SOR法の並列ベクトル計算*

横 川 三 津 夫^{*1}

Parallel-Vector Calculation of the SOR Method

Mitsuo YOKOKAWA

The successive overrelaxation method, also called the SOR method, is one of the iterative methods for solving a linear system of equations, and has been used in many programs. With the advent of vector processors, the SOR method is executed efficiently in parallel with the red-black or hyperplane ordering on vector processors. In this paper, a parallel scheme termed the 4-color SOR method is revised and compared with the natural and red-black SOR methods on a multiprocessor system. The 4-color SOR method has the highest parallel performance of the three. The parallel-vector calculation of the 4-color method with 4 processors is about 10 times faster than the scalar one with one processor.

Key Words: Finite Difference Method, Numerical Analysis, Computational Mechanics, Successive Overrelaxation Method, 4-Color Ordering, Parallel-Vector Calculation

1. は じ め に

計算機の進歩に伴い、大規模な数値シミュレーションが可能となってきた。連立一次方程式を解く過程は、数値シミュレーションにおいて特に重要なツールであり、その高速解法が要求されている。

逐次過大緩和法(SOR法)は、連立一次方程式の反復解法のひとつであるが、コーディングの容易さ、収束に関する理論があること⁽¹⁾などの理由によって、さまざまな分野でよく使われる解法である。このSOR法は、従来 natural オーダリングと呼ばれる未知変数の順序付けを用いてスカラ計算機上で実行されていたが、ベクトル計算機の出現以降、red-black オーダリングや hyperplane オーダリングを用いたSOR法に変更され高速に実行されている。また最近では、並列計算機の開発が進み、SOR法の並列実行が可能となってきた。SOR法の並列処理方式はいくつか提案されている^{(2)~(4)}。

本報では、三次元ラプラス演算子を7点差分で離散化する時に得られる連立一次方程式を解く場合、並列

ベクトル計算可能な4-color SOR法⁽⁵⁾が他のSOR法と比較して並列処理効率や実行時間の面で有効であることを示す。実際に、マルチプロセッサシステム上で natural, red-black SOR法と比較し、計算時間や対スカラ速度向上率を検討した。

2. 4-color SOR法

単位立方領域 Ω 上で、次のポアソン方程式のディリクレ問題を考える。

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} &= g(x, y, z) & \text{in } \Omega \\ u(x, y, z) &= \varphi(x, y, z) & \text{on } \partial\Omega \end{aligned} \quad \dots\dots\dots (1)$$

このような方程式は、均質場における定常熱拡散問題や、電場ポテンシャル、流れ場の圧力分布を求める場合に得られる。この領域を x, y, z 方向にそれぞれ l, m, n 等分した離散化格子(図1)を考え、格子点 (i, j, k) での u の値を $u_{i,j,k}$ とすれば、式(1)を7点差分で離散化した式は点 (i, j, k) で次のようになる。

$$\begin{aligned} a_{i,j,k} u_{i,j,k-1} + b_{i,j,k} u_{i,j,k+1} + c_{i,j,k} u_{i-1,j,k} \\ + d_{i,j,k} u_{i,j,k} + e_{i,j,k} u_{i+1,j,k} + f_{i,j,k} u_{i,j,k+1} \\ + g_{i,j,k} u_{i,j,k+1} = z_{i,j,k} \dots\dots\dots (2) \end{aligned} \quad (i=1 \sim l-1, j=1 \sim m-1, k=1 \sim n-1)$$

* 平成元年11月28日 第2回計算力学講演会において講演、原稿受付 平成元年5月17日。

^{*1} 正員、日本原子力研究所 (〒319-11 茨城県那珂郡東海村)。

図 1 計算格子 ($l=m=n=4$)

図 2 4-color オーダリング

式(2)は全部で $(l-1) \times (m-1) \times (n-1)$ 個あり, $u_{i,j,k}$ を適当に並べた変数ベクトル $\mathbf{u}$ を考えると, 行列形式で

$$A\mathbf{u} = \mathbf{z} \quad \dots\dots\dots (3)$$

と書くことができる. 行列 A の非ゼロ要素パターンは, 変数 $u_{i,j,k}$ のオーダリング(並べ方)をいろいろ変化させることによって変わり, それによって SOR スキームも変化することになる. red-black SOR 法は, 計算格子を赤と黒の2色でグループ分けした時に得られる.

いま, この離散化格子の z 方向のインデックス k に対して, 図2のように奇数平面を赤と黒, 偶数平面を黄と緑で一松模様状に色分けしたとする. すなわち,

- k : 奇数 $i+j$: 偶数……赤 (○印)
- k : 奇数 $i+j$: 奇数……黒 (●印)
- k : 偶数 $i+j$: 偶数……黄 (□印)
- k : 偶数 $i+j$: 奇数……緑 (■印)

とする. この色分けに対して, 変数ベクトルを次のようにする.

$$\mathbf{u}^{4c} = \begin{bmatrix} \mathbf{u}_R \\ \mathbf{u}_B \\ \mathbf{u}_Y \\ \mathbf{u}_G \end{bmatrix} \quad \dots\dots\dots (4)$$

ここで, $\mathbf{u}_R$, $\mathbf{u}_B$, $\mathbf{u}_Y$, $\mathbf{u}_G$ はそれぞれ赤, 黒, 黄, 緑の

図 3 非ゼロ要素パターン

4色で色分けした格子変数を並べたベクトルである. このオーダリングを 4-color オーダリングと呼ぶことにする. このとき, 7点差分によって得られる連立一次方程式を,

$$A^{4c}\mathbf{u}^{4c} = \mathbf{z}^{4c} \quad \dots\dots\dots (5)$$

とすると, 係数行列 A^{4c} の非ゼロ要素は図3のようになる. 方程式(5)に対して, SOR 法を適用すると加速係数 ω を用いて, 次の漸化式が得られる. ただし, 簡単のために行列 A^{4c} の対角成分は1にスケールされているとする.

$$\begin{aligned} \mathbf{u}^{n+1} &= \mathcal{L}_{4c}(\omega)\mathbf{u}^n + \omega(I - \omega L^{4c})^{-1}\mathbf{z}^{4c} \\ \mathcal{L}_{4c}(\omega) &= (I - \omega L^{4c})^{-1}\{(1-\omega)I + \omega U^{4c}\} \\ A^{4c} &= I - L^{4c} - U^{4c} \quad \dots\dots\dots (6) \end{aligned}$$

$\mathbf{u}^n$ は第 n 近似解, L^{4c} , U^{4c} はそれぞれ A^{4c} の狭義下三角行列, 狭義上三角行列である. これを 4-color SOR 法と呼ぶことにする. $\mathcal{L}_{4c}(\omega)$ は SOR 反復行列と呼ばれるが, ω を変化させることによって $\mathcal{L}_{4c}(\omega)$ のスペクトル半径を最小にすることができ, 収束を速くすることができる. このときの ω を最適加速係数という. 4-color SOR 法の最適加速係数は, natural SOR 法, red-black SOR 法と等しい⁽⁵⁾.

3. 並列計算機上での計算方法

ベクトル機能をもつプロセッサを並列に複数台並べた並列計算機を考える。各プロセッサは、ローカル・メモリをもち、各プロセッサ間は通信バスをもっているとする。このような計算機の上では、プログラムの中に潜在するベクトル処理と並列処理の相補関係を生かすことができる⁽⁶⁾。特に、多重ループにおいて、外側ループと内側ループにデータ依存関係がない場合は、外側ループを並列処理、内側ループをベクトル処理するように、ループの計算を自然にこの計算機に割当てることができる。この場合、ベクトル処理による速度向上率と並列処理による速度向上率の積である速度向上率が期待できる。

4-color SOR法をこのような計算機で実行する場合を考えよう。1回の更新過程において、ある色の変数ベクトルは他の色の変数だけで更新できる。したがって、このSOR法のアルゴリズムは次のようになる。

(4-color SOR法の収束ループ)

```

for all odd k
  for any i, j such that i+j is even
    (赤変数の更新)
  for any i, j such that i+j is odd
    (黒変数の更新)
for all even k
  for any i, j such that i+j is even
    (黄変数の更新)
  for any i, j such that i+j is odd
    (緑変数の更新)

```

(収束判定)

このアルゴリズムの外側kループを各プロセッサに割り当て内側ループをベクトル計算する。各内側ループは他のループとデータ依存関係がないから、独立にベクトル計算可能である。プロセッサ間の通信バスは更新されたデータを転送するために使われる。

4. 数値計算結果

4-color SOR法をマルチプロセッサシステム

Alliant FX/8上で最大4台のプロセッサを用いて実行させ並列処理効率を調べた。FX/8は各プロセッサがベクトル演算機能を持っている。このとき、natural SOR法とred-black SOR法との比較も行った。

テスト問題は、ラプラス方程式のディリクレ問題である。

$$\Omega = [0, 1] \times [0, 1] \times [0, 1]$$

$$\nabla^2 u = 0 \quad \text{in } \Omega$$

$$u(x, y, z) = \begin{cases} \sin(\pi x) \sin(\pi y) & (x, y, 1) \in \partial\Omega \\ 0 & \text{otherwise} \end{cases} \quad \dots\dots\dots (7)$$

$l=m=n=40$ として、離散化した方程式を解いた。収束条件は、最大値ノルム $\|\cdot\|$ を用いて $\|A\mathbf{u}^n - \mathbf{z}\|/\|\mathbf{z}\| < 10^{-5}$ である。初期ベクトルは $\mathbf{u}^0=0$ とした。最適加速係数 ω_{opt} は、分割数 $n=40$ に対して次の式で与えた⁽¹⁾。

$$\omega_{opt} = \frac{2}{1 + \sin(\pi/n)} \quad \dots\dots\dots (8)$$

また、red-black, 4-color SOR法のプログラムでは、Fortranコンパイラに対しベクトル計算可能であることを示すために、ベクトル化指示行を挿入した。

まず、収束に必要な反復回数は、natural, red-black, 4-color SOR法で、それぞれ119, 77, 77であった。これは、初期残差ベクトルに対し、各SOR反復行列の同じ固有値に対する固有ベクトル成分が異なるためである⁽⁵⁾。

次に、各SOR法の実行時間を表1に示す。この実行時間の中には、収束判定のための残差計算の時間も含まれている。表においてScalarはスカラ・モード、Vectorはベクトル・モード、PV2, PV3, PV4はそれぞれ2, 3, 4台の並列ベクトル・モードでの実行を示している。

スカラ・モードでは、各SOR法の1回の反復に必要な計算時間がほぼ等しい。これは、1回の反復に必要な演算量がどのSOR法においても等しいからである。ベクトルまたは並列ベクトル・モードでのnatural SOR法は、並列処理部分が少ないため実行時間が短縮されず、他の方法と比較してかなり遅いことがわかる。また、4-color SOR法はこれらの中で最も速いことが明らかになった。

各方法の並列処理による速度向上率を図4に示す。図中maximumで示された直線は、 n 台のプロセッサで n 倍の速度向上が得られる場合の最高速度向上率である。4-color SOR法は4台のプロセッサで約3.2倍の速度向上が得られた。これは、他のSOR法と比較して最も高い。図5に4-color SOR法において、スカラ・モードの実行時間を1とした時のベクトルおよび

表1 各SOR法の実行時間

(sec.)

	Scalar	Vector	PV2	PV3	PV4
Natural	271.4	203.6	305.3	296.4	293.0
Red-black	204.0	74.1	44.8	33.2	28.1
4-color	218.2	71.9	38.0	27.6	22.3

図4 並列処理による速度向上率

並列ベクトル・モードの実行時間比を示す。ベクトル処理だけで約3.0倍の速度向上が見られる。また、並列ベクトル処理によって、ベクトル処理と並列処理との速度向上率の積である約9.8倍の速度向上が得られていることがわかる。これは、4-color SOR法の二重ループが外側ループを並列、内側ループをベクトル処理できるためである。

5. ま と め

オーダリングの異なるSOR法をマルチプロセッサシステム上で比較して、実行時間や並列処理効率の点で4-color SOR法が有効であることを示した。連立一次方程式の数値解法はいろいろあるが、SOR法はプログラミングが容易であり、並列計算によって高速化できるので、プログラム開発の最初のスキームとして最適である。

図5 4-color SOR法の速度向上率

また、多重ループにデータ依存関係がない場合には、並列ベクトル計算によって、ベクトル処理と並列処理による速度向上の積である速度向上が得られることを、4-color SOR法を用いて示した。このような多重ループは、数値シミュレーションのプログラムにおいて良く見かけられるので、並列ベクトル計算によってプログラムをより高速化できる。

Alliant FX/8上でプログラムを実行していただいた(株)日本アポロコンピュータ福田幾夫氏に感謝する。

文 献

- (1) Varga, R. S., *Matrix Iterative Analysis*, (1962), 88-117, Prentice-Hall.
- (2) Meier, U., *Parallel Computing*, 3(1986), 205-215.
- (3) Evertsz, C., ほか1名, *Supercomputer*, (1986-3), 35-43.
- (4) Adams, L. M., ほか1名, *SIAM J. Sci. Stat. Comput.*, 7(1986), 490-506.
- (5) Yokokawa, M., *JAERI-M* 88-017, (1988).
- (6) Asai, K., ほか12名, *JAERI-M* 87-136, (1987).