2G-05

Matrix Powers Kernel の反復解法への適用

野地優希† 熊谷洋佑† 藤井昭宏† 田中輝雄†

工学院大学†

1 はじめに

連立一次方程式を解く反復解法として共役勾配法(CG 法)が広く用いられている．CG 法では内積計算の通信コストが高並列になるにつれて大きくなる．そこで，Chebyshev 多項式を基底とすることで内積計算の回数を削減する Chebyshev 基底共役勾配法(CBCG 法)が提案されている[1]. 熊谷らにより京上での CBCG 法の性能分析がされている[2]. CBCG 法は通信削減をすると同時に新たに行列計算が必要となる．そこで，近年$\{\boldsymbol{r}, A\boldsymbol{r}, A^2\boldsymbol{r}, \dots, A^k\boldsymbol{r}\}$といった行列のべき乗演算時間を削減することにより，反復解法を高速化する手法として Matrix Powers Kernel(MPK)がある[3]. MPK は計算を指数方向も含めてブロック化を行うことで，キャッシュを効率的に再利用する手法である．ブロックの重複により計算量が増えるが，メモリアクセスが改善される．MPK の性能面に関しては，Dehnavi らにより GPU 上において，黒田らにより共有メモリ環境において高速になることが報告されている[4] [5].

本研究では，反復解法である CBCG 法に対して MPK を適用し，共有メモリ環境での演算時間の削減を目的とする．

2 Matrix Powers Kernel

2.1 重複有り計算

MPK は$\{A\boldsymbol{r}, A^2\boldsymbol{r}, \dots, A^k\boldsymbol{r}\}$の計算で$A^k\boldsymbol{r}$を分割し，その領域計算に必要な要素を一度に計算する．それにより，キャッシュヒット率を向上させる手法である．一般的な MPK では，計算領域ごとに並列に実行可能であるが，隣接した計算領域において重複計算が発生する．以下では，この手法を重複有り計算とする．

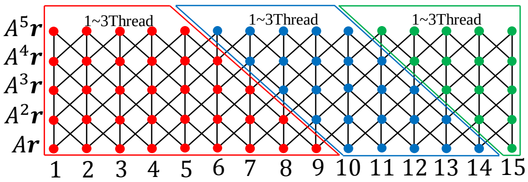

図 1　三重対角行列での重複無し MPK

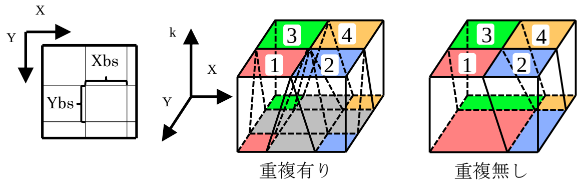

図 2　2 次元差分での重複有り/無し MPK

2.2 重複無し計算

要素数 15, k=5 の三重対角行列での MPK の計算を図 1 に示す．行は$A^i\boldsymbol{r}$の要素である．$A^i\boldsymbol{r}$の j 番目の要素を更新する際は，$A^{i-1}\boldsymbol{r}$の j-1，j，j+1 番目の要素が計算されている必要がある．MPK では計算順序を固定することにより重複計算を削減できることがわかっている．赤点で既に計算された要素を青の領域を計算する際に用いることができ，それにより，重複計算を削減することができる．今回は，計算領域内の並列化を行った．

2 次元 5 点差分では，$A^k\boldsymbol{r}$に相当する領域を$Xbs \times Ybs$のブロックに分割した．ブロックの計算領域は図 2 のようになる．通常，隣接した分割領域において重複計算が発生するが，既に計算された要素を別の分割領域で用いることで，重複計算を削減することができる．

2.3 重複有り/無し計算における計算量の比較

2 次元 5 点差分問題における領域サイズ$(X \times Y)$の非零要素数は，$5 \times X \times Y - 2(X + Y)$となる．MPK は図 2 で示した通り，$A^k\boldsymbol{r}$に相当する領域を$Xbs \times Ybs$のブロックに分割して計算を行った．その際の MPK 無しと MPK 有り(重複有り/無し)

表 1　2 次元 5 点差分における計算量

	計算量
MPK 無し	$\left(10X \times Y - 4(X + Y)\right) \times k$
MPK1	$\frac{k}{3}\{4k^2 + (3k + 2)(Xbs + Ybs) + Xbs \times Ybs\}$
MPK2	$\left(10X \times Y - 4(X + Y)\right) \times k$

Application to Iterative Solver Method using Matrix Powers Kernel
Yuki Noji†, Yosuke Kumagai†, Akihiro Fujii† and Teruo Tanaka†
†Kogakuin University

Algorithm The Chebyshev basis

$\eta := 2/(\lambda_{max} - \lambda_{min})$
$\zeta := (\lambda_{max} + \lambda_{min})/(\lambda_{max} - \lambda_{min})$
$\boldsymbol{s}_0 := \boldsymbol{r}$
$\underline{\boldsymbol{s}_1 := \eta A \boldsymbol{s}_0 - \zeta \boldsymbol{s}_0}$
for $j = 2$ to k **do**
  $\underline{\boldsymbol{s}_j := 2\eta A \boldsymbol{s}_{j-1} - 2\eta \boldsymbol{s}_{j-1} - \boldsymbol{s}_{j-2}}$
end for
$S_i := (\boldsymbol{s}_0, \boldsymbol{s}_1, \cdots, \boldsymbol{s}_{k-1})$
$\underline{AS_i := (A\boldsymbol{s}_0, A\boldsymbol{s}_1, \cdots, A\boldsymbol{s}_{k-1})}$

図 3 Chebyshev 基底生成部における$\boldsymbol{s}_0 \sim \boldsymbol{s}_k$のベクトルを求めるアルゴリズム

における浮動小数点の乗算と加算を含めた計算量の比較を表 1 に示す．以下，それぞれを MPK 無し，MPK1，MPK2 とする．

2.4 反復解法への適用

CBCG 法では Chebyshev 基底生成部において，ベクトル演算を含む行列べき乗演算が存在する．アルゴリズムを図 3 に示し，下線部が MPK を適用した部分となる．アルゴリズム中に現れるλ_{min}，λ_{max}はそれぞれ係数行列の最小・最大固有値である．

3 実験

3.1 実験環境と実験内容

実験には Intel Core i7-3770K(3.5GHz, 4Core, Memory : 32GB, L3cache : 8MB)を使用した．コンパイラは icc14.0.1，オプションは“-O3 -openmp -lm”を使用した．各計算領域内の計算処理を OpenMP によりスレッド並列化し，スレッド数は 8 とした. 実験では 2 次元 5 点差分により Poisson 方程式を離散化した問題を使用した．また，問題領域サイズを1000×1000と固定した．

実験は 3 種類の CBCG 法($k = 10$, 20, 30)を用い，MPK 無し，MPK1，MPK2 で計測する．反復の終了条件は相対残差が10^{-12}とした．

3.2 実験結果

$A^{k}\boldsymbol{r}$におけるブロックサイズを Xbs，Ybs 共に$100 \sim 1000$の間で変化をさせた時の MPK 無しと最適なブロックサイズにおける MPK1，MPK2 による実行時間の内訳を図 4 に示す. 図 4 では各べき乗数において，それぞれ 14.2%，8.9%，0.2%の実行時間を削減することができた. 図 4 から今回 MPK を適用した部分の実行時間の内訳を図 5 に示す. 図 5 における SpMV の時間を比較すると，各べき乗数において MPK2 は MPK 無しに対して約 34%時間を短縮することができ，ベクトル演算においても, 約半分の時間に削減することができた．これらは，キャッシュに収まるようにキャッシュブロッキングを行うことで, SpMV やベクトル演算における効果が高いことがわかった．

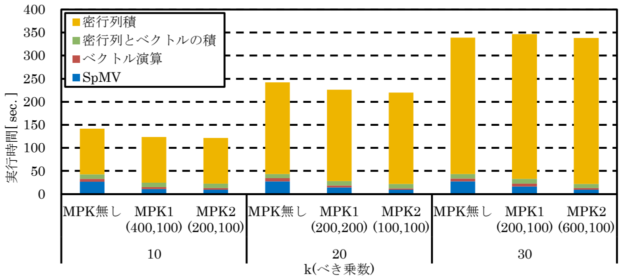

図 4 CBCG 法における実行時間の内訳

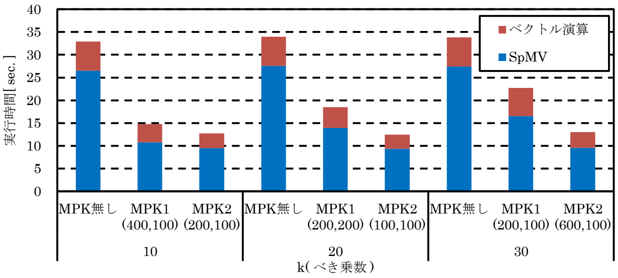

図 5 MPK を適用した部分の実行時間の内訳

4 おわりに

本研究では CBCG 法において，ベクトル演算を含む行列べき乗演算が存在する Chebyshev 基底生成部に対して MPK を適用し，共有メモリ環境での高速化を図った. MPK2 で計算を行うことで, SpMV とベクトル演算における演算時間を削減することができた．今後の課題としては，密行列積部分の改善やその他の反復解法へ適用した際の効果，分散環境における MPK の適用をした際の効果について検証をする必要がある．

謝辞 本研究の一部は JSPS 科学研究費 25330144 の助成を受けて行われた．

参考文献

[1] 須田礼仁，李聡，島根浩平，数値的に安定性な通信削減クリロフ部分空間法，計算工学講演会論文集, Vol.19 (2014).

[2] Kumagai, Y., et al.: Performance Analysis of the Chebyshev Basis Conjugate Gradient Method on the K Computer, 11th International Conference on Parallel Processing and Applied Mathematics (2015).

[3] Demmel, J., et al.: Avoiding Communication in Sparse Matrix Communications, in IEEE International Parallel and Distributed Processing Symposium (2008).

[4] Dehnavi, M.M., et al.: Communication-aviding Krylov Techniques on Graphics Processing Units, Magnetics, IEEE Transactions on, Vol.49, pp.1749-1752 (2013).

[5] 黒田勝汰，藤井昭宏，田中輝雄, Matrix Powers Kernel の共有メモリ環境への適用における Multicolor ordering による重複計算の軽減，ハイパフォーマンスコンピューティング研究発表会, Vol.2015-HPC-148, No.6, pp.1-6 (2015).