラプラス方



境界条件について

上端の境界値: $V_a (=20)$ と下端の境界値: $V_b (=50)$ は固定. 左端と右端の値は上端からの距離に応じて V_a と V_b とで線形補間した値で固定.

図6 解析領域と境界条件

Fig. 6 The domain for analysis.

程式の5点差分式は一般曲線座標系では9点差分式になる [付録2参照]. ここでは, 縦横の格子数が等しい場合(A)と横方向の格子数が縦方向のそれより非常に多い場合(B)の二つの場合について, ベクトル長の増加がどの程度効率に影響するか調べた.

(A) 縦横の格子数が等しい場合

解析条件は以下に示すとおりである.

- (i) 方程式: ラプラス方程式
- (ii) 解析領域: 図6参照
- (iii) 境界条件: ディリクレ条件 (図6参照)
- (iv) 格子数: 約2000点から約20000点まで10ケース (46², 62², 78², 90², 102², 110², 118², 126², 134², 142²)

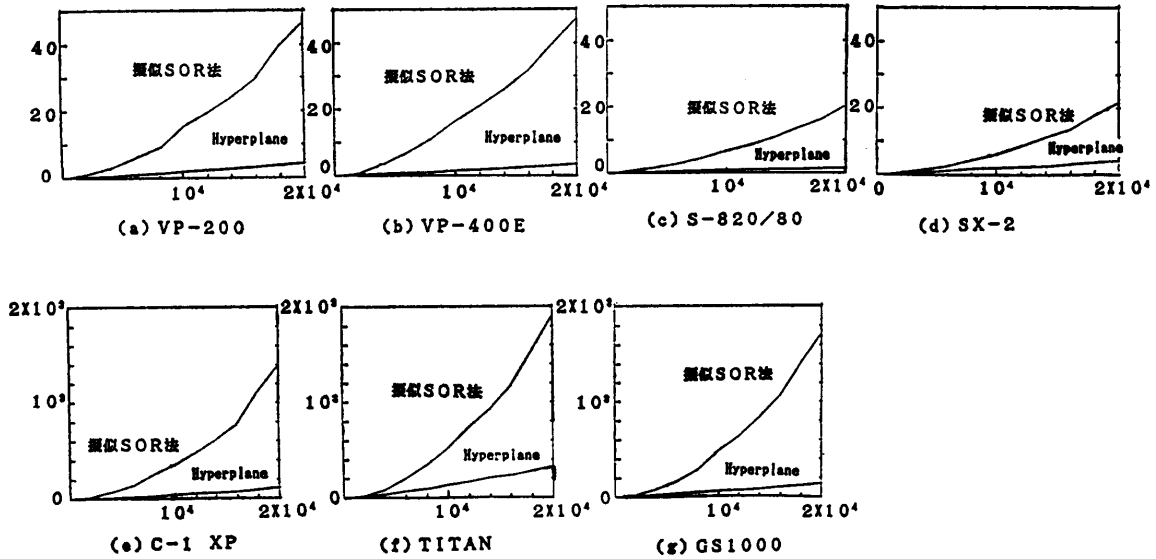


図7 9点差分の場合のCPU時間の比較

横軸は格子数, 縦軸はCPU時間(単位: 秒).

Fig. 7 Comparison of CPU time on several vector computers (in the case of 9 points FDM).

- (v) SOR法の初期値: すべて0.0
- (vi) 収束判定値: L_2 ノルムで 10^{-6}
- (vii) 加速係数: 0.02刻みで加速係数を調べた.

この条件にて数値実験を行い両解法のCPU時間を図7(a)~(g)に示す. 5点差分と同様9点差分の場合でも, Hyperplane SOR法の方が擬似SOR法より圧倒的に効率が良いことが分かった.

(B) 横方向の格子数が縦方向より多い場合

解析条件は以下に示すとおりである.

- (i) ラプラス方程式
- (ii) 解析領域: 格子数が3280 (縦×横=40×82)の分割図を図8に示す.

(iii) 境界条件: ディリクレ条件

(iv) 格子数: 次の3ケース

- (1) 格子数 3280 (縦×横=40×82)
- (2) 格子数 6480 (縦×横=40×162)
- (3) 格子数 12880 (縦×横=40×322)

- (v) SOR法の初期値: すべて0.0
- (vi) 収束判定値: (A)と同じ.
- (vii) 加速係数: 0.02刻みで加速係数を調べた.

表1(a), (b)は最も速く収束した時のCPU時間を表したものである. 二つの計算機VP-200とS-820/80の結果を示す.

(iv)項で述べたように, 格子数は縦方向が一定, 横方向のみ約2倍ずつ増加させた. この場合擬似SOR法のベクトル長は82, 162, 322と次第に長くなるの

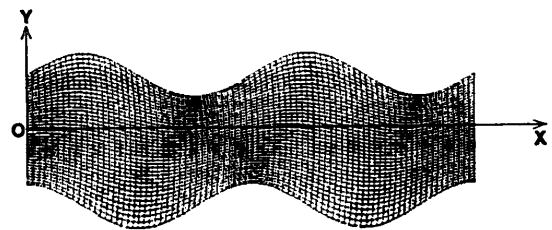


図8 解析領域 (格子数82×40の場合)

境界条件は図6と同じ.

Fig. 8 The domain for analysis (The number of grids is 82×40 and the boundary conditions are as same as in the case of Fig. 6).

表1(a) CPU時間の比較 (VP-200の場合) (単位: 秒)
Table 1(a) Comparison of CPU time of two methods on VP-200 (in seconds).

格子数	3280	6480	12880
擬似SOR法	0.49	2.55	17.2
Hyperplane SOR法	0.27	0.72	3.29

表1(b) CPU時間の比較 (S-820/80の場合) (単位: 秒)

Table 1(b) Comparison of CPU time of two methods on S-820/80 (in seconds).

格子数	3280	6480	12880
擬似SOR法	0.17	0.82	9.03
Hyperplane SOR法	0.08	0.27	1.69

で, この計算は擬似SOR法にとって有利になるはずである. しかしながら表1(a), (b)は, 格子数を増

せば増すほど Hyperplane SOR 法の方がより速く収束することを表している。収束性の保持を第1と考える計算方法の方がベクトル計算機では効率が高いことが分かる。

3.3 一般曲線座標系の混合型境界値問題

ここではより実的な問題として、図9(a)~(f)に示す偏心2重円上で定義されたポアソン方程式の問題を取り上げることにする。すなわち、外側の円は固定し、内側の円の中心を次第に左に移動させ、その二つの円で囲まれた領域をとる。

図9の左側の図(a)は、 r 方向の分割の割合をゆるやかに変化させたものであり、それを偏心させたものが(b)、(c)である。また右側の図(d)は内側の円の周囲に格子を密集させたものであり、これを偏心させたものが(e)、(f)である。最小格子と最大格子との面積の比は、図(a)は2.60、図(b)は6.79、図(c)は45.1、そして図(d)は60.1、図(e)は157.0、図(f)は1040.0であった。このような格子は一般曲線座標系で2次元円柱回りの流れを計算する時によく用いられる⁸⁾。実際の流れ解析ではさらに物体近傍で格子をできるだけ直交させるなどの処理を施すのが通例

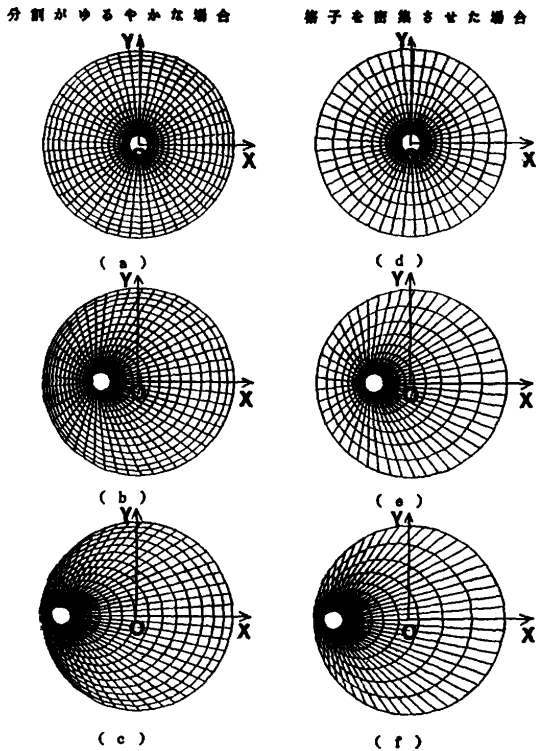


図9 偏心2重円の格子図

Fig. 9 The grid systems of eccentric double circle.

であるが、ここではこれらの処理は行っていない。また流れ解析では境界条件としてノイマン条件を使用する方が自然であることが多く、この章ではディリクレ条件だけでなく、ディリクレ・ノイマン条件両方が混在する境界条件の場合の収束性も調べた。

解析条件は以下のとおりである。

(i) 解く方程式: $U_{xx} + U_{yy} = -2\sin(X)\sin(Y)$
を一般曲線座標系に変換した式
【付録2参照】

(ii) 解析領域: 円

外側の円: $X^2 + Y^2 = 5^2$

内側の円: (a), (d): $X^2 + Y^2 = 1^2$

(b), (e): $(X+2)^2 + Y^2 = 1^2$

(c), (f): $(X+4)^2 + Y^2 = 1^2$

(iii) 境界条件: θ 方向は周期条件採用, 内側の円はディリクレ条件に固定, 外側の円はディリクレ条件とノイマン条件, 二つの場合について調べた。

(iv) 格子数: r 方向: 16分割

θ 方向: 50分割

(v) SOR法の初期値: すべて0.0

(vi) 収束判定値: L_2 ノルムで 10^{-6}

(vii) 解析解: $U = \sin(X)\sin(Y)$

またここではSOR法の収束性の議論をより精密に行うため、二つのSOR法の反復行列のすべての固有値を以下のようにして求めた⁹⁾。連立1次方程式の係数行列を A ($n \times n$ の正方行列, n は未知数の数)とすると、これは次のように分解できる。

$$A = D + L + U \tag{3}$$

ただし D は対角行列, L は下三角行列, そして U は上三角行列とする。すなわち,

$$D = \begin{bmatrix} a_{11} & & 0 \\ & a_{22} & \\ 0 & & a_{nn} \end{bmatrix} \quad L = \begin{bmatrix} 0 & & 0 \\ a_{21} & 0 & \\ a_{31} & a_{32} & 0 \\ \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & 0 \end{bmatrix}$$

$$U = \begin{bmatrix} 0 & a_{12} & a_{13} & \dots & a_{1n} \\ & 0 & a_{23} & \dots & a_{2n} \\ & & & \ddots & \\ & & & & 0 \end{bmatrix}$$

この時SOR法の反復行列は (ω は加速係数),

$$(D + \omega L)^{-1} \{ (1 - \omega)D - \omega U \} \tag{4}$$

で表される。一方、擬似SOR法の反復行列は、対角要素の一つ下の要素の取扱いが変わり、次の行列 E と

に単調に小さくなる。

また、ノイマン条件の時のスペクトル半径は、ディリクレ条件の時と比べると両方法において大きな値になるが、とくに擬似SOR法の場合、収束限界1.0に非常に近くなってしまふ。

(A)の場合において加速係数と収束状況との関係を表したものを図10(a), (b)に示す。この図からわかるように、擬似SOR法では最適な加速係数よりわずかに(0.02~0.04)でも大きい加速係数を使用すると発散してしまう。そのためこの擬似SOR法を使って問題を解く時は、発散するのを防ぐために最適値より相当小さい加速係数を使用せざるを得なくなる。何故ならば、実際的な問題を解く場合には最適加速係数を調べるのに多大な手間を要することが多く、最適値を正確に決めるのが難しいからである。

一方、Hyperplane SOR法の加速係数については、最適な値よりかなり大きな値でも収束することがこの図からわかる。参考までに(A)と(C)の場合について、収束までの計算(CPU)時間を表4(a), (b)に示す。なお使用した計算機はS-820/80である。各格子分割においてHyperplane SOR法の計算時間を1.0とした時の比率を括弧内に示す。

ノイマン条件の場合には、ディリクレ条件の場合よりいずれの解法ともに多くの計算時間を必要とする。しかしその増加率はHyperplane法によるSOR法よりも擬似SOR法の場合の方がかなり大きい。

以上示してきた結果はすべて最適な加速係数を使用した場合を比較したものである。したがって、最適値より少しでも大きい加速係数を使うと発散の可能性が高い擬似SOR法は、効率的にも決して良い解法でないだけでなく実用上きわめて使いにくい方法であることがわかる。

4. 結 論

ベクトル計算機向き書換えとして擬似SOR法を取り上げHyperplane法によるSOR法と比較しその収

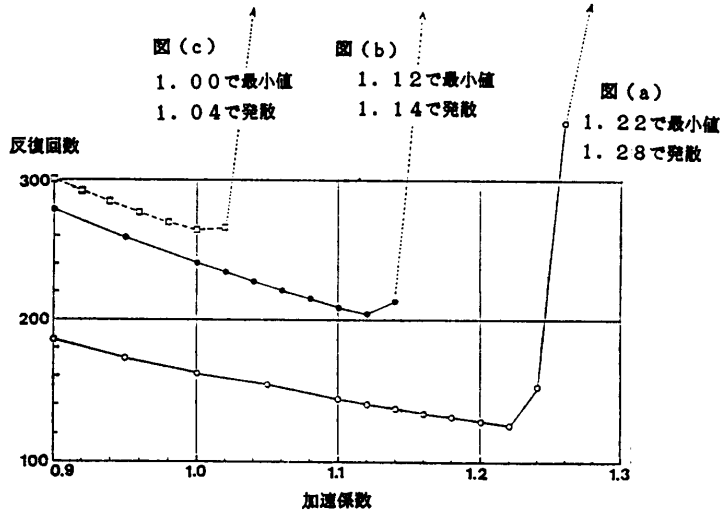


図10(a) 擬似SOR法の加速係数と収束状況の関係
Fig. 10(a) The relation between the acceleration parameter and the convergence of the pseudo SOR method.

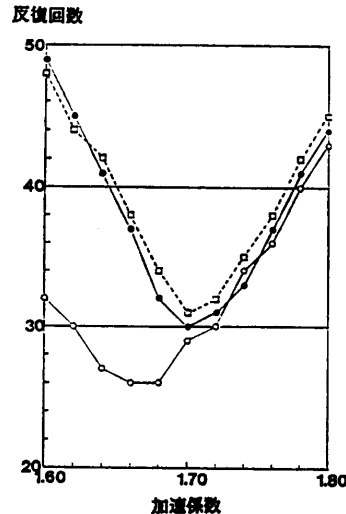


図10(b) Hyperplane SOR法の加速係数と収束状況の関係
Fig. 10(b) The relation between the acceleration parameter and the convergence of the Hyperplane SOR method.

束性について考察した。その結果、以下のことがわかった。

- ①5点差分、9点差分いずれの場合も、縦横の格子数が一定の時、擬似SOR法はHyperplane法によるSOR法と比較して効率が非常に悪い。
- ②縦横の格子数の比を約2~8倍と増加させるとベクトル長が長くなるため一見擬似SOR法の方が効率面で有利に思われる。しかしこのようなケースでも擬似SOR法はHyperplane法によるSOR法と比較してやはり効率が悪い。

③ 実際的な問題として選んだ一般曲線座標系における混合型境界値問題の数値実験結果から、以下のことがわかった。

(3-1) 2重円の偏心の度合いが大きくなると、擬似SOR法の最適加速係数は次第に小さくなり加速限界の1.0に近づいてしまう。一方、Hyperplane SOR法の最適加速係数にはあまり大きな値の変動はない。

(3-2) 擬似SOR法では、最適値よりもわずかも大きな加速係数を使うと発散してしまう。

(3-3) ノイマン条件で定義された境界の割合が増えると擬似SOR法のスペクトル半径は収束限界の1.0に極めて近くなる。したがってディリクレ条件の場合よりもHyperplane SOR法との効率の差は一層大きくなる。

④ 結局、その解法本来の収束率を保持するようにベクトル化書換えを行うことが最も大切である。

参考文献

- 1) Lamport, L.: The Parallel Execution of DO Loops, *Comm. ACM*, Vol. 17, pp. 83-93 (1974).
- 2) 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, No. 514, pp. 110-134 (1984).
- 3) 杉原正顕, 小柳義夫, 森 正武, 藤野清次: ベクトル計算機におけるSOR的方法の効率について, 情報処理学会論文誌, Vol. 31, No. 6, pp. 930-938 (1990).
- 4) 村田健郎, 小国 力, 三好俊郎, 小柳義夫: 工学における数値シミュレーション, 丸善, 東京 (1988).
- 5) Ortega, J. M.: *Introduction to Parallel and Vector Solution of Linear Systems*, Plenum, New York and London (1988).
- 6) Thompson, J. F. and Warsi, Z. U. A.: *Numerical Grid Generation*, North-Holland, Amsterdam (1974).
- 7) Vinokur, M.: On One-dimensional Stretching Functions for Finite-Difference Calculations, *J. Comput. Phys.*, Vol. 50, No. 2, pp. 215-234 (1983).
- 8) Kawamura, T., Takami, H. and Kuwahara, K.: Computation of High Reynolds Number Flow around a Circular Cylinder with Surface Roughness, *Fluid Dynamics Research*, Vol. 1, pp. 145-162 (1986).

付録1 擬似SOR法

単位正方形領域で定義されたラプラス方程式を差分法で解くことを考える。境界条件は全周ディリクレ条件とし、領域を N^2 等分した各格子点における未知数を配列 U で表す。ラプラス方程式を差分近似(5点差分)して U に関する連立1次方程式を解く時のSOR法および擬似SOR法のプログラムの主要部分は次のようになる。加速係数を OMEGA で表す。5点差分近似で使用される格子点を図11に示す。

<通常のSOR法のプログラム例>

```
DO 10 i=1, N-1
DO 20 j=1, N-1
U(i, j)=U(i, j)+OMEGA
*(0.25*(U(i, j-1)+U(i, j+1)
+U(i-1, j)+U(i+1, j))-U(i, j))
```

20 CONTINUE

10 CONTINUE

このままではベクトル化はできない。そこで、中間変数として1次元配列 V を用いてプログラムを次のように書き換えればベクトル化が可能になる。

<擬似SOR法のプログラム例>

```
DO 10 i=1, N-1
DO 20 j=1, N-1
V(j)=U(i, j)+OMEGA
*(0.25*(U(i, j-1)+U(i, j+1)
+U(i-1, j)+U(i+1, j))-U(i, j))
```

20 CONTINUE

```
DO 30 j=1, N-1
```

```
U(i, j)=V(j)
```

30 CONTINUE

10 CONTINUE

上のプログラムで内側の20のDOループと30のDOループがベクトル化される。この方法のベクトル長は $N-1$ 、すなわち、 j 方向の分割数 -1 になる。20のDOループの計算では右辺として更新前のデータだけを使っているため、厳密な意味でのSOR法にはなっ

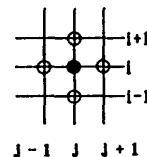


図11 5点差分近似の格子点の配置

Fig. 11 Distribution of gridpoints of 5-point difference approximation.

ていない。厳密な意味での SOR 法では右辺の $U(i, j-1)$ として更新された値を使わなければならない。

付録 2 一般曲線座標系と 9 点差分

物理座標を (x, y) , 計算面の座標を (ξ, η) とした時, 例えばラプラスアンは一般曲線座標系で表示すると, 次のようになる。

$$\nabla^2 = \frac{1}{J^2} \left(\sigma \frac{\partial}{\partial \xi} + \tau \frac{\partial}{\partial \eta} + \alpha \frac{\partial^2}{\partial \xi^2} - 2\beta \frac{\partial^2}{\partial \xi \partial \eta} + \gamma \frac{\partial^2}{\partial \eta^2} \right)$$

$$D = \frac{1}{J} \left(\frac{\partial y}{\partial \eta} \frac{\partial u}{\partial \xi} - \frac{\partial y}{\partial \xi} \frac{\partial u}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial v}{\partial \xi} + \frac{\partial x}{\partial \xi} \frac{\partial v}{\partial \eta} \right)$$

計量 $\alpha, \beta, \gamma, J, \sigma, \tau$ は

$$\alpha = \left(\frac{\partial x}{\partial \eta} \right)^2 + \left(\frac{\partial y}{\partial \eta} \right)^2, \quad \beta = \frac{\partial x}{\partial \xi} \frac{\partial x}{\partial \eta} + \frac{\partial y}{\partial \xi} \frac{\partial y}{\partial \eta}$$

$$\gamma = \left(\frac{\partial x}{\partial \xi} \right)^2 + \left(\frac{\partial y}{\partial \xi} \right)^2, \quad J = \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi}$$

$$\sigma = \frac{1}{J} \left(D_y \frac{\partial x}{\partial \eta} - D_x \frac{\partial y}{\partial \eta} \right), \quad \tau = \frac{1}{J} \left(D_x \frac{\partial y}{\partial \eta} - D_y \frac{\partial x}{\partial \xi} \right)$$

$$D_x = \alpha \frac{\partial^2 x}{\partial \xi^2} - 2\beta \frac{\partial^2 x}{\partial \xi \partial \eta} + \gamma \frac{\partial^2 x}{\partial \eta^2}$$

$$D_y = \alpha \frac{\partial^2 y}{\partial \xi^2} - 2\beta \frac{\partial^2 y}{\partial \xi \partial \eta} + \gamma \frac{\partial^2 y}{\partial \eta^2}$$

である。ここで交差微分項: $\frac{\partial^2 u}{\partial \xi \partial \eta}$ は,

$$0.25(U_{i+1, j+1} + U_{i-1, j-1} - U_{i+1, j-1} - U_{i-1, j+1})$$

のように離散化されるので, 方程式は全体として 9 点差分式になる。

(平成 2 年 9 月 13 日受付)

(平成 2 年 12 月 18 日採録)



藤野 清次 (正会員)

1950 年生。1974 年京都大学理学部 (専攻数学) 卒業。同年三洋電機 (株) に入社。以後, 電磁場計算, 医療機器画像処理計算, 熱流体計算の研究・開発に従事。1988 年計算流体力学研究所 (株) 入社。研究部主任研究員。偏微分方程式の離散化から生じる大型疎連立一次方程式のベクトル計算機向き高速解法に興味を持つ。応用数学会会員。



杉原 正順 (正会員)

昭和 29 年生。昭和 52 年東京大学工学部計数工学科卒業。昭和 57 年同大学院博士課程修了。工学博士。同年東京大学助手, 筑波大学助手, 昭和 62 年講師, 昭和 62 年一橋大学講師, 昭和 63 年同助教授, 現在に至る。専門は数値解析, 応用数学。日本応用数学会, 日本数学会各会員。



小柳 義夫 (正会員)

昭和 18 年生。昭和 41 年, 東京大学理学部物理学卒業。昭和 46 年, 東京大学大学院理学系研究科物理学専門課程博士課程修了。理学博士。同年, 東京大学理学部物理学教室助手。昭和 48 年, 高エネルギー物理学研究所物理研究系助手。昭和 53 年, 筑波大学電子・情報工学系講師・助教授を経て教授。数値解析とその物理学への応用を研究。とくにベクトル計算機, 並列処理, スーパーコンピューティングなど。格子ゲージ理論のための超並列計算機 QCDPAX を製作。日本物理学会, 日本統計学会, 応用統計学会, 計算機統計学会各会員。



森 正武 (正会員)

昭和 12 年生。昭和 36 年東京大学工学部応用物理学卒業。昭和 40 年同大学院博士課程中退。同年同大学助手, 昭和 45 年京都大学助教授, 昭和 54 年筑波大学教授を経て, 平成元年から東京大学工学部物理工学科教授。工学博士。専門は数値解析。著書「数値解析」, 「曲線と曲面」, 「有限要素法とその応用」, 「FORTRAN 77 数値計算プログラミング」ほか。日本応用数学会, 日本数学会, 日本物理学会各会員。