

誤差ベクトルサンプリングに基づく非対称行列向け Subspace Correction 前処理法による複数連立一次方程式の求解高速化

多森 浩俊¹, 深谷 猛², 岩下 武史³

¹ 北海道大学大学院情報科学院, ² 北海道大学情報基盤センター,

³ 京都大学学術情報メディアセンター

e-mail: yuzu-daikonn@eis.hokudai.ac.jp

1 はじめに

本研究では, $\mathbf{Ax}_{(k)} = \mathbf{b}_{(k)}$, ($k = 1, 2, \dots$) で表される, 同一の非対称実行列 \mathbf{A} を係数に持ち右辺ベクトルが異なる複数の n 元連立一次方程式を反復法によって順に解く過程を ES-SC 前処理法 [1] により高速化する. ES-SC 前処理法では, 最初 ($k = 1$) の求解過程で, 近似解ベクトルをサンプリングし, 求解過程終了後, 得られた解ベクトルを用いて, サンプリングした近似解ベクトルに対応する誤差ベクトルを導出することで, 収束が遅い誤差成分を同定し, これらを基底とする部分空間を利用した Subspace Correction 前処理の追加により, 2 番目以降 ($k \geq 2$) の求解過程を高速化する. 非対称行列向け ES-SC 前処理法 [2] では, 前処理行列 \mathbf{M}^{-1} は $\mathbf{M}^{-1} = \mathbf{M}_s^{-1} + \tilde{\mathbf{V}}(\tilde{\mathbf{U}}^\top \mathbf{A} \tilde{\mathbf{V}})^{-1} \tilde{\mathbf{U}}^\top$ で与えられる. ここで第 1 項の \mathbf{M}_s^{-1} は ILU 前処理等の標準的前処理, 第 2 項の $\tilde{\mathbf{V}}(\tilde{\mathbf{U}}^\top \mathbf{A} \tilde{\mathbf{V}})^{-1} \tilde{\mathbf{U}}^\top$ は ES-SC 前処理である. 第 2 項に現われる, $\tilde{\mathbf{V}}, \tilde{\mathbf{U}}$ はそれぞれ, 誤差ベクトルの張る空間からレイリーリッツ法を利用して作成された, 係数行列の小さい特異値に対応する右特異ベクトル, 左特異ベクトルの近似を列ベクトルとして並べた密行列である.

2 数値実験

ILU 前処理付き BiCGSTAB 法に対して, ES-SC 前処理を付加した場合と付加しない場合を比較することで ES-SC 前処理の効果を調査する. 疎行列の公開データベースである SuiteSparse Matrix Collection から選出したテスト行列を係数行列とし, 乱数を用いて生成した異なる右辺ベクトルを持つ 11 個の連立一次方程式を順に解く. 2 から 11 番目の 10 個の方程式の求解過程について, 求解までに要した反復回数と計算時間の平均を比較する.

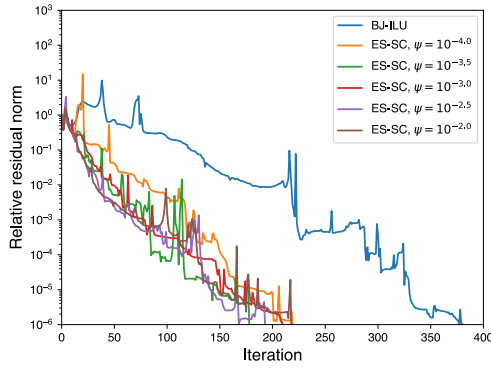
各ソルバの収束判定基準は, 相対残差ノルムが 10^{-6} 以下とした. ES-SC 法において, サンプリングする誤差ベクトルの本数 \bar{m} は 20 とした. リッツベクトルを選択する際の閾値の設定は, $\psi = 10^{-\alpha}$ ($\alpha = 2.0, 2.5, 3.0, 3.5, 4.0$) とし, 作成された補助行列の列数を \tilde{m} で表す. 数値実験は, 北海道大学情報基盤センターのスーパーコンピュータ Grand Chariot の 1 ノードを用いて行った. プログラムは C 言語を用いて作成し, OpenMP によりスレッド並列化を行った.

3 数値実験結果

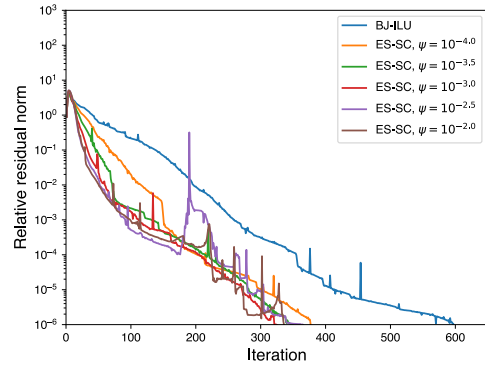
本研究では, 逐次実行時やオーダリングに基づく並列処理時の調査も行っているが, 本稿では, 紙面の都合上, ブロックヤコビ ILU(BJ-ILU) 前処理に対して, ES-SC 前処理を付加した場合の反復回数と計算時間について 2 例を紹介する. 表 1 に, テスト行列として Transport と t2em を用いた場合における, それぞれの手法が収束するまでに要した反復回数と計算時間を示す. また, 図 1 に収束履歴を示す. いずれのデータにおいても, ES-SC 前処理を付加することで, 反復回数を改善していることがわかる. Transport については計算時間も改善しているが, t2em については ES-SC 前処理の使用に伴い 1 反復当たりの計算時間が増えたため, 全体の計算時間が短縮できなかった.

表 1: 40 スレッド並列実行時の反復回数と計算時間（秒）

		Transport			t2em		
precondition	α	\tilde{m}	#ites	time	\tilde{m}	#ites	time
BJ-ILU			390.7	3.36		532.4	1.66
ES-SC-ILU	2.0	15	199.6	3.31	20	337.7	2.76
ES-SC-ILU	2.5	13	186.2	2.79	19	330.9	2.76
ES-SC-ILU	3.0	11	211.3	2.98	17	338.8	2.66
ES-SC-ILU	3.5	8	185.6	2.64	13	387.1	2.57
ES-SC-ILU	4.0	4	207.8	2.43	7	408.7	2.19



(a) Transport



(b) t2em

図 1: 収束履歴

4 まとめ

非対称行列向け ES-SC 前処理法の効果について、ILU 前処理付き BiCBSTAB 法に適用した際の反復回数と計算時間に関する数値実験結果を報告した。逐次実行時やオーダリングに基づく並列処理時の実験結果を含むより詳細な調査結果については、当日の発表で報告する予定である。

謝辞 本研究は、JST 次世代研究者挑戦的研究プログラム (JPMJSP2119)、JSPS 科研費 (23H00462) の支援を受けています。

参考文献

- [1] Iwashita, T., Ikehara, K. Fukaya, T. Mifune, T., Convergence acceleration of preconditioned conjugate gradient solver based on error vector sampling for a sequence of linear systems, Numerical Linear Algebra with Applications, Vol. 30 (2023), pp. e2512.
- [2] Tamori, H., Fukaya, T. and Iwashita, T., Subspace Correction Preconditioning for Solving a Sequence of Asymmetric Linear Systems Using the Bi-CGSTAB Method, Journal of Information Processing, Vol. 31, (2023), pp. 875–884.

複素対称線形方程式に対する COCR 法の精度切り替えについて

梶井 晃基¹

¹ 大阪大学

e-mail : masui@ist.osaka-u.ac.jp

1 背景

大規模高周波電磁場解析 [1] の過程で現れる反復法は、その問題の特性から収束性が悪いことが知られている。これらの問題は、その係数行列が複素数かつ疎であるため、共役直交共役勾配 (Conjugate Orthogonal Conjugate Gradient: COCG) 法 [2] や、共役直交共役残差 (Conjugate Orthogonal Conjugate Residual: COCR) 法 [3] などの Krylov 部分空間法が用いられる。これらの反復法の収束性は丸め誤差の影響を受けるため、さらに収束性は悪化し、問題によっては倍精度演算では収束しないことがある。

これに対し、倍々精度演算などの高精度演算を用いることで、丸め誤差の影響を小さくし、安定的に収束解を得ることができるが、1 反復あたりの計算コストが増えるため、計算時間が削減されるケースは少ない。また、これらの研究の多くは、反復法の計算中で使う変数の計算精度は不変である。計算時間の削減を目的として計算の途中で精度を変えた反復法を実装した研究例 [4,5] もあるが、有効な計算精度の切り替え手法については不明である。そこで、本稿では反復法で計算精度を変更する手法を提案および実装し、その性能について報告する。

2 倍々精度演算

本稿で用いる、高精度演算である倍々精度演算について述べる。シミュレーションの数値計算においては、IEEE754 準拠の倍精度浮動小数点数 (binary64) が広く使われており、10 進数にすると約 16 桁程度の精度である。これに対し、倍々精度演算は倍精度演算数を 2 つ用いて、それぞれの仮数部を上位ビットと下位ビットと割り当てることで、4 倍精度より精度は少し劣るものの、高速かつ高精度な演算を実現している。本稿で対象としている問題は複素数問題であるので、この倍々精度演算を実部と虚部に組み込むことで、複素数の倍々精度演算を実装している。

3 提案手法

丸め誤差の影響を大きく受ける反復法においては、その残差が大きく振動したり、停滞したりすることで、収束までの反復回数が増える。また、収束せず停滞する場合は残差ノルムが初期残差ノルムからあまり落ちず、そのまま停滞することが多い。これを防ぐために、始めに倍々精度で計算し、その後計算精度を倍精度に切り替える反復法を提案する。本稿では、まず、精度切り替えの有効性を確認するために、手動で精度の切り替えタイミングを設定した時の結果について報告する。

4 数値実験

倍精度では収束しない問題として、SuiteSparse Matrix Collection[6] にある行列 qc2534 に対して、前処理無し COCR 法を適用することを考える。ただし、qc2534 は複素対称行列であり、初期解 $x^0 = 0$ とする。最終的な収束判定 τ は相対残差ノルム $\|r^n\|_2 / \|r^0\|_2$ が $\tau \leq 10^{-9}$ となったときとし、反復法の途中で計算精度を倍々精度から倍精度に切り替え、精度を切り替える判定 τ' を $\tau' = 10^{-6}$ から $\tau' = 10^{-8}$ まで変化させたときの性能を評価した。

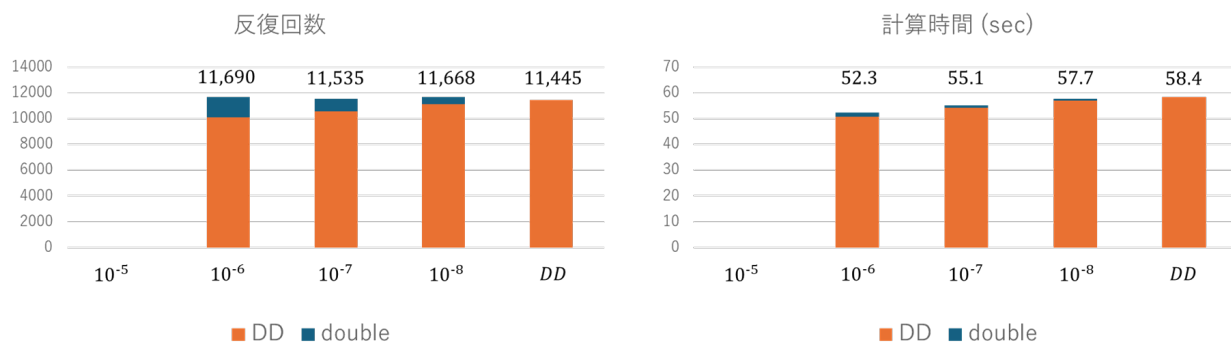


図 1. qc2534 に対する性能評価

図 1 に全て倍々精度で計算した時、および途中で計算精度を変えた提案手法の反復回数と計算時間を示す。ただし、横軸は精度を切り替えるタイミング τ' を示しており、 τ' が 10^{-5} より大きい場合は反復法が収束しなかったため、結果は載せていない。

図より、提案手法は倍々精度演算と比べて、計算時間を最大 10.4% 削減することに成功した。ただし、切り替えるタイミングによっては収束しないことがあるので、注意が必要である。

5 まとめ

本研究では、計算精度を途中で切り替える反復法を実装し、倍精度では収束しない問題に対する性能を評価した。その結果、計算時間を最大で 10.4% 削減し、提案手法の有効性を示した。今後は、精度を自動で切り替える手法や、複数回切り替える手法について検討していく。

謝辞

本研究は JSPS 科研費 24K20780 の助成を受けたものである。

参考文献

- [1] 武居 周, 吉村 忍, 金山 寛, 階層型領域分割法による高周波電磁場の大規模解析, 電気学会論文誌 A, Vol.128-A, No. 9 (2008), pp. 591–597.
- [2] Van der Vorst, H. and Melissen, J, A Petrov-Galerkin type method for solving $Ax=b$ and where A is symmetric complex, Trans. Magn., Vol. 26 (1990), 706–708.
- [3] Sogabe, T. and Zhang, S, A COCR method for solving complex symmetric linear systems, J. Comput., Vol. 199-2 (2007), pp. 297–303.
- [4] 小武守 恒, 藤井 昭宏, 長谷川 秀彦, 西田 晃, 反復法ライブラリ向け 4 倍精度演算の実装と SSE2 を用いた高速化, 情報処理学会論文誌コンピューティングシステム, Vol. 1, No. 1 (2008), pp. 73–84.
- [5] 梶井 晃基, 伊野 文彦, 高周波電磁場解析における反復法の計算精度の切り替えについて, 研究報告ハイパフォーマンスコンピューティング (HPC), 2022-HPC-187, pp. 1–7, 2022.
- [6] SuiteSparse Matrix Collection, <https://sparse.tamu.edu/>.

ICCG 法における反復回数と前処理後の係数行列の固有値分布の関係性

岩下 武史¹, 深谷 猛²

¹ 京都大学学術情報メディアセンター, ² 北海道大学情報基盤センター
e-mail : iwashita@i.kyoto-u.ac.jp

1 はじめに

ICCG 法（不完全コレスキー分解前処理付き共役勾配法）は、応用分野の解析で最もよく用いられる前処理付き（クリロフ部分空間）反復法の一つである。反復法による求解時間は、予め定めた相対残差ノルム等の収束判定基準を満たすまでに必要な反復回数に概ね比例するため、反復あたりどの程度残差や誤差のノルムが低減するかを意味する収束特性は反復法の重要な性能指標である。主に対称正定値な係数行列を持つ連立一次方程式を対象とする CG 法では、係数行列の条件数により収束特性を評価できることが知られており、また固有値分布が収束特性に与える影響も大きいと考えられている [1]。しかしながら、実応用分野で解かれる問題のサイズが一般に大きいことから、計算コストの大きい固有値解析により収束特性を分析した例は少なく、特にブロックヤコビ IC 前処理等の並列処理向けの IC 分解前処理について、前処理後の係数行列の固有値分布を調査した例はこれまでにあまり見られない。そこで、本稿では、行列データベースである SuiteSparse Matrix Collection から取得した行列を用いて、これらを係数行列とする連立一次方程式を ICCG 法で解き、その反復回数を調査すると共に、前処理後の係数行列の固有値分布について調査する。

2 数値実験

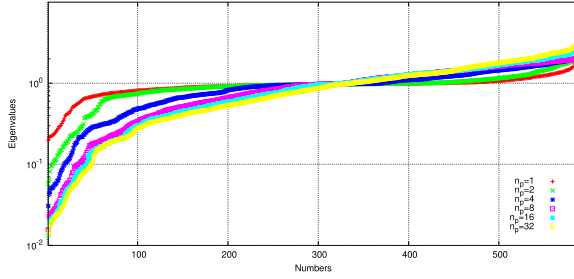
北海道大学情報基盤センターの Grand Chariot システム、1 ノードを用いて数値実験を行った。プログラミング言語には Fortran を使用し、固有値計算には LAPACK の DSYEV ルーチンを使用した。コンパイラは、Intel Fortran Compiler version 19.1.3.304 を使用し、コンパイルオプションには、`-qopenmp`, `-O2`, `-ip` を指定した。収束判定基準は相対残差ノルム（2-ノルム）が 10^{-7} 以下となる条件とし、反復開始から残差ベクトルの 2-ノルムが本基準を満たすまでの反復回数を調査した。右辺ベクトルにはすべての要素を 1 としたベクトルを用いた。IC 分解前処理では、しばしば分解の破綻を防ぐために、係数行列をシフトした行列に対して IC 分解を行い、得られた行列を前処理行列として用いることが行われる。本数値実験では、データセット bcsstk34 に対して、本技術を用いた。シフトの方法については選択肢がいくつかあるが、本数値実験では係数行列の対角成分を 1.03 倍した後に IC 分解を行う方法を採用した。

3 数値実験結果

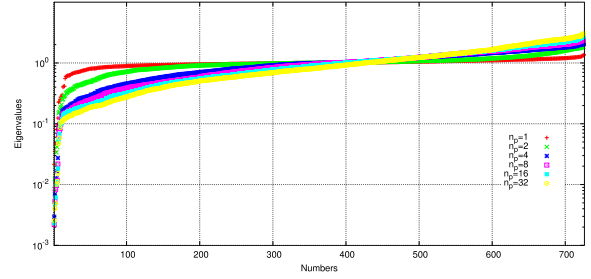
本研究では、未知変数のオーダリングの影響や前処理行列を単精度化した場合についても調査を行っているが、本稿では、紙面の都合上、ブロックヤコビ IC (BJ-IC) 前処理において、並列数（ブロック数）を変化した場合の反復回数と前処理後の係数行列の固有値分布について、2 例を紹介する。BJ-IC 前処理では、並列処理のために係数行列をスレッド数やプロセス数で行・列共に分割し、得られた対角ブロックに対して IC 分解を行い前処理行列とする。即ち、対角ブロック以外の非ゼロ要素は分解中に全て無視する。この操作により、反復中の前処理過程である前進・後退代入計算は、ブ

表 1. 反復回数, Remainder matrix のノルム

(a) bcsstk34			(b) msc00726		
並列数	反復回数	R_F	並列数	反復回数	R_F
1	25	6.47×10^6	1	37	1.32×10^9
2	47	3.45×10^7	2	63	5.23×10^9
4	60	5.87×10^7	4	87	8.82×10^9
8	88	8.80×10^7	8	103	1.02×10^{10}
16	104	9.75×10^7	16	111	1.09×10^{10}
32	118	1.01×10^8	32	122	1.18×10^{10}
40	123	1.02×10^8	40	119	1.19×10^{10}



(a) bcsstk34



(b) msc00726

図 1. 前処理後の係数行列の固有値分布 (n_p : 並列数)

ロック単位で並列化できる。BJ-IC 前処理では、並列数を増加した場合、分解中に無視される非零要素が増加するため、一般に前処理効果が低下する。本数値実験では、その影響について分析を行った。

表 1 に係数行列データとして、bcsstk34 と msc00726 を用いた場合の並列数の変化に対する BJ-IC 前処理付き CG 法の反復回数を示す。また、ICCG 法の収束特性の評価指標の一つである Remainder matrix のフロベニウスノルム（本稿では R_F と表記）[2] についても併せて掲載する。表 1 より、いずれのデータセットにおいても、並列数を増加させるに従い、反復回数と R_F 値が増加する傾向が見て取れる。次に、図 1 に前処理後の係数行列の固有値分布を示す。いずれのデータセットにおいても、並列数が増加するに従い、小さい固有値はより小さく、大きい固有値はより大きく変化し、条件数が悪化していることが分かる。また、多くの固有値が変化しており、例えば減次法によって収束性の劣化に対処することが難しいことを示唆している。

4 まとめ

ブロックヤコビ IC 前処理付き CG 法の並列数（ブロック数）の影響について、反復回数、前処理後の係数行列の固有値分布の調査を行い、報告した。他の IC 前処理に関する結果を含む、より詳細な調査結果については、年会会場で報告する予定である。

参考文献

- [1] Y. Saad, Iterative Methods for Sparse Linear Systems, Second ed., SIAM, 2003.
- [2] I. S. Duff and G. A. Meurant, BIT, 29 (1989), pp. 635–657.

通信・計算オーバーラップによる並列多重格子法の最適化

中島 研吾¹

¹東京大学情報基盤センター

e-mail: nakajima@cc.u-tokyo.ac.jp

1 概要

大規模な並列計算機を使用する場合、ノード数の増加によって通信のオーバーヘッドは増加する傾向にある。並列計算において通信は必須のプロセスであるが、通信をできる限り効率的に実施し、削減することは EFLOPS 級システムにおいて重要である。通信の削減がアルゴリズムに不安定をもたらす可能性もある。有限要素法、差分法等から得られる大規模疎行列の解法としては共役勾配法 (Conjugate Gradient, CG) に代表されるクリロフ部分空間法が広く使用されている。通信が生じる代表的なプロセスとしては、①疎行列ベクトル積 (1 対 1 通信)、②内積 (集団通信) がある。著者等は両者について検討を実施してきたが [1, 2], 本稿では特に①に関する検討を中心に実施する。

2 通信・計算のオーバーラップ

並列有限要素法, 有限体積法などにおける Halo を含む並列データ構造において, 内点を, Halo と接している点 (境界点) とそうでない点 (純内点) にわけて並び替え, 後者の計算とプロセス間をオーバーラップさせる方法が考えられる [1, 2]. この手法は, 本来ステンシル計算 (陽解法) のために考案されたものである. [2] では, Halo 通信に OpenMP の動的ループスケジューリング機能を適用している. この手法では, 通信をマスタースレッドが受け持ち, 純内点の計算とオーバーラップさせる. 純内点の部分の計算には, OpenMP の動的ループスケジューリング機能を適用し, 適当なチャンクサイズを設定し, 通信終了次第マスタースレッドが計算に参加している. 著者等は, 従来, 陽解法, 疎行列ベクトル積に使用されてきた, 通信・計算オーバーラップを ICCG 法, 多重格子法のスムージングにおける前進後退代入への通信・計算オーバーラップの適用に関する研究を実施してきた [3, 4]. 本稿では, 三次元不均質多孔質媒体中の地下水流れを有限体積法 (FVM) によって解くプログラム「pGW3D-FVM [3, 4]」の MGCG 法ソルバー (多重格子法前処理付き共役勾配法) を対象として, 動的ループスケジューリングの他, マスタースレッドを Halo 通信に関連した処理にのみ適用する Manual Loop Scheduling [5] について検討し, Wistereia/BDEC-01 (Odyssey) (東大情報基盤センター) [6] の最大 4,096 ノードを使用した計算事例について紹介する。

3 Manual Loop Scheduling

動的ループスケジューリング適用による通信・計算オーバーラップによって, 性能向上が得られることは多くの事例で示されているが [2, 3], Halo 通信の進行をチェックするループサイクル数 (Chunk Size) の最適値が, 問題サイズ, ハードウェア環境によって変動する。

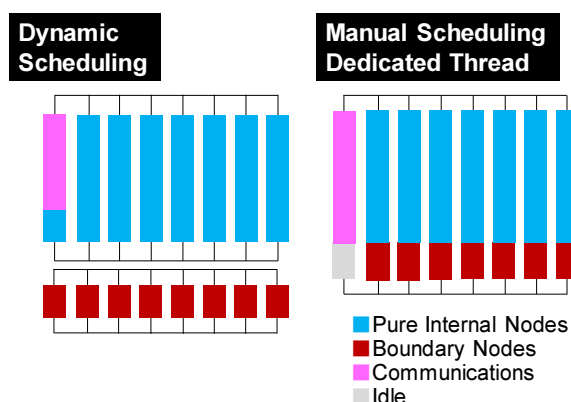


Fig.1: Manual Loop Scheduling [5]

Manual Loop Scheduling (Manual Scheduling) [5] は、マスタースレッドを Halo 通信への利用に限定することによって、オーバーヘッドを削減する手法である. Fig. 1 に動的ループスケジューリング (Dynamic Scheduling) と Manual Scheduling の比較を示す.

4 計算結果・まとめ

Fig. 2 は、1 ノード当たりの問題規模を $128 \times 64 \times 64$ メッシュに固定し、ノード数を 128 から 4,096 まで変化させた場合の Weak Scaling の結果を示す. 各ケースの頭の「u」は問題規模 (ultra-tiny の略, ノード当たり問題規模が最も小さい場合), org: オリジナル実装, sta: 静的ループスケジューリング, dyn: 動的ループスケジューリング, man: Manual Scheduling, 前進後退代入への通信・計算オーバーラップの有無 (無し: a, 有り: b) となっている. Fig. 3 は各ケースにおける

MGCG 法の計算時間を, u-org の 128 ノードにおける時間で正規化した値である. Manual Scheduling により, 動的ループスケジューリングと比較して 5%程度, 更に前進後退代入への通信・計算オーバーラップ適用により 10%程度の性能向上が見られる. オリジナル実装と比較して最大 40%程度の速度向上が得られていることがわかる. ノード当たりの問題規模が大きい場合は, 前進後退代入への通信・計算オーバーラップ適用により性能が低下する場合もある [4]. ノード当たりの問題規模が大きい場合の提案手法の性能向上については, 今後の課題である.

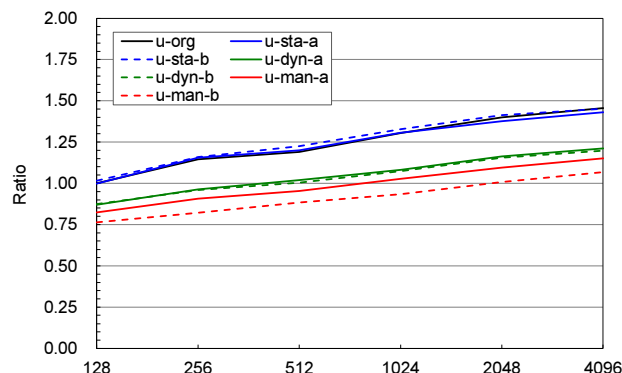


Fig.2: Computation Time for MGCG Solver, $128 \times 64 \times 64$ meshes/node, 128-4,096 nodes of Odyssey, normalized by computation time of u-org at 128 nodes

参考文献

- [1] 塙敏博, 中島研吾, 大島聡史, 星野哲也, 伊田明弘, パイプライン型共役勾配法の性能評価, 情報処理学会研究報告, 2016-HPC-157 (6), 2016
- [2] Nakajima, K., Hanawa, T., IEEE Proceedings of ICCPPW 2017, 210-219, 2017
- [3] 中島研吾, 通信・計算オーバーラップによる並列多重格子法, 情報処理学会研究報告, 2022-HPC-187 (13), 2022
- [4] Nakajima, K., "Communication-Computation Overlapping for Parallel Multigrid Methods," in 2024 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), San Francisco, CA, USA, 2024 pp. 751-760, <https://doi.ieeecomputersociety.org/10.1109/IPDPSW63119.2024.00139>
- [5] Soga, T., Yamaguchi, K., Mathur, R., Watanabe, O., Musa, A., Egawa, R., Kobayashi, H., Effects of Using a Memory Stalled Core for Handling MPI Communication Overlapping in the SOR Solver on SX-ACE and SX-Aurora TSUBASA, Supercomputing Frontiers and Innovation 7-4, 4-15, 2020
- [6] Wisteria/BDEC-01: <https://www.cc.u-tokyo.ac.jp/en/supercomputer/wisteria/service>