最新プロセッサの SIMD 演算を活用する 並列化ガウス＝ザイデルスムーザの実装方法

河 合 直 聡† 岩 下 武 史†† 中 島 浩††

1. はじめに

著者らはこれまでに偏微分方程式の有限差分解析を対象とした幾何マルチグリッド法[1) について研究を行ってきた．文献[2) では，マルチコアプロセッサ上の 3 次元ポアソン方程式を対象とした解析において，ブロック化赤-黒順序付け[3) に基づく並列化ガウス＝ザイデルスムーザ（BRBGS）を導入し，従来広く用いられていたストライドメモリアクセスを用いた赤-黒順序付けガウス＝ザイデルスムーザ（RBGS）と比べて良好な性能が得られることを述べた．ここで，BRBGS の優位性は主にブロック内の処理が逐次ガウス＝ザイデルスムーザと同一で，RBGS と比べてメモリ上で連続となるデータアクセスが多いことによる．

一方，近年の多くの汎用プロセッサでは，SIMD（Single Instruction Multiple Data）演算命令が利用可能であり，1 命令で同時に処理できるデータ量（SIMD 幅）も拡大している傾向にある．また，現在 HPC 分野で注目されている MIC（Many Integrated Core）アーキテクチャに基づく Intel Xeon Phi は，汎用マルチコアプロセッサと比べてより広い SIMD 幅を有している．しかしながら，BRBGS では単純な実装法を用いた場合，ブロック内の逐次ガウス＝ザイデル処理において SIMD 演算を活用することができない．そこで，本稿では逐次ガウス＝ザイデル法の SIMD 演算を考慮した実装法を提案し，BRBGS のブロック内計算に導入する．本並列化スムーザを従来の BRBGS や SIMD 演算を用いた RBGS と比較し，その性能を評価する．

2. ガウス＝ザイデルスムーザの SIMD 化

本稿で対象とする問題は，3 次元ポアソン方程式を 7 点差分公式による有限差分法を用いて離散化した結果生ずる連立一次方程式であり，これを幾何マルチグリッド法により解くことを考える．ただし，本稿ではスムージング処理のみを議論の対処とし，性能評価には最密グリッド上の 1 回のスムージング処理に要する計算時間を用いる．

```
do k = 1, NZ
  do j = 1, NY
    do i = 1, NX
      phi( i, j, k) = -rho( i, j, k)
 + zp × phi( i, j, k+1) + zm × phi( i, j, k-1)
 + yp × phi( i, j+1, k) + ym × phi( i, j-1, k)
 + xp × phi( i+1, j, k) + xm × phi( i-1, j, k)
    enddo
  enddo
enddo
```

図 1 逐次ガウス＝ザイデルスムーザのサンプルコード

図 1 は辞書式順序付けに基づいた逐次ガウス＝ザイデルスムーザのサンプルコードである．ここで，NX, NY，NZ は各座標方向の対象領域サイズであり，rho, phi は（倍精度）実数型の配列で，xp，xm，yp，ym，zp，zm は（倍精度）実数型の変数である．ここで，図 1 の最内ループは，配列要素 phi(i-1,j,k) と phi(i,j,k) の間にデータ依存関係があり，このままではベクトル化（SIMD 化）できない．しかし，当該の phi(i,j,k) の更新式において，他の配列要素と phi(i,j,k) はデータ依存関係を持たない．そこで，図 1 の最内ループをステンシル毎のループに分割した図 2 のような実装法を提案する．この場合，最内のループの内，最後のループは SIMD 化できないが，その他のループは SIMD 化が可能となる．本実装法は BRBGS のブロック内処理に活用することができる．

3. 数値実験

提案手法を用いて SIMD 化した BRBGS（以降では BRBGS(SIMD) と表記）と従来の BRBGS（BRBGS(Org) と表記）及びストライドメモリアクセスを用いた RBGS の比較を数値実験により行う．数

† 京都大学大学院 情報学研究科
†† 京都大学学術情報メディアセンター

```
do k = 1, NZ
  do j = 1, NY
    !$DEC SIMD
    do i = 1, NX
  tmp( i ) = -rho( i ,j ,k) + zm × phi( i ,j ,k-1)
    enddo
    !$DEC SIMD
    do i = 1, NX
     tmp( i ) = tmp( i ) + ym × phi( i ,j-1 ,k)
    enddo
        ⋮
    do i = 1, NX
    phi( i ,j ,k) = tmp(i) + xm × phi( i-1 ,j ,k)
    enddo
  enddo
enddo
```

図 2 提案する実装手法でのガウス=ザイデルスムーザのサンプルコード

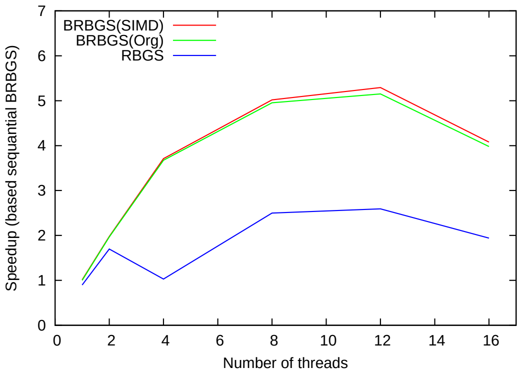

図 3 GreenBlade 上における各手法の速度向上率（BRBGS の逐次実行を基準）

値実験で用いる差分格子のサイズ（格子点数）は 512×180×960（x, y, z 方向の順）である．実験に使用する計算機は京都大学学術情報メディアセンターの Appro GreenBlade8000（Intel XeonE2670 搭載）の 1 ノードおよび Xeon Phi を有する Intel SE10 ボードである．図 3 及び図 4 に，BRBGS(Org) の 1 スレッド実行時の計算時間に対する各スムーザの速度向上率を示す．

実験の結果，GreenBlade において，提案する BRBGS(SIMD) は BRBGS(Org) と比べて逐次実行で 1.1%，16 スレッドで 4.8%の性能向上を得た．また，同様の比較において SE10 上では 4 スレッド（1 コア）で 9.4%，240 スレッド（60 コア）で 4.3%の性能向上を得た．また，図 2 による BRBGS(SIMD) の実装プログラムをプロファイラにより調査した結果，各ループ内における 3 次元配列に関するアドレス計算の影響が大きいことが分かった．従って，3 次元配列を 1 次元化することによりさらに性能改善を行う事ができる可能性が示唆される．

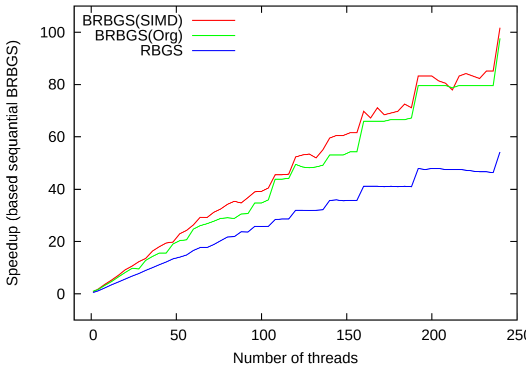

図 4 SE10 上における各手法の速度向上率（BRBGS の逐次実行を基準）

また，実験で用いた二つの計算機に関する比較では，BRBGS(SIMD) の GreenBlade，16 スレッドでの計算時間は 2.83 秒，SE10，240 スレッドでは 0.72 秒と SE10 での結果が 3.9 倍速かった．一方，RBGS と BRBGS との比較では，どちらの環境でも BRBGS の方が速かった．ストライドアクセスを用いた RBGS は SIMD 演算を活用することが可能であるが，文献[2] の結果と同様にキャッシュデータの再利用性が低いことが性能に影響したと考えられる．その結果，BRBGS(SIMD) は RBGS に対して BreenBlade 上で約 2.1 倍，SE10 上で約 1.9 倍高速となった．

参 考 文 献

1) W.L. Briggs, V.E. Henson, and S.F. McCormick. *A Multigrid Tutorial Second Edition.* SIAM, Philadelphia , PA, 2000.
2) 河合直聡, 岩下武史, 中島浩. ブロック化赤−黒順序付け法に基づく並列マルチグリッドポアソンソルバ. 情報処理学会論文誌コンピューティングシステム（ACS）, Vol. 5, No. 3, pp. 1–10, May 2012.
3) T. Iwashita and M. Shimasaki. Block red-black ordering: a new ordering strategy for parallelization of ICCG method. *Int. J. Parallel Prog.*, Vol. 31, pp. 55–75, 2003.