

精度保証付き数値計算ライブラリの運用に向けて

片桐孝洋¹⁾, 石黒史也²⁾, 荻田武史³⁾, 尾崎克久⁴⁾, 大島聡史¹⁾, 永井亨¹⁾

- 1) 名古屋大学 情報基盤センター
 - 2) 名古屋大学 大学院情報学研究科
 - 3) 東京女子大学 現代教養学部
 - 4) 芝浦工業大学 システム理工学部
- katagiri@cc.nagoya-u.ac.jp

Toward Operation for Verified Numerical Computation Libraries

Takahiro Katagiri¹⁾, Fumiya Ishiguro²⁾, Takeshi Ogita³⁾, Katsuhisa Ozaki⁴⁾, Satoshi Ohshima¹⁾,
Toru Nagai¹⁾

- 1) Information Technology Center, Nagoya Univ.
- 2) Graduate School of Informatics, Nagoya Univ.
- 3) Division of Mathematical Sciences, Tokyo Woman's Christian Univ.
- 4) College of System Engineering and Science, Shibaura Institute of Tech.

概要

本発表では、得られた計算結果の理論的な検証がほとんど考慮されていなかった数値計算シミュレーションにおいて、演算精度を保証する基盤を提供する試みである精度保証付き数値計算ライブラリの開発について述べる。また、開発した数値計算ライブラリを稼働中のスーパーコンピュータで提供する際の運用と普及に関する方法を述べる。

1 はじめに

近年の計算機は、CPUのマルチコア化、Graphics Processing Unit (GPU)の活用がなされている。一方、最高性能を達成するスーパーコンピュータだけではなく、サーバやPCにおいても非常に大規模かつ複雑なコンピュータシミュレーションが行われるようになってきている。このような計算機環境では、計算結果を得るまでに大量の演算やデータ処理が必要となる。ところが、演算精度について、以下の問題が発生している：

- 精度劣化 (数値計算における誤差の増大)
- 計算結果の不安定化 (計算プログラムの最適化/並列化にともなう再現性の喪失)

すなわち、高性能計算 (HPC) 分野で行われる数値計算処理の計算の品質を向上させつつ、計算結果の信頼性を高めるアルゴリズムやフレームワークを開発する必要がある。

そこで本研究では、精度保証付き数値計算ライブラリを開発することを目的にする。さらに、運用中のスーパーコンピュータで開発ライブラリを提供することで、数値シミュレーションのユーザ

に精度保証計算を普及させることを目指す。

2 精度保証付き数値計算ライブラリ

2.1 プロジェクト概要

本開発は、平成28年度採択課題、ポスト「京」萌芽的課題「極限の探究に資する精度保証付き数値計算学の展開と超高性能計算環境の創成」で行われている。本プロジェクトでは、HPCにおいて「演算精度」の軸を新たに導入する。すなわち、正しい計算結果を得るために、最高性能の計算環境を提供するフレームワークを確立することにある。このことは、スーパーコンピュータの新しい指針を定めることも含まれている。そのため、必要なベンチマークの設定を学問的に検討することも目標の1つである。

多くの場合、解くべき問題に対する演算性能は以下のように定義される：

$$\text{性能} = \text{実行速度} \times \text{演算精度} \quad \dots(1)$$

ここで、「実行速度」は「計算量/計算時間」、

「演算精度」は「数値解の正しい桁数」を表す。実行速度が大きいとしても、その数値解の演算精度が非常に低い場合は、式(1)の定義による性能は低くなる。一方、数値解の演算精度が高くても、実行速度が小さい場合は、性能は低くなる。結論として、実行速度と演算精度のバランスが取れる場合に高性能であると定義される。

著者の荻田らによって開発された独自の数値計算法である、数値線形代数における高速精度保証法[1] や、エラーフリー変換法[2][3] を活用すると、スーパーコンピュータ「富岳」に代表される高性能スーパーコンピュータ上でも、数学的に正しい結果を数値計算によって実用的に得ることが可能となる。したがって、式(1)で定義する性能を高めることができる。このことにより、スーパーコンピュータでしか取り扱うことのできない超大規模計算を必要とするユーザにおいて、演算精度の観点で解くことが困難であった様々な難問の解決ができる。

2.2 研究体制

本プロジェクトの研究体制は以下のとおりである。荻田を中心として、「超高性能計算環境向け精度保証付き数値計算法の開発」、および、「アプリケーションソフトウェアの精度保証化・高精度化」に取り組む。早稲田大学の柏木を中心として、「京」コンピュータ、および、スーパーコンピュータ「富岳」における精度保証付き数値計算アルゴリズムの開発及び実装に取り組む。片桐を中心として「京」コンピュータ、および、スーパーコンピュータ「富岳」におけるベンチマークの整備に取り組む。尾崎を中心に、超高性能計算環境向け高精度数値線形代数アルゴリズムの開発に取り組む。

2.3 開発内容

本プロジェクトでは、以下の数値目標を設定した。

1. 密行列系では 100 万次元、スパース系では数億次元の問題（連立一次方程式、固有値問題等）について、所望の精度を持つ解を得られることを示す。
2. 問題の困難さに応じ、従来の近似計算の数倍から数十倍で精度保証が可能となることを示す。

成果物として、本研究課題終了時にスーパーコンピュータ「富岳」向け、および、その他の計算機向けに汎用化した数値計算ライブラリを世界に公開する。そして、スーパーコンピュータ「富岳」

の運用開始 5 年後までに、本研究成果のアウトリーチ活動を展開し、他に類を見ない「演算精度」の軸を持ったスーパーコンピュータ「富岳」の超高性能性を世界中に認知させることを目指す。

アウトカム成果として、スーパーコンピュータ「富岳」運用開始 5 年後までに、演算精度に関する問題でこれまでに解くことが困難であった基礎科学における各種の難問に、本研究成果が適用され始め、運用開始 10 年後には、いくつかの難問が解決されることを期待している。

本研究課題の成果を、最新計算機であるスーパーコンピュータ「富岳」を用いて普及することで、超高性能計算環境の概念はスーパーコンピュータからワークステーション、PC のレベルまでダウンサイジング化されることを示す。最終的に、あらゆる計算機上での様々な計算結果が精度保証化されるようになる。

また、名古屋大学情報基盤センターの共同利用・共同研究拠点のスーパーコンピュータで提供することで新規ユーザの拡大を狙うとともに、第二階層のスーパーコンピュータでの精度保証付き数値計算プログラム開発を促進し、スーパーコンピュータ「富岳」における同プログラム開発を支援する。

以上のことにより、数値計算を必要とする諸分野において計算結果の品質の劇的向上を目指している。

2.4 直近の研究成果

実対称行列に対する固有ベクトルの反復改良アルゴリズムの開発

実対称行列に対して、高精度な固有ベクトルを得るための反復改良アルゴリズムを開発した[4][5]。本アルゴリズムでは行列積が主計算となっているため、計算機の性能を引き出しやすいという特徴がある。また、本アルゴリズムを十分な演算精度を用いて実行することによって、2 次収束することが理論的に示されている。

固有ベクトルの計算で問題となるのは、近接した固有値がある場合である。そのような場合、固有ベクトルの精度が非常に悪くなる。応用上、高精度な固有ベクトルを得ることが必要な分野もあるため、固有ベクトルの精度を改善することは重要である。文献[4]では、固有ベクトルの反復改良の基本アルゴリズムを提案した。十分に固有値が離れている場合、この基本アルゴリズムを適用可能である。また、文献[5]では、基本アルゴリズム

をベースとして、さらに密集した固有値を検知し、それらに対応する固有ベクトルの精度を改善するアルゴリズムを提案した。

数値実験によって提案アルゴリズムの有効性を示した。具体的には、固有値が密集している場合も含め、様々な固有値分布を持つ行列に対して提案アルゴリズムを適用し、数回の反復で高精度な固有ベクトルが得られることを確認した。また、計算速度については、多倍長精度計算と比較し、提案アルゴリズムが数倍から数十倍程度高速であることを示した。

連立一次方程式に対する精度保証アルゴリズムの実装

大規模分散並列計算機において、密行列を係数とする連立一次方程式の精度保証付き数値計算アルゴリズムの開発・実装とその性能評価を行った。

まず、数値線形代数で使用される関数の性能評価として、精度保証に必要な PBLAS や ScaLAPACK のルーチンの性能を調べ、実装に用いる関数の決定を行った。

特に逆行列の計算においては、`pdgetrf` と `pdgetri` を組み合わせて用いるよりも、`pdgesv` を用いるほうが 20% 程度高速であることがわかった。

また、行列積を計算するルーチンの性能が線形方程式を解くルーチンの性能より良いため、近似解を得る時間と精度保証を行う実時間の比は、計算量の比に対してより小さくなることを確認した。

また、現状の数値計算の精度を把握するために、誤差が厳密に分かるテスト問題の生成法を開発した。精度保証のアルゴリズムとしては、文献[6]の方式を用いた。まずはアルゴリズムの性質を調べるため、特定の問題から現れる行列ではなく、疑似乱数行列を係数行列とした連立一次方程式 $Ax = b$ の数値解 bx の誤差を、「京」コンピュータを用いて確認した。

高精度計算を用いた残差反復を行うことにより、100 万円を超える連立一次方程式においても、真の解が浮動小数点数である場合、その真の解自身を正確に数値計算で求められることを確認できた。

高性能実装技法の開発

精度保証付き数値計算ライブラリの開発だけでなく、スーパーコンピュータの計算機アーキテクチャ特性を考慮した高性能実装技法の開発も行う。そのため、精度保証計算における単体性能や通信性能の高効率化を行う。特に近年の計算機アーキテクチャにおいては、精度保証計算の問題

レベルの並列性の利用、階層メモリ最適化（キャッシュブロッキングや NUMA (Non Uniform Memory Access)最適化、など) を考慮しなくてはならない。また、CPU においても、50 コア程度のマルチコア CPU に加えて、GPU (Graphics Processing Unit) を考慮した高性能実装技法を開発する必要がある。

以上に加えて、精度保証計算で用いる行列は、入力が密行列でも、エラーフリー変換法で現れる行列のいくつかは疎行列になる。そのため、疎行列演算に対する高性能実装技法を開発する必要がある。

本プロジェクトで開発した数値計算ライブラリには、精度保証計算特有のチューニングパラメータが存在する。例えば、倍精度演算の精度限界まで保証する高性能行列-行列積アルゴリズム（尾崎の方法）では、疎行列と見なす疎度、疎行列圧縮形式における疎行列-ベクトル積演算 (SpMV (Sparse Matrix-Vector Multiplication)) において同時に計算する複数右辺の数、および、実装方式選択（数値計算ライブラリ BLAS における `dgemm` ルーチンを使う実装方式や疎行列圧縮方式を用いて SpMV する実装方式）など[7]がチューニングパラメータとなる。

先行研究[8]では、「京」コンピュータ型スーパーコンピュータ FX100 において、精度保証のための無誤差変換を行う際、密行列から疎行列になるという特性を利用し、密行列から疎行列への変更を行い、かつスレッド並列化方式を SpMV 演算する高性能実装方式を評価した。その結果、密行列としてベンダーが提供した高性能 BLAS(`dgemm`) での実行時間に対し、疎行列演算を用いた実装方式を採用すると、疎行列化による演算量の削減により、最大で 38 倍の速度向上が得られることを明らかにした。

2.5 成果物の公開

本プロジェクトで開発した精度保証付き数値計算ライブラリ群は、オープンソースコードとして、VNC-HPC (Verified Numerical Computations High-Performance Computing) の HP (<http://www.math.twcu.ac.jp/ogita/post-k/index.html>) で公開している。2019 年 9 月現在、公開されている精度保証付き数値計算ライブラリは以下である：

- **LINSYS_VR: Verified Solution of Linear Systems with Directed Rounding**

[for K Computer, FX100]

連立一次方程式に対する精度保証プログラム (丸めモードの変更機能付き)

● **LINSYS_V: Verified Solution of Linear Systems**

[for PC, K Computer, FX100]

連立一次方程式に対する精度保証プログラム

● **DHPMM_F: High-precision Matrix Multiplication with Faithful Rounding**

[for PC, K Computer, FX100]

高精度行列積プログラム

● **OzBLAS**

[for PC, GPU]

尾崎スキームに基づく高精度基本線形計算ライブラリ

3 典型的な性能例

ここでは、本プロジェクトで開発した精度保証付き数値計算ライブラリの一部について性能評価した結果を紹介する。

3.1 計算機環境

名古屋大学情報基盤センター設置の「京」コンピュータ型スーパーコンピュータ FX100 を利用する。詳細は以下である。

●マシン名 : Fujitsu PRIMEHPC FX100

- プロセッサ名 : Fujitsu SPARC64 XIfx
- プロセッサ数(コア数) : 2 ソケット(32 コア)
- 周波数 : 2.2 GHz
- 理論演算性能 : 1126.4 GFlops/node
- メモリ容量 : 32 GB/node
- メモリ帯域幅 : 480GB/sec/node
- コンパイラ : Fujitsu Fortran90 コンパイラ Version 2.0.0 P-id: T01815-02 (Jul 12 2018 09:17:03)
- コンパイラオプション : -Kfast

東京大学情報基盤センター設置のスーパーコンピュータ Reedbush-H を利用する。詳細は以下である。

●マシン名 : SGI Rackable C1102-GP8

➤ CPU

- ◇ プロセッサ名 : Intel Xeon E5-2695v4 (Broadwell-EP)
- ◇ プロセッサ数(コア数) : 2 ソケット(36 コア)

◇ 周波数 : 2.1 GHz (Turbo boost 時最大 3.3 GHz)

◇ 理論演算性能 : 1209.6 GFlops/node

◇ メモリ容量 : 256 GB/node

◇ メモリ帯域幅 : 153.6 GB/sec/node

◇ コンパイラ : Intel C++ コンパイラ ver.18.1.163

◇ コンパイラオプション : -xHost -O3 -qopenmp -mkl

➤ GPU

◇ NVIDIA Tesla P100 (Pascal)

◇ コア数 (単体) : 56 コア

◇ メモリ容量 (単体) : 16 GB

◇ メモリ帯域幅 (単体) : 732 GB/sec

◇ 理論演算性能 (単体) : 5.3 TFlops

◇ 搭載数 : 2 台

◇ コンパイラ : Intel C++コンパイラ version 18.1.163

◇ オプション : -xHost -O3 -fPIC -DNDEBUG -DADD_ -fp-model strict -qopenmp -lmkl_intel_lp64 -lmkl_intel_thread -lmkl_core -liomp5 -lpthread -lcublas -lcudart -lcusparse

3.2 性能評価

連立一次方程式に対する精度保証

高精度計算法を行列-ベクトル積に拡張し、分散並列化を行った。PBLAS の pdgemv と同じプログラム・インタフェース (API (Application Programming Interface)) で利用可能である。連立一次方程式の数値解を求めるために要した時間と精度保証法に要した時間の比を表 1 に示す。

表 1 FX100 での実行時間の比

行列次元	利用ノード数	計算時間比
12 万元	36	5.12
24 万元	144	4.75
36 万元	324	4.51
48 万元	576	4.12

精度保証は近似計算を得るコスト (演算回数) の 9 倍程度を要するが、実測は 9 倍以下である。

この理由は、線形方程式の求解(pdgesv)と行列-行列積(pdgemm)の対ピーク性能の差である。

高精度行列-行列積 (尾崎の方法)

尾崎の方法において行列サイズが大きいときの主演算は、エラーフリー変換法中で生じる多数の行列-行列積である。ここで、行列サイズが小さいと高性能 BLAS ライブラリ(dgemm)を用いても性能が出ないことが知られている。そこで、Batched BLAS を利用し、高速化する実装方式を採用した。

ここで Batched BLAS は、複数の行列-行列演算について、同種の BLAS 演算を並列実行する API を提供する。本プロジェクトでは、その API 中の GEMM 演算について記載する。従来の BLAS ライブラリで計算機アーキテクチャ上の並列資源を十分に使い切れない小規模サイズの複数の行列-行列積を、その演算をまとめて同時計算することで、小さな GEMM 演算を単純に連続して呼び出す場合に対して高性能化する。したがって、行列サイズが小さい時の尾崎の方法の主演算に Batched BLAS を適用するメリットがある。

以下の式 (2) に示す、多数の組の GEMM 演算を並列計算することを考える。

$$C_i \leftarrow \alpha_i A_i B_i + \beta_i C_i \quad i = 1 : N \quad \dots \quad (2)$$

ここで、Batched BLAS では、式 (2) の行列サイズと α_i , β_i の値を、1 回の呼び出し (バッチ) 全体で一定に保つか、可変にするかをアプリケーションに応じて、個別に設定できる。

本性能評価では、すべての行列-行列積演算を倍精度で行った。CPU 環境においては、1 ノードでスレッド数を 1 スレッドから 36 スレッドまで変化させて実行時間を測定した。Batched BLAS は、Intel Math Kernel Library 2018.1.163 で提供されている Batched 機能を用いた。ここでは、Intel Math Kernel Library の Batched 機能の中の関数 cblas_dgemm_batch を用いた。

GPU 環境では CUDA 8.0.44 を使用した。CPU のスレッド数を 35 に固定して測定した。今回、35 スレッドを利用した理由は、基本的に実行時間はスレッド数を増やすと減少したが、36 スレッドの時だけ遅くなったためである。Batched BLAS については、MAGMA ver.2.2.0 の関数 magmablas_dgemm_batch を用いて、性能評価を行

った。cuSPARSE については、CUDA 8.0.44 の関数 cusparseDcsrmmv を使用した。

CPU および GPU 双方の実行環境において、実行時には NUMA Affinity 設定を行っている。それは、KMP_AFFINITY=granularity=fine,compact1,0 と numactl の指定である。また、性能評価では、すべての時間計測に関数 omp_get_wtime を用いた。ただし GPU を利用した部分の時間計測だけ omp_get_wtime の前に cudaThreadSynchronize を追加した。

図 1、図 2 に東京大学の Reedbush-H での実行時間を示す。なお、図中の error_free はエラーフリー変換法の実行時間、kernel は複数の行列-行列積の時間、other_kernel はこの 2 つの実行時間以外の実行時間を示している。

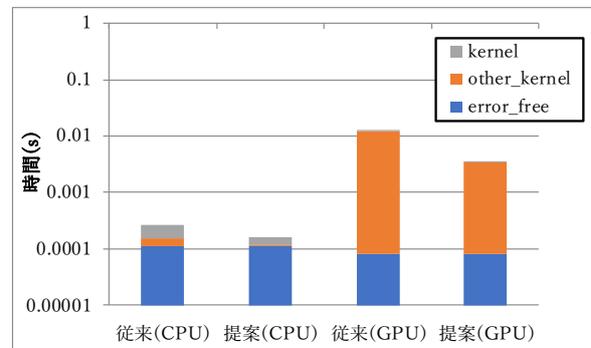


図 1 N=50 での CPU, GPU 環境のルーチン全体の演算時間

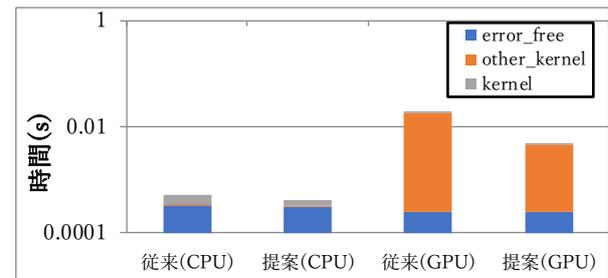


図 2 N=100 での CPU, GPU 環境のルーチン全体の演算時間

図 1、図 2 から、開発した高性能実装方式を用いた場合、行列サイズが N=50 の時 (図 1) は CPU 環境で最大 58.9%まで、GPU 環境ではサイズが N=100 の時 (図 2) に、従来の dgemm を用いた実装に対して最大で 25.7%まで実行時間が短縮されることを確認した。

次に演算性能の観点での性能評価を行う。図 3 に、CPU 環境および GPU 環境における演算性能 (GFLOPS 値) を示す。

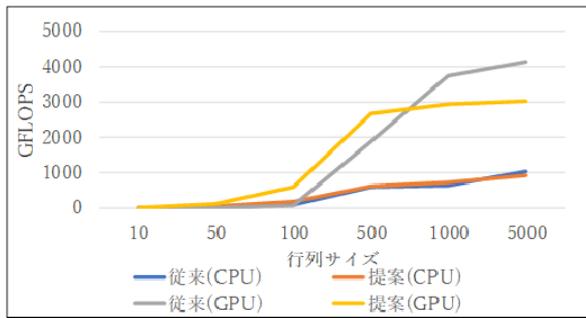


図3 CPU および GPU 環境での演算性能

図3から、行列サイズが大きくなると演算性能が向上することがわかる。また、行列サイズが大きいときは、提案手法 (batched BLAS による実装) の演算性能は従来手法に比べ性能が劣化した。N=5000 のとき従来手法では、CPU 環境、GPU 環境ともに、ハードウェアの理論性能に対して、8 割ほどの演算性能が出ている。そのため、従来の BLAS を利用した尾崎の方法、提案する実装方式の Batched BLAS を用いた尾崎の方法ともに高性能実装であることを示している。

4 運用構想

本プロジェクトで開発する精度保証付き数値計算ライブラリは、従来から大型計算機センター／情報基盤センターで行ってきた数値計算ライブラリ開発と運用のフレームワークが利用できる。しかしながら従来のフレームワークに対し、近年では、オープンソースの普及やスーパーコンピュータセンター群で資源提供する HPCI (High Performance Computing Infrastructure)での共用という新しい潮流が定着している。

そこで、以下の方針のもと、運用中のスーパーコンピュータを用いた精度保証付き数値計算ライブラリの普及と運用を図っていく。

1. スーパーコンピュータ仕様を含めることによるプリインストール：

名古屋大学情報基盤センターの次期システム (数値計算／データサイエンス融合型スパコン) の仕様に VNC-HPC ライブラリを含めることで、名古屋大学のスーパーコンピュータのユーザ、および、共同利用・共同研究拠点の事業である JHPCN 共

同研究課題の研究者に広く公開し、普及を促す。

2. オープンソースによるユーザの拡大：

2章で紹介した VNC-HPC ライブラリは、プロジェクトの HP において、MIT ライセンスでソースコードが公開されている。

3. HPCI 登録ソフトウェアによる HPCI ユーザへの普及：

HPCI では、HPCI システムで利用可能なソフトウェアを HP で公開している (http://www.hpci-office.jp/pages/project_categories_hardware?tab=software)。本プロジェクトの開発ソフトウェアも、HPCI で公開を予定している。

4. お試し講習会の実施による普及：

名古屋大学情報基盤センターが現在行っているシステム利用型講習会 (無料) における VNC-HPC ライブラリ利用講習会を開催する。特に、次期システムを利用した講習会開催を計画している。

5 おわりに

本報告では、精度保証付き数値計算ライブラリの開発の現状と、将来稼働するスーパーコンピュータを想定した運用について説明した。

謝辞

本研究は、平成 28 年度採択課題、ポスト「京」萌芽的課題「極限の探究に資する精度保証付き数値計算学の展開と超高性能計算環境の創成」、および、学際大規模情報基盤共同利用・共同研究拠点の支援による(課題番号: jh190015-NAJ)。

参考文献

- [1] 荻田武史、大石進一、大規模連立一次方程式のための高速精度保証法、情報処理学会論文誌：数理モデル化と応用、Vol.46, SIG10 (TOM12)、pp.10-18、2005.
- [2] T. Ogita, S. M. Rump, S. Oishi, Accurate sum and dot product, SIAM J. Sci. Comput., Vol. 26, No. 6, pp.1955-1988, 2005.
- [3] S. M. Rump, T. Ogita, S. Oishi, Accurate floating-point summation part I: Faithful rounding, SIAM J. Sci. Comput., Vol. 31, No.1, pp. 189-224, 2008.
- [4] Y. Kobayashi, T. Ogita, Accurate and efficient

- algorithm for solving ill-conditioned linear systems by preconditioning methods, *Nonlinear Theory and Its Applications*, IEICE, Vol.7, No.3, pp.374-385, 2016.
- [5] Y. Kobayashi, T. Ogita, K. Ozaki, Acceleration of a preconditioning method for ill-conditioned dense linear systems by use of a BLAS-based method, *Reliable Computing*, Vol. 25, pp.15-23, 2017.
- [6] T. Ogita, S. M. Rump, S. Oishi: Verified solution of linear systems without directed rounding, Technical Report, 2005-04, Advanced Research Institute for Science and Engineering, Waseda University, Tokyo, Japan, 2005.
- [7] 片桐孝洋、尾崎克久、荻田武史、大石進一、高精度行列-行列積アルゴリズムの疎行列演算化による高速化、日本応用数理学会「行列・固有値問題の解法とその応用」研究部会第15回研究会, SWoPP2013, 2013.
- [8] Shuntaro Ichimura, Takahiro Katagiri, Katsuhisa Ozaki, Takeshi Ogita, Toru Nagai, Threaded Accurate Matrix-Matrix Multiplications with Sparse Matrix-Vector Multiplications, The Thirteenth International Workshop on Automatic Performance Tuning (iWAPT2018) (In Conjunction with the IEEE IPDPS2018), Proceedings of IPDPSW2018, pp.1093-1102, 2018. DOI: 10.1109/IPDPSW.2018.00168