

小規模ブロック化行列の多項式を用いた 共役勾配法の前処理手法の開発

田中伸厚[†] 寺坂晴夫[†]

計算機上で対称行列を共役勾配法を用いて解析する場合、収束を加速するための前処理は非常に重要である。逐次処理型の計算機では不完全コレスキー分解による方法が主流であり、ベクトル型スーパーコンピュータにおいても、データの再帰参照を避ける方法である超平面法との組み合わせで汎用的に用いられている。しかし、ベクトル計算機上では、その手法もベクトル長が短く、計算効率の点から見ればそれほど優れた方法ではない。さらに、最近注目されている並列計算機、特に超並列計算機では、上記のような手法は並列処理効率が非常に悪く、前処理なしの場合に比べかえって計算時間がかかる。これに対し、ベクトル処理および並列処理性の優れた方法として行列の多項式を用いる方法があるが、ノイマン級数を用いる場合は収束性に問題があった。本研究では、そうした問題点を解決するため、小規模ブロック化の手法を適用し、収束性を改善する。ここでは、まず、ブロック化行列の性質を調べることにより、その方法が共役勾配法の前処理として適していることを証明し、ベクトル計算機上での実際の解析によりその有効性を確認する。さらに、多項式の係数を最適化することにより収束性の向上を図る。

Development of a Preconditioning Method for Conjugate Gradient Algorithms using Polynomial of a Matrix Blocked in Small Scale

NOBUATSU TANAKA[†] and HARUO TERASAKA[†]

Preconditioning plays a very important role in the computational conjugate gradient method for symmetric matrix analysis. The incomplete Cholesky decomposition has been one of the most popular preconditioning method on sequential computers, and has also been used on traditional vector supercomputers, combined with the hyperplane method to avoid recurrence relations of data. However, such a combined method does not achieve good computing performance, because the speedup due to vectorization is not so high. On parallel computers, especially on massively parallel ones, the method is never effectively processed in parallel, and needs more computing time than that without preconditioning. Another way of preconditioning uses the Neumann series of a matrix, which is excellent in parallel processing, but inferior in convergence. This study describes the preconditioning method using blocked matrix polynomial (BMP) in which a technique of blocking in small scale is combined with the former method for improvement of the convergence. Analysis of the characteristics confirmed that the method satisfies the requirements of the preconditioning for the conjugate gradient scheme. Then, its efficiency in a practical application was demonstrated by numerical experiments on a vector computer. It was also shown that even better performance can be achieved when the general polynomial takes place of the Neumann and the coefficients are optimized for the conjugate gradient method.

1. 序 論

流体解析、構造解析など計算力学の問題は、自然現象をモデル化して偏微分方程式を構成し、それを有限差分法や有限要素法で離散化近似することによって、最終的には大規模な連立一次方程式を解くことに帰着

されることが多い。それらの解析において、計算時間の大部分がその連立一次方程式を解く部分に消費され、効率の良い解析を行うことはこの連立一次方程式をいかに速く解くかにかかっているといえる。スーパーコンピュータの出現により大規模な連立一次方程式を高速に解くことが可能となり、大規模な解析が行えるようになってきた。しかし、より詳細な解像度を求め、さらに大規模で複雑な問題を高速に解きたいという需要も多くあり、そうした点から超並列計算機へ

[†] 株式会社東芝 原子力技術研究所

の期待が高まっている。一方、近年、次々と高性能な並列計算機も発表され、それらを用いた実用的な解析も行われ始めている^{21,22}。われわれは、今までこうした超並列計算機を用いて、有限差分法による非圧縮性流体解析を行い、その利点、効率、問題点などについて考察を行ってきた²³⁻⁴¹。その中で、特に問題となったのは上でも述べた連立一次方程式の解法である。有限差分法による非圧縮性流体解析でも、その計算時間の大部分を占めるのが圧力に関するポアソン方程式の解法部分である。ポアソン方程式を有限差分法で離散化して得られる連立一次方程式の係数行列は対称正定値となるため、共役勾配法がよく用いられる。さらにその収束性を加速するための前処理を施すことが一般的である。前処理としては不完全コレスキー分解¹⁰が主流であるが、その方法は、反復ごとに前進消去・後退代入を行う必要があり、ベクトル型のスーパーコンピュータでは、データの再帰参照により、そのままではベクトル処理できない。この問題点を解決するために用いられるのが超平面法である。現在、その手法との組み合わせにより不完全コレスキー分解を前処理とする共役勾配法はもっとも一般に用いられている。しかし、この超平面法にはデータの再帰参照が残っているため、ベクトル長を長くすることができず計算効率は良くない。さらに、並列度の高い超並列計算機では、その部分がボトルネックとなり、計算効率が非常に悪くなることが確認されている^{39,40}。

一方、P. Dubois ら⁵¹は前処理として、係数行列 A に関するノイマン級数を途中で打ち切ったものを用いることを提案している。この前処理手法は、完全に陽的(データの依存関係がない)である点は注目に値するが、収束性の点で問題があり、前処理を行わない方法に比べてたいした効果をもたらさないことが多い。このような問題点を解決するため、ここでは、その方法に行列の小規模ブロック化の手法を導入し、収束性の向上を図る。一方、O. G. Johnson ら⁶¹および Y. Saad⁷¹は、P. Dubois らの手法を改良し、係数行列に関する一般多項式を用いることにより収束性を向上させる方法を提案している。本研究でも、上記の手法に、こうした考え方を取り入れ、収束性の向上を試みる。また、本研究の手法はベクトル計算機や並列計算機上で処理効率良く動作することが考慮されており、ここではベクトル計算機上でその点も確認する。

2. 理論的考察

ここで対象とする問題は、2次元のポアソン方程式とする。

$$-\Delta\phi = \psi \text{ in } \Omega \subset \mathbb{R}^2$$

with

$$\phi = \zeta \text{ or } \frac{\partial\phi}{\partial n} = \zeta \text{ on } \partial\Omega \tag{2.0.1}$$

ここで n は単位法線方向を表すベクトルである。この偏微分方程式を、有限差分法により離散化して得られる連立一次方程式

$$Ax = b \tag{2.0.2}$$

を解くこととする。その場合、係数行列 A は対称正定値行列となる。また、 A はスケーリングによりその対角項はすべて1になっているものとする。

2.1 前処理付き共役勾配法

まず、前処理付きの共役勾配法のアルゴリズムをまとめておく。

前処理因子である行列 K が正定値であれば、 $K = V^T V$ となる V が存在する。前処理付き共役勾配法では、この V を用いて $\tilde{A} = VAV^T$ と変換し、(2.0.2)式のかわりに

$$\tilde{A}\tilde{x} = \tilde{b} \tag{2.1.1}$$

に共役勾配法を適応するものである。ただし

$$\tilde{x} = V^{-T}x, \tilde{b} = Vb \tag{2.1.2}$$

である。 K として

$$K \equiv A^{-1} \tag{2.1.3}$$

となるものを選べば、(2.0.2)式より(2.1.1)式を解くほうが収束が速い。図1に前処理付き共役勾配法のアルゴリズムを示す。

このアルゴリズムの中に現われる $\tilde{r}_0 = Kr_0$ や \tilde{r}_{k+1}

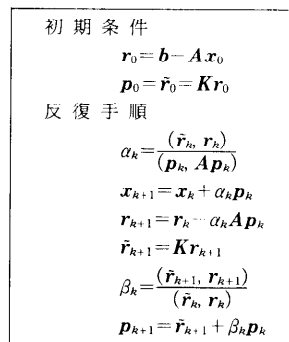


図1 前処理付き共役勾配法のアルゴリズム
Fig. 1 Algorithm for preconditioned conjugate gradient method.

$=K\mathbf{r}_{k+1}$ においては、 K^{-1} が既知なことが一般的である。その場合

$$K^{-1}\tilde{\mathbf{r}}_{k+1}=\mathbf{r}_{k+1} \quad (2.1.4)$$

を解く必要がある。これは、見方を変えると、(2.1.3)式の関係より

$$A\tilde{\mathbf{r}}_{k+1}=\mathbf{r}_{k+1} \quad (2.1.5)$$

を近似的に解くことを意味している。不完全コレスキー分解法¹⁰⁾や次の章で述べるノイマン級数展開を用いた方法⁵⁾は、この近似解法の1つである。一方、 A に関する多項式を用いる方法^{6),7)}は、(2.1.5)式を近似的に解くことによって K を求めるのではなく、 K を何らかの行列の多項式の形で直接表現し、(2.1.1)式における \tilde{A} の固有値分布を、共役勾配法で解きやすい形のものに最適化することを目的とする。この点が、他の前処理方法とは異なった特徴である。

2.2 ノイマン級数行列

ノイマン級数を用い、(2.1.5)式を近似的に解くことを考える。ここで、 D を正定値行列であるとして、 $A=D+(A-D)$ と正則分離し、

$$R=I-D^{-1}A \quad (2.2.1)$$

とおく。

$$A=D(I-R) \quad (2.2.2)$$

であるから、 R のスペクトル半径 $\rho(R)$ が $\rho(R)<1$ であれば

$$A^{-1}=f_n(R)D^{-1} \quad (2.2.3)$$

と級数展開することができる。ただし

$$f_n(x)=1+x+\dots+x^n \quad (2.2.4)$$

とする。ここで、 $f_n(R)$ を途中で打ち切った不完全級数展開行列を

$$A^{-1}\cong M_n=(I+R+\dots+R^n)D^{-1} \\ =f_n(R)D^{-1} \quad (2.2.5)$$

とおくと、この M_n を前処理因子として使うことができる。この M_n を n 次のノイマン級数行列と呼ぶ。

D^{-1} の固有値は D の固有値の逆数だから、 D^{-1} も正定値である。したがって D が対称なら

$$D^{-1}=U^T U \quad (2.2.6)$$

となる正則行列 U が存在する。ここで、

$$Q=I-UAU^T \quad (2.2.7)$$

とおくと M_n は以下のように書き直せる。

$$M_n=U^T(I+Q+\dots+Q^n)U \\ =U^T f_n(Q)U \quad (2.2.8)$$

したがって、 M_n は $f_n(Q)$ と同じ符号の固有値をもつ。 Q の固有値を q とすると $f_n(Q)$ の固有値

は $f_n(q)$ に等しく、したがって $|q|<1$ であれば $n=0, 1, 2, \dots$ に対し、 $f_n(q)>0$ となり M_n は正定値行列となる。ここで Q と R は同じ固有値をもつから、 $\rho(R)<1$ であれば M_n は正定値行列である。また、 A が対称であれば UAU^T も対称であり、 Q も対称行列となるので $f_n(Q)$ も対称である。よって、 M_n は対称行列となる。

2.3 D行列の選択

ベクトル計算機、並列計算機で効率良く処理するには、 D^{-1} が簡単に求まらなければならず、またその D^{-1} を求めるための手続きにデータの再帰参照がないものを選ぶ必要がある。

Duboisらは D 行列として、スケーリングされた A に対し、その対角成分のみを取り出した単位行列を用いている。しかし、こうした方法では収束性を改善することができず、理論的には、前処理しないほうが多数の反復の後では良い近似解を得ることができるという報告がある¹¹⁾。こうした問題点を解決するためには、 R 行列の固有値分布を改善するような D 行列を用いる方法が考えられる。ここでは、 A から、小規模のブロック化により新しい D 行列を構成し、(2.2.1)式で表される R 行列の多項式により前処理を行う方法を提案する。

一般に、行列のブロック化は、図2に示すように、 A を適当にブロック化し、 D 行列としてこの A の対角項付近のブロック行列を取り出したもの(図3)を用いるものである。ここでは、以下、この D をブロック化行列と呼び、 D^s を部分行列と呼ぶこととする。このような D を用いると、 D^{-1} は図3に示すように

$$A = \begin{bmatrix} D_1^s & B_1 & 0 & \dots & C_1 & 0 & \dots \\ B_1 & D_2^s & B_2 & \dots & 0 & C_2 & \dots \\ 0 & B_2 & D_3^s & \dots & \dots & 0 & \dots \\ \dots & \dots & \dots & \dots & B_{i-1} & \dots & \dots \\ C_1 & 0 & \dots & B_{i-1} & D_i^s & B_i & \dots \\ 0 & C_2 & 0 & \dots & B_i & D_{i+1}^s & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix}$$

図2 Aのブロック化 Fig. 2 Blocking method for matrix A.

$$D = \begin{bmatrix} D_1^s & & & 0 \\ & D_2^s & & \\ & & \dots & \\ 0 & & & \dots \end{bmatrix} \quad D^{-1} = \begin{bmatrix} (D_1^s)^{-1} & & & 0 \\ & (D_2^s)^{-1} & & \\ & & \dots & \\ 0 & & & (D_i^s)^{-1} \\ & & & & \dots & \\ & & & & & \dots & \end{bmatrix}$$

図3 ブロック行列Dとその逆行列D^{-1} Fig. 3 Blocked matrix D and its inverse matrix D^{-1}.

各部分行列の逆行列より構成される。

従来、こうしたブロック化は、SLOR 法⁹⁾や、共役勾配法の前処理におけるブロック・コレスキー分解法¹¹⁾において用いられている。しかし、それらはいずれも、ある座標方向の1列の格子点すべてをブロック化する

というような大規模なものであり、部分行列の逆行列を求めるのに直接法や近似法¹²⁾を、毎反復ごとに使う必要があった。この部分は、ベクトル処理や並列処理には向かず計算効率を悪化させる要因となる。そのため、ここでは、ベクトル処理や並列処理を考慮し、小規模なブロック化を行う手法をとる。以下、その方法を SSB (Small Scale Blocking) 法と呼ぶ。この SSB 法におけるブロック化においては、部分行列の逆行列 ($D_{i \times m}^{-1}$) の大きさを記憶可能な程度のものとする。すると、 D^{-1} は反復ごとに計算して求める必要はなく計算効率の面で有利であり、ポアソン方程式を等間隔格子で差分して解くような場合は、多くの部

$$D_{l \times m^S} = \begin{bmatrix} 1 & -0.25 & \cdot & \cdot & \cdot \\ -0.25 & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & -0.25 \\ \cdot & \cdot & \cdot & -0.25 & 1 \end{bmatrix}$$

(a) 部分行列 $D_{l \times m^S}$
(a) Sub-matrix $D_{l \times m^S}$

$$D_{l \times m} = \begin{bmatrix} D_{l \times m^S} & & 0 \\ & D_{l \times m^S} & \\ 0 & & D_{l \times m^S} \\ & & & \cdot \\ & & & & \cdot \end{bmatrix}$$

(b) ブロック化行列 $D_{l \times m}$
(b) Blocked matrix $D_{l \times m}$

図 5 図 4 のようなブロック化を行った場合の部分行列とブロック化行列
Fig. 5 Sub-matrix and blocked matrix when the grids are decomposed into $l \times m$ size blocks shown in Fig. 4.

分行列を同じものにとることが可能であるので、記憶容量の増大も抑えることができる。ベクトル処理、並列処理の面から見れば、部分行列を反復ごとに直接法などのベクトル化、並列化の困難な方法で解く必要がないため、完全に陽的な前処理となり、高いベクトル化率、並列化率が期待できる点が重要である。

いま、(2.0.1) 式を $N \times N$ の等間隔格子で離散化し、すべての格子点を $l \times m$ にブロック化し、格子点の順序づけを図 4 のようにすると、部分行列 $D_{i \times m}^S$ は図 5 (a) に示すようになる。また、そのブロック化行列は ($l \times m$) 行 ($l \times m$) 列の大きさの部分行列より構成される (図 5 (b))。例えば、同じ 4 行 4 列の大きさの部分行列への分解方法も、格子点を 4×1 にブロック化し図 6 (a) のように順序づけを行った場合、その部分行列 $D_{i \times 1}^S$ は図 7 (a) に示す行列となり、格子点を 2×2 にブロック化し図 6 (b) のような順序づけを

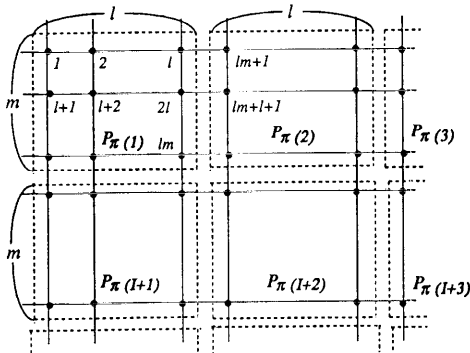


図 4 ポアソン方程式を等間隔格子で離散化し、格子点を $l \times m$ にブロック化した場合のブロックと格子点の順序づけ

Fig. 4 Blocking and ordering method when the Poisson equation is discretized on regular grids and the grids are decomposed into $l \times m$ size blocks.

$$D_{4 \times 1}^S = \begin{bmatrix} 1 & -0.25 & 0 & 0 \\ -0.25 & 1 & -0.25 & 0 \\ 0 & -0.25 & 1 & -0.25 \\ 0 & 0 & -0.25 & 1 \end{bmatrix}$$

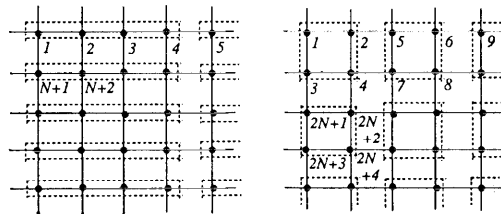
(a) 4×1 のブロック化における部分行列
(a) Sub-matrix in the 4×1 blocking

$$D_{2 \times 2}^S = \begin{bmatrix} 1 & -0.25 & -0.25 & 0 \\ -0.25 & 1 & 0 & -0.25 \\ -0.25 & 0 & 1 & -0.25 \\ 0 & -0.25 & -0.25 & 1 \end{bmatrix}$$

(b) 2×2 のブロック化における部分行列
(b) Sub-matrix in the 2×2 blocking

図 7 異なるブロック化手法による部分行列の違い

Fig. 7 Difference of sub-matrices for different blocking methods.



(a) 4×1 のブロック化 (b) 2×2 のブロック化
(a) 4×1 blocking (b) 2×2 blocking

図 6 ブロック化規模の大きさの同じブロック化におけるブロック化手法の違い

Fig. 6 Different methods for blocking with the same blocking size.

する場合、その部分行列 $D_{2 \times 2}$ は図 7 (b) に示す行列となる。

次に、このようなブロック化行列を用いた場合の R 行列の性質について調べる。

$A = D + (A - D)$ は行列 A についての正則分離⁹⁾ となるから、 R のスペクトル半径は

$$\rho(R) = \rho(D^{-1}(D - A)) = \frac{\rho(A^{-1}(D - A))}{1 + \rho(A^{-1}(D - A))} < 1 \quad (2.3.1)$$

となり、(2.2.3) 式は収束する。したがって、2.3 節より M_n は正定値行列であり、共役勾配法の前処理として適応可能であることもわかる。

また、 R に関するブロック有効グラフについて考えることにより、 R は次のような正と負の対となった固有値を持つことがわかる。

$$\lambda = 0, \pm\sigma_1, +\sigma_2, \dots, \pm\sigma_r, \quad |\sigma_1| \leq |\sigma_2| \leq \dots \leq |\sigma_r| \quad (2.3.2)$$

ここで、スペクトル半径の定義より $\rho(R) = |\sigma_r|$ である。

ここで、格子点を $l \times m$ にブロック化した場合のブロック化行列 $D_{l \times m}$ に対応するブロック・ヤコビ行列を $R_{l \times m}$ とする。すなわち、

$$R_{l \times m} = I - D_{l \times m}^{-1} A \quad (2.3.3)$$

とする。また、 u を l の倍数、 v を m の倍数であるとする

$$1 > \rho(R_{l \times m}) > \rho(R_{u \times v}) > 0 \quad (2.3.4)$$

より、例えば

$$1 > \rho(R_{1 \times 1}) > \rho(R_{2 \times 2}) > \rho(R_{4 \times 4}) > \dots > 0 \quad (2.3.5. a)$$

$$1 > \rho(R_{1 \times 1}) > \rho(R_{2 \times 1}) > \rho(R_{4 \times 1}) > \dots > 0 \quad (2.3.5. b)$$

$$1 > \rho(R_{1 \times 1}) > \rho(R_{2 \times 1}) > \rho(R_{2 \times 2}) > 0 \quad (2.3.5. c)$$

などとなり、ブロック化が大規模なほどそのブロック・ヤコビ行列のスペクトル半径は小さくなる。

2.4 共役勾配法の前処理因子としてのノイマン級数行列

ノイマン級数行列 M_n を前処理因子とした共役勾配法の収束効率を考える。 M_n は対称正定値であるから

$$M_n = V^T V \quad (2.4.1)$$

を満たす正則行列 V が存在する。 M_n により前処理を行った共役勾配法の収束効率は VAV^T の固有値によって決定される。しかし、 VAV^T と $M_n A$ は同じ固有値をもつから、 $M_n A$ の固有値について調べればよい。

$$\begin{aligned} M_n A &= f_n(R) D^{-1} A \\ &= f_n(R) (I - R) \\ &= I - R^{n+1} \end{aligned} \quad (2.4.2)$$

したがって、 $M_n A$ の固有値 μ と R の固有値 λ との間には

$$\mu = 1 - \lambda^{n+1} \quad (2.4.3)$$

の関係がある。この (2.4.3) 式より $M_n A$ の最小固有値、最大固有値は

$$\begin{aligned} \mu_{\min} &= 1 - \rho(R)^{n+1} \\ \mu_{\max} &= \begin{cases} 1 & \dots n : \text{Odd} \\ 1 + \rho(R)^{n+1} & \dots n : \text{Even} \end{cases} \end{aligned} \quad (2.4.4)$$

となる。また、 $M_n A$ の相異なる固有値の数は

$$L_n = \begin{cases} r+1 & \dots n : \text{Odd} \\ 2r+1 & \dots n : \text{Even} \end{cases} \quad (2.4.5)$$

となる。ノイマン級数行列を前処理とする共役勾配法においては、(2.4.4)、(2.4.5) 式により $M_n A$ の固有値分布が、 n が偶数であるか奇数であるかにより傾向が変化するため、その収束性も n の奇、偶により異なったものになると予想される。

2.5 多項式行列

ここでは、前処理因子として (2.2.4) 式の $f_n(x)$ を用いる代わりに

$$g_n(x) = \alpha_0 + \alpha_1 x + \dots + \alpha_n x^n \quad (2.5.1)$$

を用い、

$$\begin{aligned} N_n &= (\alpha_0 I + \alpha_1 R + \dots + \alpha_n R^n) D^{-1} \\ &= g_n(R) D^{-1} \end{aligned} \quad (2.5.2)$$

を共役勾配法の前処理行列として用いる。以下、この N_n を n 次の多項式行列と呼ぶ。

ノイマン級数行列 M_n は、 n を十分大きく取れば、理論的には A^{-1} へ収束していくが、共役勾配法の前処理として用いる場合は、計算量を考えるとそれほど大きな n は使えないので、 M_n の A^{-1} に対する近似度はあまり良くない。そうした場合、ノイマン級数行列を前処理として用いることで共役勾配法の収束性はそれほど改善されない。一方、行列多項式 N_n を前処理行列として用いる共役勾配法においては、その収束性が改善されるように、 $N_n A$ の固有値分布を多項式の係数列 $\alpha_0, \alpha_1, \dots, \alpha_n$ により調整する。その調整は n が比較的小さくとも可能であり、最適な係数列を選べば非常に速く収束する可能性がある。

2.5.1 最適化手法

ここでは、 $g_n(x)$ の最適化手法について述べる。

$g_n(x)$ の最適化では、従来の共役勾配法に関する経験則を実現するように行う。その経験則は「共役勾配

法においては、なるべく多くの固有値が1付近に分布しており、かつ、最小固有値がなるべく大きいような固有値分布のものがよいである。そのため、元の (\mathbf{R}) 固有値分布は最小、最大固有値 $[-\rho, \rho]$ の間で一様で連続的に分布していると仮定し、前処理後の ($\mathbf{N}_n \mathbf{A}$) 固有値分布 $\mu(\lambda)$ を1に近づけるように

$$G = \int_{-\rho}^{\rho} \{1 - \mu(\lambda)\}^2 w(\lambda) d\lambda \quad (2.5.3)$$

を最小化する。ここで $w(\lambda)$ は、スペクトルの一部を強調する重み関数である。ここで (2.5.2) 式と同様にして $\mathbf{N}_n \mathbf{A} = g_n(\mathbf{R})(\mathbf{I} - \mathbf{R})$ となるから、(2.5.1) 式より

$$\begin{aligned} 1 - \mu(x) &= 1 - g_n(x)(1-x) \\ &= 1 - \alpha_0(1-x) - \alpha_1 x(1-x) - \dots \\ &\quad - \alpha_n x^n(1-x) \end{aligned} \quad (2.5.4)$$

となり、 G を最小化する係数に対し、

$$\frac{\partial G}{\partial \alpha_i} = -2 \int_{-\rho}^{\rho} \{1 - \mu(\lambda)\} \lambda^i (1-\lambda) w(\lambda) d\lambda = 0 \quad (2.5.5)$$

が成り立つ。これより、求める係数に対する連立一次方程式

$$T_j \alpha_i = t_j \quad (2.5.6)$$

を得る。ただし、

$$T_j = \int_{-\rho}^{\rho} \lambda^{i+j} (1-\lambda)^2 w(\lambda) d\lambda \quad (2.5.7)$$

$$t_j = \int_{-\rho}^{\rho} \lambda^j (1-\lambda) w(\lambda) d\lambda \quad (2.5.8)$$

である。

したがって、 G を最小化する係数列は (2.5.6) 式の連立一次方程式を解くことにより求められる。

また、 $\mathbf{N}_n \mathbf{A}$ の固有値は $\lambda = -\rho$ または $\lambda = \rho$ のときに最小となるので、重み関数としては一般にヤコビの重み関数

$$w(\lambda) = (\rho - \lambda)^\alpha (\rho + \lambda)^\beta \quad (2.5.9)$$

が用いられる。特に、 \mathbf{R} 行列が点ヤコビ行列の場合の最適化は O. G. Johnson ら⁶⁾ や Y. Saad⁷⁾ により行われている。

一般的にはスペクトル半径 ρ は解析的に求められないことが多い。この問題を解決するため、近年 Ashby らは多項式の係数を動的に適応制御する手法を提案している¹²⁾。しかし、解析する問題が十分大規模であれば

$$\rho \cong 1 \quad (2.5.10)$$

の仮定で十分であると考えられるため、ここでは (2.5.10) 式を用いるものとする。

3. 結果と考察

3.1 基本的性質

ここではまず SSB 法を用いたブロック化行列の多項式を前処理因子とする共役勾配法の基本的性質を簡単な問題で確認する。

3.1.1 ブロック化の影響

まず、ブロック化の規模および方法の固有値分布に与える影響について調べる。

2次元正方空間内のポアソン方程式

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = \psi \text{ in } [0, 1] \times [0, 1] \quad (3.1.1)$$

をディリクレ境界問題として考える。この方程式を差分法により離散化し、 20×20 の等間隔格子で解く場合の、行列 $\mathbf{D}^{-1} \mathbf{A}$ の固有値を QR 法によって求めた。格子点のブロック化を $1 \times 1, 2 \times 1, 4 \times 1, 2 \times 2, 4 \times 4$ とした場合の $\mathbf{D}^{-1} \mathbf{A}$ の固有値のうち、最小なものから5個、最大なものから5個をとりだしたものを表1に示す。

ここで、 $\mathbf{D}^{-1} \mathbf{A}$ の固有値 μ と \mathbf{R} の固有値 λ の間には $\lambda = 1 - \mu$ の関係が成立していることに注意すれば (2.3.2) 式の成立が確認できる。また、(2.3.5) 式も成り立っている。ここで、部分行列の大きさが等しい 4×1 と 2×2 のブロック化に注目すると、

$$\rho(\mathbf{R}_{4 \times 1}) > \rho(\mathbf{R}_{2 \times 2}) \quad (3.1.2)$$

となっている。この結果より、大きさの等しいブロック化においてはそのブロックが正方形に近いほうが固有値分布を改善する効果の高いことがわかる。

3.1.2 多項式の次数の影響

次に、ブロック化行列のノイマン級数および多項式の次数の影響を調べる。

表1 様々なブロック化による $\mathbf{D}^{-1} \mathbf{A}$ の固有値の変化
Table 1 Eigenvalue of matrix $\mathbf{D}^{-1} \mathbf{A}$ with various blocking methods.

	$\mathbf{A} = \mathbf{D}_{1 \times 1}^{-1} \mathbf{A}$	$\mathbf{D}_{2 \times 1}^{-1} \mathbf{A}$	$\mathbf{D}_{4 \times 1}^{-1} \mathbf{A}$	$\mathbf{D}_{2 \times 2}^{-1} \mathbf{A}$	$\mathbf{D}_{4 \times 4}^{-1} \mathbf{A}$
1	0.01171	0.01484	0.01775	0.02211	0.04295
2	0.02780	0.03657	0.04338	0.05419	0.1007
3	0.02780	0.03657	0.04412	0.05419	0.1008
4	0.04443	0.05843	0.06920	0.08526	0.1516
5	0.05510	0.07126	0.08341	0.1047	0.1805
396	1.9449	1.9287	1.9166	1.8953	1.8196
397	1.9556	1.9416	1.9308	1.9147	1.8487
398	1.9722	1.9631	1.9559	1.9458	1.8993
399	1.9722	1.9634	1.9566	1.9458	1.8994
400	1.9888	1.9852	1.9822	1.9779	1.9571

対象とする問題は 3.1.1 項と同じとし、ブロック化行列として、 2×2 のブロック化を行った $D_{2 \times 2}$ を用いる。級数は $n=0, 1, 2, 3, 4$ のノイマン級数 $f_n(x)$ を用い、 $M_n A = f_n(R) D^{-1} A$ の固有値分布を QR 法により調べた結果を図 8 に示す。図 8 (b) より級数の次数が 0 次の場合、すなわち前処理因子として D^{-1} を用いた場合、最大、最小付近の固有値分布が改良され、固有値の縮約も多くなることがわかる。また、ノイマン級数の次数が奇数、偶数で固有値分布の特性が変化しており、2.4 節の予測を裏付けている。また、次数が増えるにつれ最小付近の固有値分布が徐々に改良され、1 付近に集約していくことも確認できる。

同様にブロック化行列として $D_{2 \times 2}$ を用い、ルジャンドルの重み付け関数 ((2.5.9) 式) において $\alpha=0, \beta=0$ を用いて最適化された多項式を使用し、上記と同様の解析を行った。ただし、0 次の多項式を用いたものはノイマン級数を用いたものと同じ結果になるので省略した。結果を図 9 に示す。まず、多項式の次数

が 1 次の場合の固有値分布 (図 9 (a)) をノイマン級数を用いたもの (図 8 (c)) と比較すると、多項式を用いたもののほうが、0 近傍の固有値を 0 より隔離する効果が高く、全体の固有値分布を 1 に近づける作用も大きいことがわかる。この傾向は、次数が高くなるほどより明確に確認できる。また、多項式を用いた場合の固有値分布は、ノイマン級数を用いたものと異なり、次数が偶数であるか奇数であるかにかかわらずほぼ同様な形をしていることも注目される。

このように、重み関数を用いて最適化された多項式を用いた前処理方法は、元の固有値分布を、共役勾配法で解くのに適した形に修正することができ、その収束性もノイマン級数を用いたものに比べてさらに優れたものになると思われる。

3.2 計算効率の考察

3.2.1 ディリクレ境界問題

ここではまず、ブロック化が収束性に与える影響を調べるため、3.1 節で取り上げた 2 次元正方形空間内の

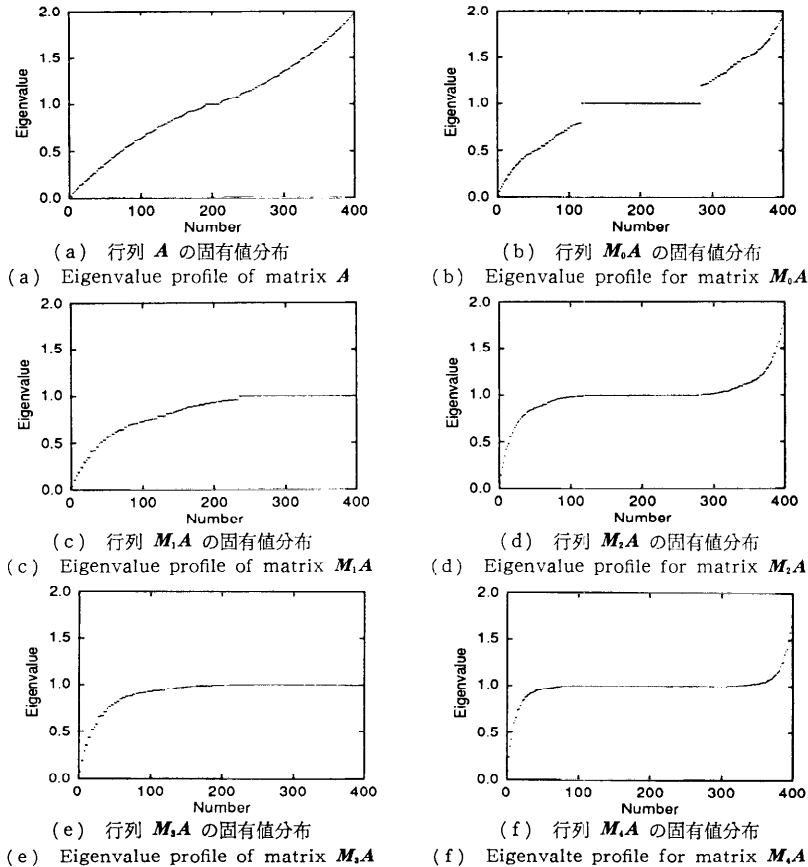


図 8 n 次のノイマン級数を用いた前処理後の行列 $M_n A$ の固有値分布
Fig. 8 Eigenvalue profile of preconditioned matrix $M_n A$ using the n -th order Neumann series.

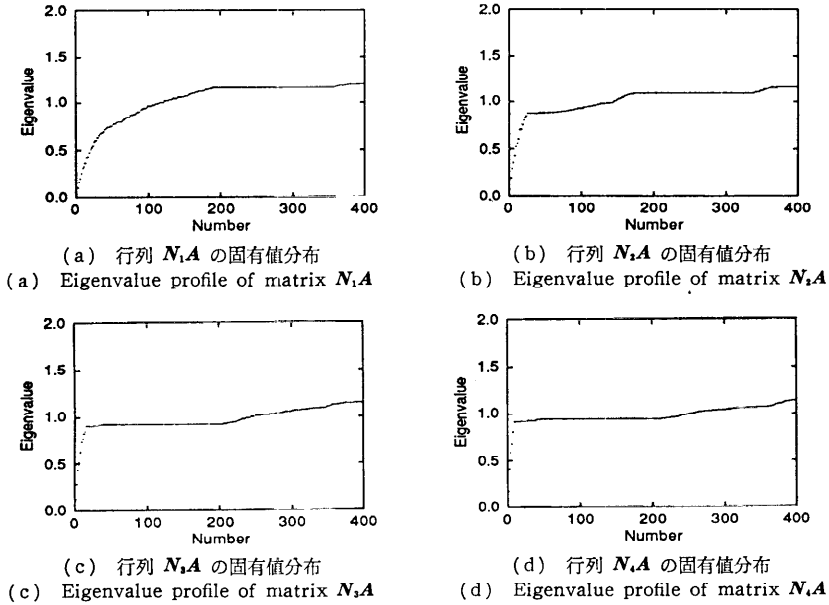


図 9 ルジャンドル重み関数 (式(2.5.9)において $\alpha=0, \beta=0$) を用いて最適化した n 次の一般級数による前処理後の行列 $N_n A$ の固有値分布

Fig. 9 Eigenvalue profile of the preconditioned matrix $N_n A$ using n -th order general polynomial optimized by the Legendre weight function which means $\alpha=0, \beta=0$ in Eq. (2.5.9).

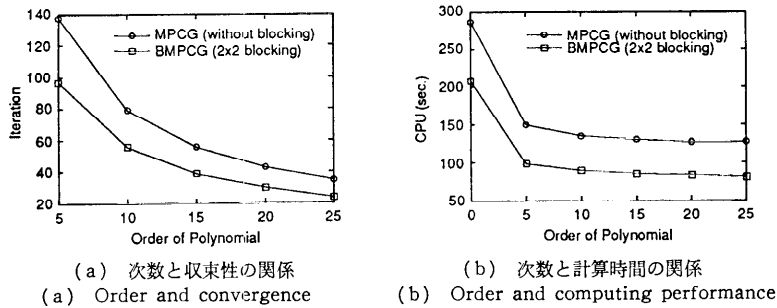


図 10 ディリクレ境界問題におけるブロック化の有無による級数の次数と収束性および計算時間の比較 (ルジャンドル重み関数 (式(2.5.9)において $\alpha=0, \beta=0$) により最適化した多項式を使用)

Fig. 10 Comparison of convergence (a) or computing time (b) with or without blocking in Dirichlet boundary problem (using the polynomial optimized by the Legendre weight function which means $\alpha=0, \beta=0$ in Eq. (2.5.9)).

ポアソン方程式のディリクレ境界問題を解くこととする。

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0 \text{ in } [0, 1] \times [0, 1]$$

with

$$\phi = \begin{cases} 1 & \text{on } y=1 \\ 0 & \text{other boundaries} \end{cases} \quad (3.2.1)$$

用いる格子は、実際の解析で見られるような大規模

な行列を考慮し、 240×240 の等間隔格子とした。ブロック化は図 6 (b) に示す 2×2 単位で行い、ブロック化しない場合と比較を行った。この場合、ブロック化行列の部分行列はすべて図 7 (b) と同じものとなる。計算機としては逐次処理型の計算機である AS 4080 を使い、収束判定条件は以下のとおりとした。

$$\frac{\|Ax_n - b\|_2}{\|b\|_2} < 1.0 \times 10^{-8} \quad (3.2.2)$$

収束性および計算時間の比較を図 10 に示す。また、同じ問題を ICCG 法で解いた場合、反復数は 204 回、計算時間は 111.3 秒であった。

図 10 より、いずれの次数においても、 2×2 のブロック化を行った手法は、ブロック化しない場合に比べ 10 回以上反復数が少ないことがわかる。計算時間においては、いずれの次数においても 40 秒程度の差があり、これはブロック化により約 30% 程度の計算時間が削減できることを示している。また、次数が 25 次の場合の計算時間について見てみると、ブロック化を行わない場合が 128.3 秒で、ICCG 法よりも計算時間が必要となることがわかる。一方、ブロック化を行った場合の計算時間は 81.0 秒であり、ブロック化を行うことにより ICCG 法よりの 27% (30 秒) 程度速く計算できる。

以上の計算が、逐次処理型の計算機上で行われていることを考慮すれば、この結果は、本解法の収束性の

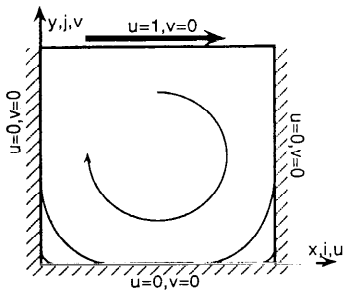


図 11 BMPCG 法の計算効率の確認のために用いた 2 次元正方空間内の流れ問題

Fig. 11 Considered problem of two dimensional flow in a square cavity to confirm the convergence performance of BMPCG.

良さを裏付けるものと考えられる。

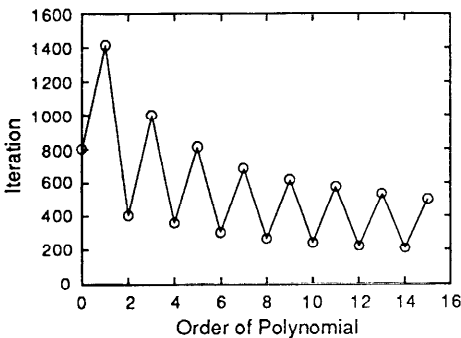
3.2.2 ノイマン境界問題

次に、より具体的な問題を対象として、本手法の計算効率を考察する。

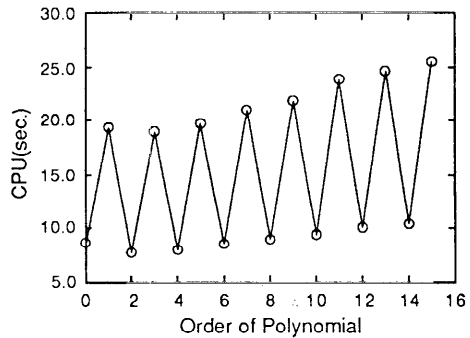
対象とする問題は、図 11 のような 2 次元正方空間内の流れとし、この問題をスタガード格子系で、SMAC 法により解析する際の、圧力補正式に関するポアソン方程式を解くものとする。収束条件、格子およびブロック化は上述の問題と同じ方法を用いた。本来、この問題はノイマン境界問題であるため、ブロック化を行う場合は、境界に面した部分だけ異なった種類の部分行列より成るブロック化行列を用いる必要があるが、ここではその近似行列として、3.1.1 項と同じブロック化行列を用いた。解析は、計算開始直後の 1 時間ステップにおけるポアソン方程式を解き、それに必要な反復数、計算時間を計測した。また、計算機としてはベクトル計算機である Cray Y-MP (1CPU) を用いた。

まず、ブロック化を行わずにノイマン級数を用い、その次数を変えたときの収束に必要な反復数と計算時間を調べた結果を図 12 に示す。この図より、2.5 節の予想が裏付けられており、級数の次数が偶数、奇数で収束性が異なっている。この場合、次数は偶数のほうが有利であることがわかる。一方、計算時間が最小となる級数の最適次数は 2 次するときであるが、その場合、0 次の場合と計算時間としてはほとんど変わっておらず、ノイマン級数を用いた前処理による計算効率の向上は期待できない。

次に、 2×2 のブロック化を施したノイマン級数を用いた前処理による結果を図 13 に示す。まず、級数



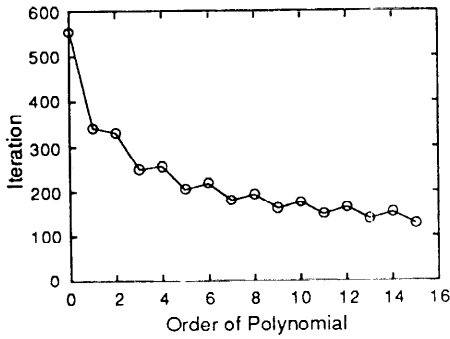
(a) 次数と収束性の関係
(a) Order and convergence



(b) 次数と計算時間の関係
(b) Order and computing performance

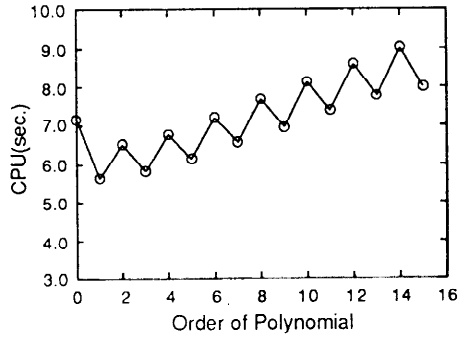
図 12 ブロック化を行わないノイマン級数を用いた場合の級数の次数と収束性および計算時間の関係

Fig. 12 Relation between order and convergence (a) or computing time (b) using the Neumann series without blocking.



(a) 次数と収束性の関係

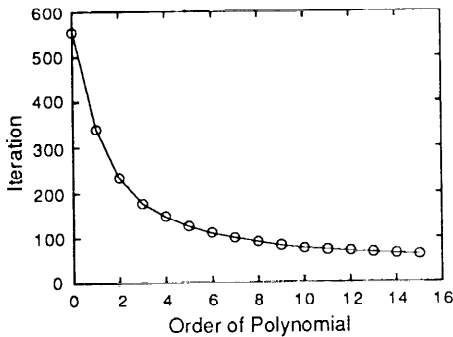
(a) Order and convergence



(b) 次数と計算時間の関係

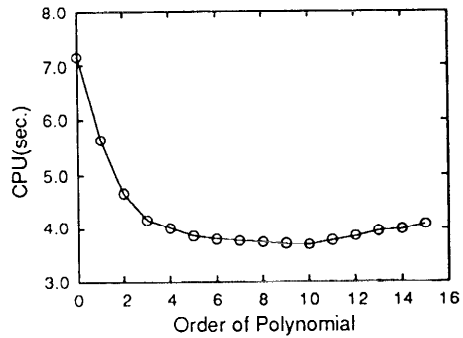
(b) Order and computing performance

図 13 2×2 のブロック化とノイマン級数を用いた場合の級数の次数と収束性および計算時間の関係
Fig. 13 Relation between order and convergence (a) or computing time (b) using the Neumann series with the 2×2 blocking.



(a) 次数と収束性の関係

(a) Order and convergence



(b) 次数と計算時間の関係

(b) Order and computing performance

図 14 2×2 のブロック化とルジャンドル重み関数により最適化した多項式を用いた場合の次数と収束性および計算時間の関係

Fig. 14 Relation between order and convergence (a) or computing time (b) with the 2×2 blocking and the polynomial optimized by the Legendre weight function.

の次数が0次の点に注目すると、ブロック化行列だけによる前処理 ($M_0 = D_{2 \times 2}$) でも計算効率向上することがわかる。また、この場合も、級数の次数の奇偶で収束性が異なっているが、上の場合ほど顕著ではなく、ここでは奇数のほうが多少有利である。ブロック化を施した場合、計算時間が最小となる級数の最適次数は1次のときである。その場合の計算時間は、級数の次数が0次の場合よりも35%程度少なく、ブロック化により、ノイマン級数を用いた前処理も収束性の改善に有効であることが確認できる。

次に、2×2のブロック化を行った行列の、ルジャンドルの重み付け関数により固有値分布の最適化を行った多項式を前処理因子として使用した場合の結果を図14に示す。一般多項式を用いた前処理によるものは、ノイマン級数を使用した場合と違い、次数が偶数か奇

数かに関係なく、次数を増やすに従い、収束に必要な反復数が単調に減少している(図14(a))。また、図14(b)よりわかるように、次数が10次を越えると計算時間が増加し始めている点だが、3.2.1項のディリクレ境界問題の場合と異なっている。それは、ブロック化行列として近似行列を用いているためである。しかし、ノイマン級数における最適次数も含め、このような級数の最適次数は不変的なものではなく、次数を上げたことによる収束性の改善、次数を上げたことによる計算量の増加、計算機の性能の関係によって決定されるものであり絶対的なものではない。

最後に、Cray Y-MP (1 CPU) 上での最適次数での反復数、計算時間および計算速度を表2にまとめる。この表より、ブロック化の手法を用いないノイマン級数を使った前処理方法は、超平面順序づけを用いた不

表 2 Cray Y-MP 上での多項式の最適次数における収束に必要な反復回数, 計算時間および計算速度
Table 2 Iteration number and computing time for the matrix solver to converge and the computing speed with the polynomial having optimized order for Cray Y-MP.

	ICCG (Hyperplane Ordering)	MPCG (no blocking and Neumann series)	BMPCG (2×2 blocking and Neumann series)	MPCG (no blocking and Legendre Weighted Polynomial)	BMPCG (2×2 blocking and Legendre Weighted Polynomial)
級数の次数	—	2 次	1 次	16 次	10 次
反 復 数	324	466	341	67	78
計算時間(秒)	7.76	7.77	5.63	3.93	3.71
計算速度 (MFlops)	108	185	192	181	200

完全コレスキー分解を用いたものと計算時間はほとんど同じであるが、反復数はかなり多くなっている。一方、ブロック化を施した場合は、超平面順序づけを用いた不完全コレスキー分解を用いたものより反復数は若干多いものの、計算時間はかなり短縮されており、収束性の良さを裏付けている。さらに、ルジャンドル重み付け関数で最適化を行った多項式による前処理を用いた結果では、不完全コレスキー分解を用いる場合に比べ、反復数は約 1/4 に減り、計算時間も 47% に減少しており、その前処理法の収束性、計算効率ともに非常に優れていることを確認できる。しかし、ブロック化を行わない手法に比べ、最適次数が低く、計算時間も 5% 程度しか短縮されていない。これは、ブロック化行列として、その近似行列を用いているためであり、問題に正確に対応したブロック化行列を用いれば計算時間はさらに短縮されると考えられる。

数値解析手法の面から見れば、こうした行列解法をベクトル計算機や並列計算機上で処理する際に、このような級数を用いる前処理方法は、不完全コレスキー分解を用いるような場合と比べ、超平面順序づけといった特殊な手法を必要としない点も注目に値する。

4. 結 論

本論文では、小規模ブロック化行列の級数を用いた、共役勾配法のための前処理方法を提案した。この前処理方法では、小規模ブロック化により、一般のブロック化で必要な反復ごとの直接解法や近似解法が不要で、データの再帰関係を免れることができる。そのため、ベクトル計算や並列計算、特に今後さらに注目されるであろう超並列計算に適しており将来の行列解法として有望である。ここでは特に逐次処理型およびベクトル処理型の計算機を対象としてその収束性や処理効率を調べ、収束性、ベクトル処理効率の良さなどを確認した。ベクトル計算機は SIMD・共有メモリ型

の中規模な並列計算機と考えることができるから、本手法は超並列計算機では、さらに有利に処理できると予想される。

本研究で提案したブロック化行列の多項式を用いる方法では、そのブロック化の方法、多項式の最適化の方法、多項式の次数の調整など非常に柔軟性が高く、解析する問題や計算機の特性によってそれらの方法を適切に使い分け計算効率を高く調整することができることも重要である。今後は、これらの特徴を活かし、超並列計算機上での計算性能の解析、計算効率の良いブロック化手法や多項式の新しい最適化手法の開発、3次元問題への適応、非対称行列への適応などについて研究を進めていく予定である。

参 考 文 献

- 1) Sethian, J. A., Brunet, J. P., Greenberg, G. and Mesirov, J. P.: Two-Dimensional, Viscous, Incompressible Flow in Complex Geometries on a Massively Parallel Processor, *J. Comput. Phys.*, Vol. 101, pp. 185-206 (1992).
- 2) 田中伸厚, 寺坂晴夫: 超並列計算機によるバックステップ流れ解析, 日本機械学会第 70 期通常総会講演会講演論文集(II), pp. 609-611 (1993).
- 3) Tanaka, N. and Terasaka, H.: A Numerical Study of Incompressible Viscous Flow with Massively Parallel Processors, *Comput. Fluid Dynamics J.*, Vol. 2, No. 2, pp. 145-160 (1993).
- 4) 田中伸厚, 寺坂晴夫: 超並列計算機における大規模疎行列の解法, 日本原子力学会 1993 年春の年会要旨集, p. 219 (1993).
- 5) Dubois, P., Greenbaum, A. and Rodrigue, G. H.: Approximating the Inverse of a Matrix for Use in Iterative Algorithms on Vector Processors, *Computing*, Vol. 22, pp. 257-268 (1979).
- 6) Johnson, O. G., Micheli, C. A. and Paul, G.: Polynomial Preconditioning for Conjugate Gradient Calculations, *SIMA J. Numer. Anal.* Vol. 20, pp. 362-376 (1983).

- 7) Saad, Y.: Practical Use of Polynomial Preconditionings for the Conjugate Gradient Method, *SIAM J. Sci. Stat. Comput.*, Vol. 6, No. 4, pp. 865-881 (1985).
- 8) Dongarra, J. J., Duff, L. S., Sorensen, D. C. and van der Vorst, H. A.: *Solving Linear Systems on Vector and Shared Memory Computers*, SIAM (1991).
- 9) Varga, R.S.: *Matrix Iterative Analysis*, Prentice-Hall (1962).
- 10) Meijerink, J. A. and van der Vorst, H. A.: An Iterative Solution Method for Linear Systems of which the Coefficient Matrix is a Symmetric M-Matrix, *Math. Comp.*, Vol. 31, No. 137, pp. 148-162 (1977).
- 11) Concus, P., Golub, G.H. and Meurant, G.: Block Preconditioning for the Conjugate Gradient Method, *SIAM J. Sci. Stat. Comput.*, Vol. 6, No. 1, pp. 220-252 (1985).
- 12) Ashby, A.: A Comparison of Adaptive Chebyshev and Least Squares Polynomial Preconditioning for Hermitian Positive Definite Linear Systems, *SIAM J. Sci. Stat. Comput.*, Vol. 13, No. 1, pp. 1-29 (1992).

付 録

A. (2.3.2) 式の証明

図4に示されるような長方形のブロック化を行う場合の R に関するブロック有効グラフ $G_r(R)^{9)}$ を図15に示す。

図15より, $G_r(R)$ は強連結であることが分かる。また, 自分自身への道のりの最大公約数は2であるから, $G_r(R)$ は2の巡回グラフであることがわかる。よって, R は2の弱巡回であり, R の固有関数は

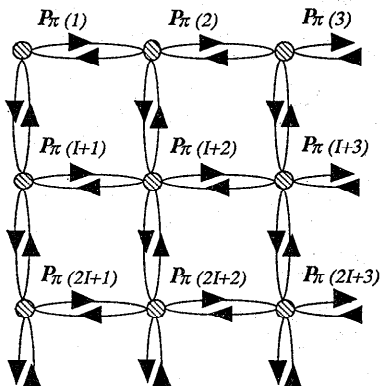


図15 図4のブロック化に対応した R に関するブロック有向グラフ $G_r(R)$ (\leftarrow は有向路を表す)
 Fig. 15 Block directed graph $G_r(R)$ corresponding to the blocking method shown in Fig. 4 (\leftarrow means a directed path).

$$\phi(\lambda) = \det(\lambda I - R) = \lambda^r \prod_{i=1}^r (\lambda^2 - \sigma_i^2) \tag{A.1}$$

と表される。したがって, R は

$$\lambda = 0, \pm\sigma_1, \pm\sigma_2, \dots, \pm\sigma_r, \quad |\sigma_1| \leq |\sigma_2| \leq \dots \leq |\sigma_r| \tag{2.3.2}$$

のような正と負の対となった固有値を持つことがわかる。

B. (2.3.4) 式の証明

次の定理を用いる。

(定理1) 既約な正方行列 H が $H \geq 0$ であれば, H のいずれの要素が増加しても $\rho(H)$ は増加する⁹⁾。

まず, ポアソン方程式を離散化して得られる係数行列 $A = (a_{ij})$ は対称正定値で, すべての要素に対し, $i \neq j$ ならば $a_{ij} < 0$ であるから

$$A^{-1} > 0 \tag{B.1}$$

である。ここで, u を l の倍数, v を m の倍数であるとする

$$D_{l \times m} - A > D_{u \times v} - A > 0 \tag{B.2}$$

なることから, (B.1) とより

$$A^{-1}(D_{l \times m} - A) > A^{-1}(D_{u \times v} - A) > 0 \tag{B.3}$$

となる。ここで定理1を用いれば

$$\rho(A^{-1} - (D_{l \times m} - A)) > \rho(A^{-1} - (D_{u \times v} - A)) > 0 \tag{B.4}$$

であるから, (2.3.1) 式より

$$1 > \rho(R_{l \times m}) > \rho(R_{u \times v}) > 0. \tag{2.3.4}$$

であることがわかる。

(平成5年10月25日受付)

(平成6年4月21日採録)



田中 伸厚 (正会員)

1964年生。1992年東京大学大学院博士課程修了。工学博士。(株)東芝原子力技術研究所に勤務し, 伝熱流動解析の仕事に従事。特に, 新しい数値流体解析手法やその並列処理を中心として研究を行っている。日本機械学会, 日本原子力学会, 日本応用数理学会各会員。



寺坂 晴夫

1950年生。1979年大阪大学大学院博士課程修了。工学博士。同年(株)東芝入社。原子力技術研究所に勤務。原子力分野の伝熱流動解析, 評価ほかに従事。日本機械学会, 日本原子力学会各会員。