論文

同期点を減らしたGPBiCG 法の並列性能評価†

藤野　清次*・岩里　洸介**

A Parallel Performance Estimation of GPBiCG Method With Reduction of Synchronization Points

Seiji Fujino* and Kosuke Iwasato**

Abstract GPBiCG（Generalized Product of BiCG）method is often used for the solution of realistic problems on parallel computers. However, the number of synchronizations points for inner products needs three times per iteration, and these synchronization points cause great increase of communication time. Then, we devised a strategy to reduce the number of synchronization points. We applied this strategy to the original BiCGSafe method and succeeded in fair reduction of their communication time. In this paper, we apply this strategy to the original GPBiCG method. Through many numerical experiments, we will make clear that the proposed parallel version of GPBiCG method with double synchronization points outperforms the conventional methods in view of parallel computation.

Key words GPBiCG Method, Synchronization Points, Inner Product, Parallel Computation

1. はじめに

大規模非対称疎行列を係数とする線形方程式 $A\boldsymbol{x} = \boldsymbol{b}$ を解くことを考える．ここで，係数行列 A は $n \times n$ の実非対称な疎行列，$\boldsymbol{x}$，$\boldsymbol{b}$ は n 次の解ベクトル，及び右辺ベクトルとする．この線形方程式の数値解法として，反復法，特に積型反復法がよく用いられる．積型反復法は残差ベクトル $\boldsymbol{r}(:= \boldsymbol{b} - A\boldsymbol{x})$ が安定化多項式と Lanczos 多項式と初期残差ベクトル $\boldsymbol{r}_0$ の積の形で表される反復法である積型反復法の一つに GPBiCG 法[8]がある．GPBiCG 法は，加速多項式に 2 種類のパラメータ ζ_k，η_k を含む 3 項漸化式を使用し，これらのパラメータを残差ベクトル $\boldsymbol{r}_{k+1}$ の 2 ノルムの局所的な最小化に基づいて決定する反復法である．また，GP-BiCG 法を高速化するためには算法自体の改良と並列化などが重要である[9]．

分散メモリ型並列計算機における並列化手法の一つに，MPI（Message Passing Interface）ライブラリを用いたプロセス並列（Flat MPI 並列と呼ばれる）化がある．Flat MPI 並列化は大規模な並列化が可能であるが，各プロセスが独自のメモリ空間を持つため，あるプロセスが他のプロセスのデータを参照する場合にプロセス間通信が必要となる．大規模な並列化を行う場合，プロセス間通信にかかる時間の割合が増加するため，並列性能が大きく低下する．そのため，分散メモリ型の計算機上で Flat MPI 並列化により並列計算を行う場合，行列ベクトル積と内積計算で発生する通信が大きな問題となる場合がある．Krylov 部分空間法において，内積計算は並列に計算する際にプロセス全体の集団通信が必要となり同期点と呼ばれる．本研究で取り上げる GPBiCG 法には同期が 3 ケ所点在し，並列数が多い場合に高速化の妨げになる場合がある．

そこで，積型反復法の 1 反復中の同期回数を削減する手法が提案された．その手法はランチョス多項式とその補助多項式の更新に用いられるパラメータ β_k を IDR 定理に基づく BiCG 法で用いられた計算式で更新

* 九州大学情報基盤研究開発センター
Research Institute for Information Technology, Kyushu University
** 九州大学大学院システム情報科学府
Graduate School of Information Science and Electrical Engineering, Kyushu University
† 2015 年 9 月 24 日受付

することにより1反復中の同期点を削減するものである．この同期点を削減する手法はBiCGStar-plus法[5]やBiCGSafe法[2,3]に適用され，並列計算において収束の高速化が報告された．ただし，これらの解法はβ_kの更新に転置行列ベクトル積が必要なため，ハイブリッド並列計算において通信量が増加する要因になることが報告されている[4]．ただし，本研究では，Flat並列の場合を扱い，前の発表[4]とは並列環境に違いがある．

本論文ではGPBiCG法のパラメータβ_kの計算式を見直し1反復中の同期点数を減らしたGPBiCG法開発に導び付けることにする．さらに，同期削減版GPBiCG法はBiCGStar-plus法やBiCGSafe法と異なり，転置行列ベクトル積を行わない算法の構築が可能であることも示す．数値実験を通して，同期削減版GPBiCG法の並列性能評価を行い，提案の反復法が従来の反復法と比べて並列性能が優れていることを明らかにする．

本論文の構成は以下の通りである．第2節でGPBiCG法の概略を示す．第3節で同期削減版GPBiCG法を導出する．パラメータβ_kをIDR定理に基づくBiCG法で用いられた計算式に変形することで反復中の同期点を削減する．また，本解法では従来の同期削減版の解法と異なり，転置行列ベクトル積を行わずに算法を構築することが出来ることを示す．さらに，第4節で数値実験を通して同期削減版GPBiCG法の並列計算における性能評価を行う．最後に，第5節でまとめを行う．

2. 元のGPBiCG法

本節では，元のGPBiCG法について述べる[8]．残差多項式$R_k(\lambda)$と補助多項式$P_k(\lambda)$は次の交代漸化式

$$R_0(\lambda) = 1, \quad P_0(\lambda) = 1, \tag{1}$$
$$R_k(\lambda) = R_{k-1}(\lambda) - \alpha_{k-1}\lambda P_{k-1}(\lambda), \tag{2}$$
$$P_k(\lambda) = R_k(\lambda) + \beta_{k-1}P_{k-1}(\lambda),\ k = 1, 2, \ldots. \tag{3}$$

を満たす．ここでλは行列Aの固有値である．加速多項式は適当な2種類のパラメータζ_kとη_kを導入し，以下の交代漸化式で表される．

$$H_0(\lambda) = 1,\ G_0(\lambda) = \zeta_0, \tag{4}$$
$$H_k(\lambda) = H_{k-1}(\lambda) - \lambda G_{k-1}(\lambda), \tag{5}$$
$$G_k(\lambda) = \zeta_k H_k(\lambda) + \eta_k G_{k-1}(\lambda).\ k = 1, 2, \ldots. \tag{6}$$

ここで，多項式$G_k(\lambda)$は

$$G_k(\lambda) = -(H_{k+1}(\lambda) - H_k(\lambda))/\lambda \tag{7}$$

を満足する．漸化式中のパラメータζ_k, η_kの値は，残差ベクトル$\boldsymbol{r}_{k+1}$の2ノルムが局所的最小で決定する．元のGPBiCG法の算法を以下に示す．GPBiCG法で発生する同期点は，α_k更新時，β_k更新時，ζ_k，η_k更新時の計3回である．ϵは収束判定用の微小値である．

Algorithm 1：元の（同期3回）GPBiCG法の算法

Let $\boldsymbol{x}_0$ be an initial guess, $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$,
Choose $\boldsymbol{r}_0^*$ such that $(\boldsymbol{r}_0^*, \boldsymbol{r}_0) \neq 0$,
Set $\beta_{-1} = 0$, $\boldsymbol{p}_{-1} = \boldsymbol{u}_{-1} = \boldsymbol{t}_{-1} = \boldsymbol{w}_{-1} = \boldsymbol{0}$,
for $k = 0, 1, \cdots$ do,
  $\boldsymbol{p}_k = \boldsymbol{r}_k + \beta_{k-1}(\boldsymbol{p}_{k-1} - \boldsymbol{u}_{k-1})$,
  Compute $A\boldsymbol{p}_k$,
  $\alpha_k = \dfrac{(\boldsymbol{r}_0^*, \boldsymbol{r}_k)}{(\boldsymbol{r}_0^*, A\boldsymbol{p}_k)}$,
  $\boldsymbol{y}_k = \boldsymbol{t}_{k-1} - \boldsymbol{r}_k - \alpha_k\boldsymbol{w}_{k-1} + \alpha_k A\boldsymbol{p}_k$,
  $\boldsymbol{t}_k = \boldsymbol{r}_k - \alpha_k A\boldsymbol{p}_k$,
  Compute $A\boldsymbol{t}_k (= \boldsymbol{q}_k)$,
  $\zeta_k = \dfrac{(\boldsymbol{y}_k, \boldsymbol{y}_k)(\boldsymbol{q}_k, \boldsymbol{t}_k) - (\boldsymbol{y}_k, \boldsymbol{t}_k)(\boldsymbol{q}_k, \boldsymbol{y}_k)}{(\boldsymbol{q}_k, \boldsymbol{q}_k)(\boldsymbol{y}_k, \boldsymbol{y}_k) - (\boldsymbol{y}_k, \boldsymbol{q}_k)(\boldsymbol{q}_k, \boldsymbol{y}_k)}$,
  $\eta_k = \dfrac{(\boldsymbol{q}_k, \boldsymbol{q}_k)(\boldsymbol{y}_k, \boldsymbol{t}_k) - (\boldsymbol{y}_k, \boldsymbol{q}_k)(\boldsymbol{q}_k, \boldsymbol{t}_k)}{(\boldsymbol{q}_k, \boldsymbol{q}_k)(\boldsymbol{y}_k, \boldsymbol{y}_k) - (\boldsymbol{y}_k, \boldsymbol{q}_k)(\boldsymbol{q}_k, \boldsymbol{y}_k)}$,
  (if $k = 0$, then $\zeta_k = \dfrac{(A\boldsymbol{t}_k, \boldsymbol{t}_k)}{(A\boldsymbol{t}_k, A\boldsymbol{t}_k)}$, $\eta_k = 0$),
  $\boldsymbol{u}_k = \zeta_k A\boldsymbol{p}_k + \eta_k(\boldsymbol{t}_{k-1} - \boldsymbol{r}_k + \beta_{k-1}\boldsymbol{u}_{k-1})$,
  $\boldsymbol{z}_k = \zeta_k\boldsymbol{r}_k + \eta_k\boldsymbol{z}_{k-1} - \alpha_k\boldsymbol{u}_k$,
  $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k\boldsymbol{p}_k + \boldsymbol{z}_k$,
  $\boldsymbol{r}_{k+1} = \boldsymbol{t}_k - \eta_k\boldsymbol{y}_k - \zeta_k A\boldsymbol{t}_k$,
  if $\|\boldsymbol{r}_{k+1}\|/\|\boldsymbol{r}_0\| \leq \varepsilon$ stop,
  $\beta_k = \dfrac{\alpha_k}{\zeta_k} \cdot \dfrac{(\boldsymbol{r}_0^*, \boldsymbol{r}_{k+1})}{(\boldsymbol{r}_0^*, \boldsymbol{r}_k)}$,
  $\boldsymbol{w}_k = A\boldsymbol{t}_k + \beta_k A\boldsymbol{p}_k$,
end do.

3. 同期削減版GPBiCG法

3.1 同期点の削減における准残差に基づく解法とGPBiCG法の違い

積型反復法において，反復中の同期点を削減する方法としてLanczos多項式のパラメータβ_kを変形する手法がある．このとき，β_kの計算は，IDR(s)法に基づくBiCG法で採用された方法に基づく[1,6]．すなわち，β_kの更新式は次式で与えられる．

$$\beta_k = -(\boldsymbol{r}_0^*,\ AH_kR_{k+1}\boldsymbol{r}_0)/(\boldsymbol{r}_0^*,\ AH_kP_k\boldsymbol{r}_0). \tag{8}$$

BiCGSafe法やBiCGStar-plus法などの准残差に基づく積型反復法では，$AH_kR_{k+1}\boldsymbol{r}_0$は算法中で更新されないため，式(8)のままでは，1反復当りの行列ベクトル

積の回数が 1 回多くなり，演算量が多くなる．したがって，式(8)の分子を

$$(\boldsymbol{r}_0^*, AH_kR_{k+1}) = (A^{\mathrm{T}}\boldsymbol{r}_0^*, H_kR_{k+1}) \tag{9}$$

と変形して，算法を構築する．そのため，准残差に基づく積型反復法では反復部分の演算を行う前に転置行列ベクトル積を行う必要があった．

しかし，GPBiCG 法では $AH_kR_{k+1}\boldsymbol{r}_0$ は Algorithm 1 の $A\boldsymbol{t}_k$ に対応しており，既に算法中で使用されているため，新たに行列ベクトル積を行う必要がない．そのため，GPBiCG 法は式(9)の変形を行わずに，式(8)を β_k の更新式に使用することが出来る．したがって，β_k の変形による同期点削減を GPBiCG 法に適用することにより，同期点を削減し，かつ転置行列ベクトル積を行わない積型反復法が導出できる．

3.2 今回の同期削減版 GPBiCG 法

ここでは，転置行列ベクトル積を行わない同期削減版 GPBiCG 法を導出する．β_k は式(8)により更新する．ここで，$AH_kP_k\boldsymbol{r}_0$ は Algorithm 1 の $A\boldsymbol{p}_k$ に対応するので，式(8)は次のように表される．

$$\beta_k = -(\boldsymbol{r}_0^*,\ A\boldsymbol{t}_k)/(\boldsymbol{r}_0^*,\ A\boldsymbol{p}_k). \tag{10}$$

これにより，β_k，ζ_k，η_k を同時に更新できるようになり，同期点を 1 反復当り 2 回に削減することができる．

以上の議論により，β_k の変形と，ベクトルの計算順序を入れ替えによる同期削減版 GPBiCG 法が導出された．今回の同期削減版 GPBiCG のアルゴリズムを以下記す．ここでは，パラメータ β_k の正負の符号を入れ替え記述した．また，補助ベクトル $\boldsymbol{s}_k := \boldsymbol{t}_{k-1} - \boldsymbol{r}_k + \beta_{k-1}\boldsymbol{u}_{k-1}$ も用いた．太字表記をした 12 ラインから 14 ラインが該当部分を指す．

Algorithm 2：同期削減版（同期 2 回） GPBiCG 法

1. Let $\boldsymbol{x}_0$ be an initial guess, $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$,
2. Choose $\boldsymbol{r}_0^*$ such that $(\boldsymbol{r}_0^*, \boldsymbol{r}_0) \neq 0$,
3. Set $\boldsymbol{p}_0 = \boldsymbol{r}_0$, $\boldsymbol{u}_{-1} = \boldsymbol{t}_{-1} = \boldsymbol{w}_0 = \boldsymbol{0}$, $\beta_{-1} = 0$
4. for $k = 0, 1, \cdots$ do,
5. 　Compute $A\boldsymbol{p}_k$,
6. 　if $\|\boldsymbol{r}_{k+1}\|/\|\boldsymbol{r}_0\| \leq \varepsilon$ stop,
7. 　$\alpha_k = \dfrac{(\boldsymbol{r}_0^*, \boldsymbol{r}_k)}{(\boldsymbol{r}_0^*, A\boldsymbol{p}_k)}$,
8. 　$\boldsymbol{s}_k = \boldsymbol{t}_{k-1} - \boldsymbol{r}_k + \beta_{k-1}\boldsymbol{u}_{k-1}$,
9. 　$\boldsymbol{y}_k = \boldsymbol{t}_{k-1} - \boldsymbol{r}_k - \alpha_k\boldsymbol{w}_k + \alpha_k A\boldsymbol{p}_k$,
10. 　$\boldsymbol{t}_k = \boldsymbol{r}_k - \alpha_k A\boldsymbol{p}_k$,
11. 　Compute $A\boldsymbol{t}_k (= \boldsymbol{q}_k)$,
12. 　$\beta_k = \dfrac{(\boldsymbol{r}_0^*, A\boldsymbol{t}_k)}{(\boldsymbol{r}_0^*, A\boldsymbol{p}_k)}$,
13. 　$\zeta_k = \dfrac{(\boldsymbol{y}_k, \boldsymbol{y}_k)(\boldsymbol{q}_k, \boldsymbol{t}_k) - (\boldsymbol{y}_k, \boldsymbol{t}_k)(\boldsymbol{q}_k, \boldsymbol{y}_k)}{(\boldsymbol{q}_k, \boldsymbol{q}_k)(\boldsymbol{y}_k, \boldsymbol{y}_k) - (\boldsymbol{y}_k, \boldsymbol{q}_k)(\boldsymbol{q}_k, \boldsymbol{y}_k)}$,
14. 　$\eta_k = \dfrac{(\boldsymbol{q}_k, \boldsymbol{q}_k)(\boldsymbol{y}_k, \boldsymbol{t}_k) - (\boldsymbol{y}_k, \boldsymbol{q}_k)(\boldsymbol{q}_k, \boldsymbol{t}_k)}{(\boldsymbol{q}_k, \boldsymbol{q}_k)(\boldsymbol{y}_k, \boldsymbol{y}_k) - (\boldsymbol{y}_k, \boldsymbol{q}_k)(\boldsymbol{q}_k, \boldsymbol{y}_k)}$,
15. 　(if $k = 0$, then $\zeta_k = \dfrac{(A\boldsymbol{t}_k, \boldsymbol{t}_k)}{(A\boldsymbol{t}_k, A\boldsymbol{t}_k)}$, $\eta_k = 0$),
16. 　$\boldsymbol{u}_k = \zeta_k A\boldsymbol{p}_k + \eta_k\boldsymbol{s}_k$,
17. 　$\boldsymbol{z}_k = \zeta_k\boldsymbol{r}_k + \eta_k\boldsymbol{z}_{k-1} - \alpha_k\boldsymbol{u}_k$,
18. 　$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k\boldsymbol{p}_k + \boldsymbol{z}_k$,
19. 　$\boldsymbol{r}_{k+1} = \boldsymbol{t}_k - \eta_k\boldsymbol{y}_k - \zeta_k A\boldsymbol{t}_k$,
20. 　$\boldsymbol{w}_{k+1} = A\boldsymbol{t}_k + \beta_k A\boldsymbol{p}_k$,
21. 　$\boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} + \beta_k(\boldsymbol{p}_k - \boldsymbol{u}_k)$,
22. end do.

4. 数値実験

4.1 計算機環境と計算条件

計算機環境は以下の通りである．計算機は CX400（CPU：Intel Xeon E5-2690, クロック周波数：2.7 GHz, メモリ：128 Gbytes, キャッシュメモリ：20 MB，OS：Red Hat Linux Enterprise）を使用した．コンパイルオプションは“-O3”を使用した．プログラムは Fortran90 により実装し，コンパイラは Fujitsu Technical Computing Sute ver.2.0 を用いた．計算は倍精度浮動小数点演算，経過時間の計測は関数 gettimeofday を用いた．

計算条件は以下の通りである．収束判定値は相対残差の 2 ノルム：$\|\boldsymbol{r}_{k+1}\|_2/\|\boldsymbol{r}_0\|_2 \leq 10^{-8}$ とした．初期近似解 $\boldsymbol{x}_0$ はすべて $\boldsymbol{0}$，最大反復回数は 10000 回とした．行列は予め対角スケーリングによって対角項を 1 に正規化した．右辺項は物理的条件から得られる値を用いた．存在しない場合には，厳密解を $\hat{\boldsymbol{x}} = (1, 1, \ldots, 1)^{\mathrm{T}}$ とし，$\boldsymbol{b} = A\hat{\boldsymbol{x}}$ で作成した．初期シャドウ残差ベクトル $\tilde{\boldsymbol{r}}_0$ は一様乱数を使用した．並列化は Flat 並列化を行い，8，16，32，64，128，256 プロセスで実験した．実験は 5 回行い，最大(小)値を除く 3 回の平均を実験結果とした．次に，**表 1** にテスト行列の主な特徴を示す．テスト行列はフロリダ大学の疎行列データベース[7] よりおよそ 100 万次元の行列を目途にして 4 個と比較的小規模行列 1 個の計 5 個を選出した．

表 1　テスト行列の主な特徴

行列	次元数	非零要素数	平均非零要素数
Freescale1	3,428,755	18,920,347	5.5
air-cfl5	1,536,000	19,435,428	12.7
atmosmodd	1,270,432	8,814,880	6.9
tmt_unsym	917,825	4,584,801	5.0
epb3	84,617	463,625	5.5

4.2　実験結果

表2に，元の(同期3回) GPBiCG 法，今回の同期2回 GPBiCG 法，そして比較のため(同期2回)BiCGSafe 法の並列性能の結果を示す．ただし，同期2回 GPBiCG 法　は“ds_GPBiCG”(double synchronized GPBiCG の略)表記した．表中の“np”はプロセス数，“Mv”は行列ベクトル積の演算回数を各々表す．“台数効果”は逐次計算時間に対する平均経過時間の倍率を表す．これは，収束までの反復回数の不規則な変動による“台数効果”の乱れや混乱を防ぐため，通常の合計経過時間ではなく平均経過時間を採用した．“通信割合”は合計経過時間において行列ベクトル積以外の経過時間が占める割合［単位：%］を示す．“Trr”(True relative residual)は真の相対残差の常用対数 $\log_{10}(\|\boldsymbol{b} - A\boldsymbol{x}_{k+1}\|_2/\|\boldsymbol{b} - A\boldsymbol{x}_0\|_2)$ の値である．表中の太字は3つの反復法での最小時間と最大台数効果を表す．**表2**より，以下のことがわかる．

1. 256並列のとき，行列 Freescale1 において同期2回 GPBiCG 法が最も短い時間で収束した．これ

表2　3種類の反復法の並列性能の比較

(a) 行列 Freescale1

反復法	np	Mv 回数	tot-t. [s.]	ave-t. [ms.]	台数効果	通信割合	Trr
GPBiCG (同期3回)	1	12,936	1335.201	103.216	1.00	-	-8.0
	8	10,654	505.036	47.403	2.18	23.52	-8.0
	16	11,050	376.652	34.086	3.03	15.87	-8.0
	32	9,776	168.923	17.279	5.97	15.05	-8.0
	64	9,010	109.602	12.164	8.49	10.54	-8.0
	128	10,952	102.633	9.371	11.01	5.93	-8.0
	256	11,878	78.132	6.578	**15.69**	4.70	-8.0
ds_GPBiCG (同期2回)	1	12,138	1222.973	100.756	1.00	-	-8.1
	8	9,993	452.129	45.245	2.23	23.32	-8.0
	16	9,281	306.498	33.024	3.05	15.46	-8.0
	32	10,441	174.474	16.710	6.03	13.82	-8.0
	64	15,815	184.853	11.688	8.62	8.52	-8.0
	128	9,839	93.818	9.535	10.57	4.67	-8.0
	256	9,553	**61.464**	6.434	15.66	4.17	-8.0
BiCGSafe (同期2回)	1	10,154	958.456	94.392	1.00	-	-8.0
	8	9,498	417.300	43.936	2.15	10.73	-8.0
	16	9,820	316.055	32.185	2.93	6.48	-8.0
	32	9,988	161.852	16.205	5.82	5.84	-8.0
	64	10,002	114.194	11.417	8.27	4.06	-8.0
	128	10,336	96.664	9.352	10.09	2.67	-8.0
	256	10,636	68.851	6.473	14.58	2.47	-8.0

(b) 行列 air-cfl5

反復法	np	Mv 回数	tot-t. [s.]	ave-t. [ms.]	台数効果	通信割合	Trr
GPBiCG (同期3回)	1	50	3.300	66.000	1.00	-	-8.2
	8	48	1.099	22.896	2.88	26.04	-8.2
	16	48	0.592	12.333	5.35	25.93	-8.2
	32	48	0.358	7.458	8.85	24.08	-8.2
	64	48	0.257	5.354	12.33	20.98	-8.2
	128	48	0.164	3.417	19.32	27.91	-8.2
	256	48	0.116	2.417	27.31	33.20	-8.2
ds_GPBiCG (同期2回)	1	48	3.229	67.271	1.00	-	-8.3
	8	51	1.072	21.020	3.20	26.83	-8.2
	16	51	0.580	11.373	5.92	26.64	-8.2
	32	51	0.350	6.863	9.80	22.46	-8.2
	64	51	0.259	5.078	13.25	22.52	-8.2
	128	51	0.163	3.196	21.05	26.96	-8.2
	256	51	0.117	2.294	**29.32**	32.87	-8.2
BiCGSafe (同期2回)	1	52	3.257	62.635	1.00	-	-8.3
	8	50	1.069	21.380	2.93	20.54	-8.2
	16	50	0.569	11.380	5.50	20.27	-8.2
	32	50	0.345	6.900	9.08	17.18	-8.2
	64	50	0.258	5.160	12.14	18.90	-8.2
	128	50	0.163	3.260	19.21	24.60	-8.2
	256	50	0.116	2.320	27.00	31.48	-8.2

(c) 行列 atmosmodd

反復法	np	Mv 回数	tot-t. [s.]	ave-t. [ms.]	台数効果	通信割合	Trr
GPBiCG (同期3回)	1	500	19.103	38.206	1.00	-	-8.1
	8	496	5.167	10.417	3.67	42.95	-8.0
	16	502	2.688	5.355	7.14	38.89	-8.1
	32	498	1.416	2.843	13.44	33.73	-8.2
	64	486	0.794	1.634	23.39	28.98	-8.1
	128	488	0.298	0.611	62.57	35.62	-8.1
	256	500	0.173	0.346	110.42	49.90	-8.1
ds_GPBiCG (同期2回)	1	522	20.914	40.065	1.00	-	-8.2
	8	501	4.826	9.633	4.16	45.70	-8.1
	16	509	2.543	4.996	8.02	41.07	-8.1
	32	501	1.370	2.735	14.65	34.51	-8.1
	64	477	0.762	1.597	25.08	30.96	-8.0
	128	507	0.297	0.586	68.39	35.21	-8.3
	256	525	0.160	0.305	**131.46**	44.66	-8.0
BiCGSafe (同期2回)	1	498	17.660	35.462	1.00	-	-8.5
	8	524	4.903	9.357	3.79	35.26	-8.1
	16	500	2.362	4.724	7.51	30.41	-8.4
	32	496	1.282	2.585	13.72	25.45	-8.2
	64	482	0.733	1.521	23.32	24.51	-8.1
	128	494	0.267	0.540	65.61	29.77	-8.0
	256	500	**0.151**	0.302	117.42	42.61	-8.2

(d) 行列 tmt_unsym

反復法	np	Mv 回数	tot-t. [s.]	ave-t. [ms.]	台数効果	通信割合	Trr
GPBiCG (同期3回)	1	10,622	267.992	25.230	1.00	-	-7.2
	8	10,594	65.337	6.167	4.09	47.98	-4.6
	16	9,748	27.804	2.852	8.85	44.68	-6.7
	32	10,538	12.999	1.234	20.45	39.74	-6.7
	64	10,952	4.584	0.419	60.28	43.07	-6.8
	128	10,512	2.586	0.246	102.56	47.53	-6.4
	256	14,894	3.044	0.204	123.45	62.25	-6.9
ds_GPBiCG (同期2回)	1	10,338	215.087	20.805	1.00	-	-5.2
	8	12,503	70.490	5.638	3.69	52.68	-7.1
	16	11,011	29.517	2.681	7.76	45.48	-6.8
	32	10,731	13.069	1.218	17.08	40.92	-7.0
	64	9,677	3.655	0.378	55.08	39.38	-7.2
	128	10,375	2.204	0.212	97.94	44.04	-5.8
	256	13,711	2.040	0.149	139.84	49.74	-6.6
BiCGSafe (同期2回)	1	10,298	237.381	23.051	1.00	-	-6.7
	8	9,036	48.684	5.388	4.28	39.49	-6.8
	16	9,912	24.029	2.424	9.51	31.50	-6.8
	32	10,182	10.932	1.074	21.47	30.73	-6.9
	64	10,492	3.638	0.347	66.48	33.67	-6.7
	128	9,556	1.918	0.201	114.85	38.33	-6.6
	256	10,054	**1.490**	0.148	**155.54**	47.65	-6.8

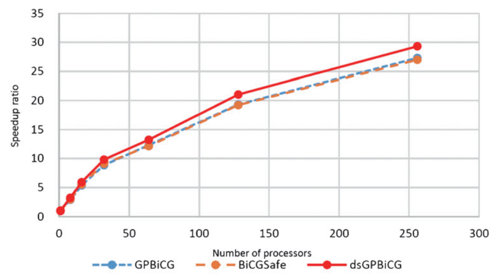
（e）行列 epb3

反復法	np	Mv 回数	tot-t. [s.]	ave-t. [ms.]	台数効果	通信割合	Trr
GPBiCG（同期 3 回）	1	3,774	7.114	1.885	1.00	-	-8.3
	8	3,764	0.942	0.250	7.53	37.45	-8.1
	16	3,872	0.523	0.135	13.96	41.27	-8.1
	32	4,014	0.349	0.087	21.68	47.42	-8.0
	64	3,858	0.297	0.077	24.49	62.79	-8.0
	128	3,968	0.327	0.082	22.87	75.80	-8.1
	256	3,916	0.359	0.092	20.56	78.24	-8.1
ds_GPBiCG（同期 2 回）	1	4,074	7.057	1.732	1.00	-	-8.0
	8	3,807	0.868	0.228	7.60	36.69	-8.2
	16	3,999	0.483	0.121	14.34	37.25	-8.1
	32	3,845	0.310	0.081	21.48	43.47	-8.1
	64	3,811	0.235	0.062	28.09	55.94	-8.0
	128	3,785	0.233	0.062	28.14	68.18	-8.1
	256	4,031	0.274	0.068	25.48	72.97	-8.0
BiCGSafe（同期 2 回）	1	3,626	6.302	1.738	1.00	-	-8.0
	8	3,792	0.833	0.220	7.91	28.86	-8.0
	16	3,998	0.464	0.116	14.98	30.76	-8.0
	32	3,594	0.267	0.074	23.39	38.17	-8.0
	64	3,452	**0.215**	0.062	27.91	52.78	-8.1
	128	3,654	0.223	0.061	**28.48**	68.00	-8.0
	256	3,730	0.255	0.068	25.42	72.01	-8.0

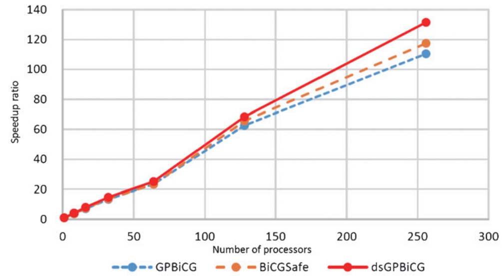
は，行列の次元数の大きさが非常に関係深いと思われる．また，256 並列を超える多並列の場合でも台数効果がまだ伸びる可能性があると思われるが，これは元々台数効果が最大でも 16 倍程度で他の行列の場合よりも相対的に低いのが主な理由と思われる．

2. 同期 2 回 GPBiCG 法は 5 個中 2 個の行列で最も高い台数効果を示し，他の 2 個の行列で BiCG-Safe 法と同等の台数効果を示し，並列性能が高いことを示した．
3. 特に，行列 atmosmodd では 256 並列で 131.46 倍の平均経過時間の台数効果を示した．これは，1 行当りの平均非零要素数が少ないことおよび非零要素の分布，すなわち，対角要素付近の密集度に関係が深いと思われる．
4. 元の GPBiCG 法と比較すると，行列 Freescale1 以外の 4 個の行列で 256 並列の時に同期 2 回 GPBiCG 法の方が高い台数効果を示し，速く収束した．また，行列 Freescale1 の場合の 15.66 倍は，同期 3 回 GPBiCG 法の 15.69 倍と比べて遜色がない台数効果であることがわかる．

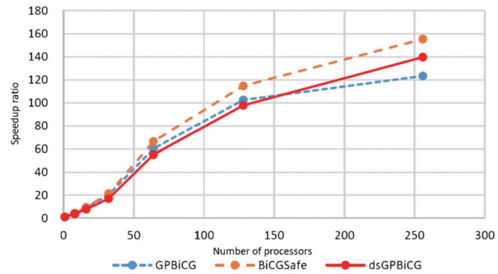
図 1 に 3 種類の反復法の台数効果の増加の様子を示す．**図 1** の結果から，同期 2 回 GPBiCG 法が従来の GPBiCG 法よりも高い台数効果を示したことがわかる．



(a) 行列 air-cfl5



(b) 行列 atmosmodd



(c) 行列 tmt_unsym

図 1　3 種類の反復法の台数効果の増加の様子

5. まとめ

本論文では，パラメータ β_k の計算法の工夫により，同期点が 3 回必要であった元の GPBiCG 法を同 2 回に減らした GPBiCG 法が導出できた．数値実験を通して，同期 2 回 GPBiCG 法は元の GPBiCG 法よりも高い並列性能を示すことがわかった．

参　考　文　献

1）Abe, K., Sleijpen, G.L.G.: Solving linear equations with a stabilized GPBiCG method, Appl. Numer. Math., Vol.67, pp.4–16, 2013.

2）Iwasato, K., Fujino, S., Murakami, K.: A strategy for reduction of number of synchronization points of parallel Krylov subspace methods, HPS 2015, p.104, Hanoi, March 16–20, 2015.

3）Fujino, S., Fujiwara, M., Yoshida, M.: A proposal of preconditioned BiCGSafe method with safe convergence, Proc. of The 17th IMACS World Congress, 2005.

4）村上啓一，藤野清次，同期回数を削減した新しい積型反復法の並列性能評価，情報処理学会研究会，3月，2013.
5）村上啓一：同期点を削減した並列計算向き Krylov 部分空間法の提案，九州大学大学院システム情報科学府修士論文，March，2013.
6）Sonneveld, P., van Gijzen, M. B.: IDR(*s*): a family of simple and fast algorithms for solving large nonsymmetric linear systems, SIAM J. Sci. Stat. Comput., Vol.31, No.2, pp.1035–1062, 2008.
7）Florida Sparse Matrix Collection: http://www.cise.ufl.edu/research/sparse/matrices/index.html.
8）Zhang, S.-L.: GPBi-CG: Generalized product-type methods preconditionings based on Bi-CG for solving nonsymmetric linear systems, SIAM J.Sci.C., pp.537–551, 1997.
9）Zhu, S.X., Gu, T.X., Liu, X.P.: Minimizing synchronizations in sparse iterative solvers for distributed supercomputers, Computers & Mathematics with Applications, Vol.67, Issue 1, pp.199–209, January 2014.

著　者　紹　介

藤野　清次（正会員）

九州大学情報基盤研究開発センター　教授

岩里　洸介

九州大学大学院システム情報科学府修士課程2年