複素基本線形計算と多項式評価計算を最適化した多倍長精度演算ライブラリ BNCmatmul 0.22

BNCmatmul 0.22: The multiple-precision numerical computation library optimizing complex BLAS and evaluation of polynomial

幸谷 智紀 (Tomonori Kouya)[1]
[1] 追手門学院大学 (Otemon Gakuin University)
e-mail : t-koya@haruka.otemon.ac.jp

1　初めに

我々は 2000 年代初頭より binary64 を超える仮数部長をサポートした多倍長精度数値計算ライブラリ BNCpack を開発し，当初は GNU MP(GMP) の任意精度浮動小数点演算 (MPF)，ほどなく IEEE754 規格に基づく丸めモードや初等関数をサポートした厳格な任意精度浮動小数点演算をサポートした MPFR を用いて，線形計算，多項式演算，数値積分，常微分方程式の初期値問題に対応した関数を開発してきた。その後，マルチコンポーネント方式の固定精度浮動小数点演算もサポートした基本線形計算機能を最適化した BNCmatmul を開発した。

本講演ではこの BNCmatmul について，その概要を解説しつつ，新たに実装した複素基本線形計算 (complex BLAS) と多項式評価計算の最適化効果について述べる。また次の開発目標についても，現在のトレンドと計算環境に基づいた知見を述べる。

2　BNCmatmul 概要

現在のデジタル社会における主力になりつつある AI が必要とする浮動小数点演算は，IEEE754 binary32 や binary16 といった，短い仮数部長を持つ浮動小数点数が主として使用される．一方，科学技術演算においては binary64 以上の仮数部長の浮動小数点数が主として用いられている．ハードウェアの進化は市場のニーズに左右されるものであり，前者の性能向上を図ることが多くなっている．そのため，binary64 以上の仮数部長を必要とする悪条件問題においては，尾崎スキームに代表されるように，短い精度と長い精度の浮動小数点演算を組み合わせた混合精度計算を行い，パフォーマンスを挙げる取り組みが盛んである．

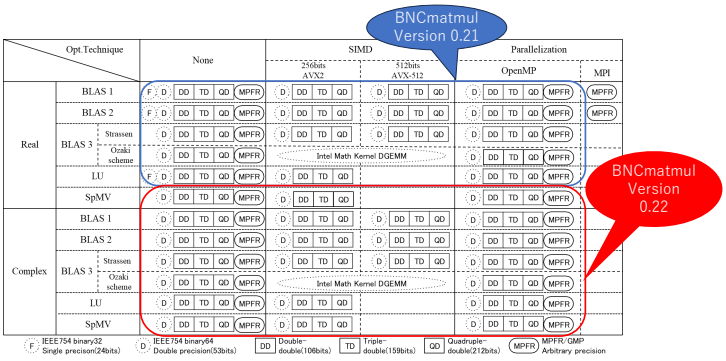

図 1. Current development status of BNCmatmul

我々は主として悪条件問題に対応するために，binary64 以上の精度を持つ多倍長精度浮動小数点演算をサポートした最適化基本線形計算ライブラリ BNCmatmul を開発してきた．現在の並列化を

前提としたアーキテクチャに適した最適化については C++ コンパイラが提供する物以上のものではない．マシン最適化された Intel Math Kernel[2] や OpenBLAS[3] が存在するように，最適化された多倍長精度 BLAS が必要である．

我々の BNCmatmul は主として x86 環境に最適化した BLAS 機能をサポートしたものであり，2025 年 3 月現在，MPBLAS の主機能やサポート外の TD(Triple-Double) 精度演算やランダム疎行列・ベクトル乗算 (SpMV) も実数・複素数も含めて実装を完了した．多項式評価についても，複素係数・実係数どちらについても Estrin 法に基づいた AVX2 による高速化を行っており，同時反復法の高速化を達成することができている [4].

3　複素線形計算の最適化とその応用

前述したように，我々は既に実基本線計算と LU 分解を AVX2 をはじめとする SIMD 演算と，OpenMP による並列化をサポートした多倍長精度線形計算ライブラリを実装済みである。MPLAPACK より高速な DD，QD 精度と TD 精度をサポートしたマルチコンポーネント型固定精度演算と，MPFR をベースとする任意精度浮動小数点演算をサポートしている．これをベースに複素線形計算を高速化した結果については既に報告している [5]．また任意精度 LU 分解についても成果をまとめて公開している [1]。

4　多項式評価関数の最適化とその応用

Estrin 法は多項式関数 $p_n(x) = \sum_{i=0}^{n} a_i x^i$ の値を求めるための方法で，演算量は標準的な Horner 法と同じであるが，並列化が容易に行えることから，C++, Julia 等，様々な環境下で実装が行われている。我々の BNCmatmul でも AVX2 による最適化を行った多項式関数の高速評価実装を行っており，その効果は例えば Durand-Kerner 法のような同時反復解法の高速化に寄与することが確認されている [4]

5　今後の課題

128-bit SIMD 最適化や Web 上の高速高精度計算，LLM の活用による開発効率の向上，等を視野に入れたものにしたいと考えている。

謝辞

本研究は科研費 23K11127 の助成を受けて実施されたものである．

参考文献

[1] Tomonori Kouya. Acceleration of complex matrix multiplication using arbitrary precision floating-point arithmetic. In *2023 International Conference on Engineering and Emerging Technologies (ICEET)*, pp. 1–6, 2023.

[2] Intel Math Kernel Library. `http://www.intel.com/software/products/mkl/`.

[3] OpenBLAS. `http://www.openblas.net/`.

[4] 幸谷智紀. 多倍長精度代数方程式ソルバー高速化の試み. Technical report, Mar 2025.

[5] 幸谷智紀．マルチコンポーネント型多倍長精度複素行列乗算の実装と性能評価．Technical Report 190, 研究報告ハイパフォーマンスコンピューティング (HPC), july 2023.

Cerebras CS-2 における疑似倍精度行列積

Pseudo double-precision matrix multiplication on Cerebras CS-2

松崎 竜之介 (Ryunosuke Matsuzaki)[1] 椋木 大地 (Daichi Mukunoki)[2]
宮島 敬明 (Takaaki Miyajima)[1]
[1] 明治大学理工学研究科情報科学専攻 (Computer Science Course, Meiji University)
[2] 名古屋大学 情報基盤センター (Information Technology Center, Nagoya University)
e-mail : ce245011@meiji.ac.jp, takaaki.miyajima@cs.meiji.ac.jp
e-mail : mukunoki@cc.nagoya-u.ac.jp

1 Cerebras CS-2 への疑似倍精度ライブラリの実装

Cerebras CS-2 は、世界最大の巨大チップである WSE-2（Wafer-Scale Engine 2）を搭載した、ディープラーニング専用のネットワークアタッチドアクセラレータである。本アーキテクチャにおける処理要素（Processing Element: PE）の総数は 853,104（792 × 1078）であり、総 SRAM 容量は 40GB、メモリ帯域幅は 20PB/s、動作周波数は 850MHz で構成されている。本システムにおける PE は完全に非同期に動作し、メモリの一貫性も保証されていない。そのため、CS-2 は本質的に分散メモリ型アーキテクチャとして捉えることができる。各 PE は、演算処理を担う Computing Element（CE）と、PE 間の通信を担う Router から構成される。CE には 48KB のスクラッチパッドメモリと SIMD 型の演算器が搭載されており、メモリ階層は 1 段のみである。

高精度浮動小数点演算を実現する手法の一つとして、ある仮数部長の浮動小数点演算を用いてその 2 倍長の仮数部の浮動小数点演算を実現する手法がある [1]。この手法では 2 倍長仮数部の浮動小数点数数が基本となる 2 つの浮動小数点数のペア（和）として表現される。例えば倍精度（double）型を 2 つ用いた double-double 演算や 4 つ用いた quad-double 演算を実装した QD ライブラリが知られている。本研究では、QD ライブラリに着想を得て、単精度浮動小数点数のみを用いて倍精度相当の精度を実現する擬似倍精度数値型（df_real）を Cerebras CS-2 上で実装する。具体的には、1 つの倍精度浮動小数点数を、2 つの単精度浮動小数点数 a_{hi}, a_{lo} に分解して表現する形式を DF（double-float）型と呼び、この形式に基づく加算および乗算を実装した数値ライブラリ（double-float library）を構築した。加算および乗算の演算アルゴリズムには、QD ライブラリで用いられている Sloppy アルゴリズムを採用した。特に乗算アルゴリズムにおいては、Fused Multiply-Add（FMA）命令を活用しており、これは Cerebras System Language（CSL）において組み込み関数@fmacs として提供されている。本実装により、DF 型による疑似倍精度積和演算 $a \times b + c$ は、16 回の単精度浮動小数点演算命令によって実現可能である。

2 疑似倍精度 GEMM の実装

GEMM は、深層学習や科学技術計算で活用されている最も一般的な基本線形代数サブプログラム（BLAS）の 1 つである。double-float libruary を用いて、CS-2 上で GEMM の実装を行った。CSL のビルトイン関数を使っておらず、ナイーブの実装となっているため、3 重 for 文で書かれている。最適化をすることができれば、2 重 for 文に抑えることが可能である。並列分散環境において GEMM を高効率かつスケーラブルに計算する Scalable Universal Matrix Multiplication Algorithm(SUMMA) がある。SUMMA は、大まかに積と和の計算 (FMA) と x,y 方向の 2 つの

Broadcast から成る。WSE-2 のような 2 次元メッシュトポロジの計算機には適した通信パターンである。通信レイテンシが大きい点も、WSE-2 の低い通信レイテンシが有効である。

3　評価

本実験では、Cerebras CS-2 上において、疑似倍精度 GEMM の強スケーリングおよび弱スケーリング性能を評価した。単体 PE が持てる計算要素数を Mt とし、総 PE が持つ計算要素数を M とする。強スケーリングの計測では、全体問題サイズを M ∈ 1000,2000,4000,8000,16000 に固定しつつ、使用する PE 数を 50 × 50 から 750 × 750 まで、50 刻みで段階的に増加させて性能を測定した。弱スケーリングにおいては、各 PE が保持する行列サイズ Mt=27 に固定し、使用する PE 数を 50 × 50 から 600 × 600 まで、全体の問題サイズを比例的に拡大しながら計測を行った。なお、どちらの評価においても計算、通信、制御を含んだサイクル数で計測を行っている。使用する PE の数によっては M が PE 数で割り切れないことがある。図 1 は、強スケーリングの計算性能の推移図、図 2 は、弱スケーリングの計算性能の推移図になる。図における TDFFlops/s(Tera Double Float Flops) は疑似倍精度演算 (df_mul や df_add)1 回を 1TDFFlop と数える。強スケーリングにおいては最大で 8.1TFDFlops/s を達成した。SGEMM 最大性能の 1/30 の性能が見られる [2]。ワークロードが大きいため、並列化による通信オーバーヘッドが相対的に小さく、計算効率が高い。PE 数が増加しても性能のばらつきが小さく、グラフが滑らかに伸びていることから、CS-2 における低レイテンシ通信の恩恵が顕著に表れている。弱スケーリングは、処理要素数の増加に対してサイクル数がほぼ線形に増加しており、性能は上昇している。これは演算負荷が適切に分散されていること、すなわちハードウェアアーキテクチャが高い並列効率を実現していることを強く示している。

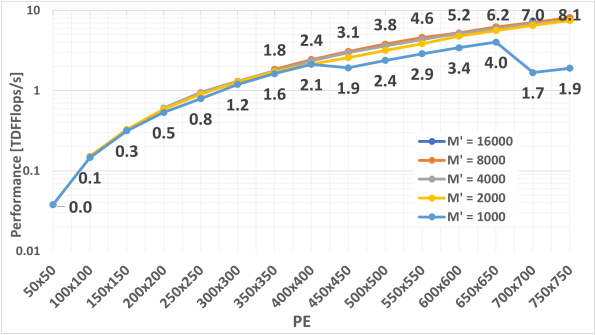

図 1. 強スケーリング

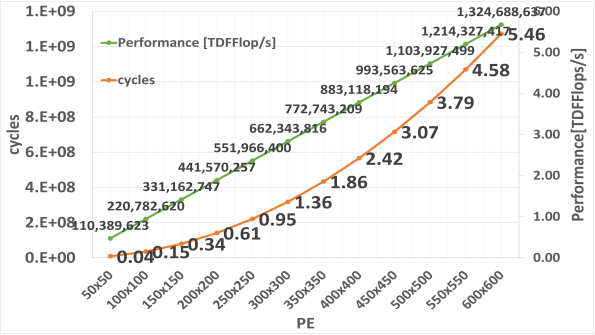

図 2. 弱スケーリング

4　まとめ

本稿では、疑似倍精度 GEMM の性能評価を試みた。最大性能は 8.1TDFFlops/s で、強/弱スケーリング共に PE が増えるにつれて性能は上昇した。全体サイズが大きいほど性能の向上が見られた。

参考文献

[1] T. J. DEKKER*, A Floating-Point Technique for Extending the Available Precision, Springer-Verlag, t8, 224-242, July 26, 1971

[2] 松崎 竜之介, 福岡 玲音, 宮島 敬明, Ryunosuke M, Reon F, Takaaki M, Cerebras CS-2 における単精度 GEMM の性能評価, 情報処理学会ハイパフォーマンスコンピューティング研究発表会, 第 193 回, 6-7, 2023

AI向け低精度演算の動向と尾崎スキームによるFP64演算への適用可能性の検討

Recent Trends of Low-precision Arithmetic for AI and its Potential for FP64 Computation with Ozaki Scheme

椋木 大地 (MUKUNOKI Daichi)[1], 林 俊一郎 (HAYASHI Shun-ichiro)[2],
[1] 名古屋大学情報基盤センター (Information Technology Center, Nagoya University),
[2] 名古屋大学大学院情報学研究科 (Graduate School of Informatics, Nagoya University)
e-mail : mukunoki@cc.nagoya-u.ac.jp

1 概要

AI技術の急激な発展と需要の拡大により，AI計算が必要とする低精度演算の性能を強化したプロセッサが開発されている．AI計算では大きなモデルをメモリに載せるためデータ表現の最適化が研究され，16ビット，8ビット，4ビット等の小型のデータ型の利用が考案された．そしてAI向けプロセッサではこれらを効率的に計算するための演算器の実装も進んだ．

低精度行列積を用いて高精度行列積を行う尾崎スキーム [1] は，行列の各要素の指数部レンジに応じて行列を要素の仮数部方向にいくつかの行列にスライスし，各スライスの全対全の行列積を計算し，最後にそれらを足し込むことで高精度行列積を計算する．指数部と仮数部を分離して計算することで，指数部長・仮数部長の短い低精度演算を計算に利用できる．例えばNVIDIA GPUが搭載するAI計算向けの行列演算器テンソルコアのFP16とFP32の混合精度演算を利用して，FP64等の行列積を計算することができる [2]．さらに2018年頃よりAI計算に8ビット等の整数演算が用いられるようになると，それらの演算性能を強化したハードウェアが登場し，尾崎スキームにおいても整数演算を利用する方法が開発された [3].

AI技術とそれに応じたハードウェアの変化は急速であり，これらを他用途において活用するにあたっては，その技術動向を押さえることが重要である．本研究では最新のAI向け演算の動向と，それらの尾崎スキームへの適用可能性について検討を行う．

2 AI向け低精度演算の動向

本稿執筆時点でAI向けプロセッサの演算性能に関する最も注目すべき点は，これまで性能向上が続いてきた8ビット整数演算（INT8）の性能が削られ，代わりに8ビット浮動小数点型（FP8）や4ビット型（FP4）の演算性能が強化されつつあることである．一例としてNVIDIA社のGPUを参照すると，現世代のGB200 NVL72システムではFP8とINT8がそれぞれ20 PFlops/s，20 POps/sである．しかし次世代のGB300 NVL72の予定仕様ではFP8およびFP6が720 PFlops/s，FP4は1400 PFlops/sである一方で，INT8の性能は23 POps/sにとどまっている．なおこれらはAI計算における疎性を考慮した性能値であり，一般の密演算では性能は低下する．他社のAI向けプロセッサにおいてもINT8サポートが削られている例が見られる．このような背景にはAI技術の革新による演算要求の変化がある．また新しい浮動小数点表現として，複数要素にまたがって指数部を共通化するブロックフォーマットや，指数部の圧縮を行うDFloat11などが提案されている．将来的にはこのような型がハードウェアサポートされる可能性も考えられる．さらに各演算に対する性能比についても，ハードウェア設計上の効率だけでなく，AI側の需要により今後変化する可能性がある．

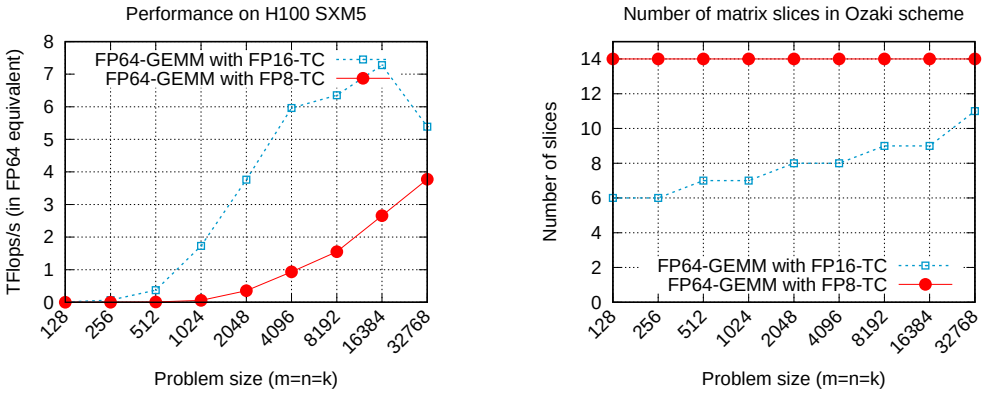

図 1. FP8 および FP16 テンソルコア（FP8-TC，FP16-TC）を用いた尾崎スキームによる FP64 行列積（FP64-GEMM）の性能（左）と，計算における行列スライス数（右）

3　尾崎スキームによる FP8 を用いた FP64 行列積

INT8 に代わり性能が強化されつつある FP8 を用いて尾崎スキームによる FP64 行列積を実装し性能を評価した．FP8 には E5M2 と E4M3 という 2 つのフォーマットがあり，E と M の後ろの数字はそれぞれ指数部，仮数部のビット長を示す．尾崎スキームでは仮数部のみを計算に用いるため，1 ビット仮数部の長い E4M3 を用いた．テンソルコアで FP8 型を計算する際には乗算は FP16 で行われ，その結果の蓄積は FP32 で行われる．

NVIDIA H100 SXM5 GPU において FP8 および FP16 を用いた尾崎スキームの性能を評価した．図 1 に FP64 換算での Flops/s 性能と行列スライス数（行列 A と B は等しい）を示す．行列スライス数の 2 乗回の行列積がテンソルコアで計算される．入力は 1 から 10 の範囲の乱数である．テンソルコアの理論ピーク演算性能は FP8 が 1979 TFlops/s，FP16 が 989 TFlops/s，FP64 が 33.5 TFlops/s である．FP8 を用いる場合の性能は FP16 の場合に比べ劣っているが，FP16 では問題サイズに応じてスライス数が増加するのに対して FP8 は一定である．これは FP8 テンソルコアが FP32 で蓄積を行うため 1 スライスが保持できる仮数部長が変化しないためである．

謝辞　本研究は学際大規模情報基盤共同利用・共同研究拠点（JHPCN）および革新的ハイパフォーマンス・コンピューティング・インフラ（HPCI）の支援による（課題番号:jh250018）．

参考文献

[1] K. Ozaki, T. Ogita, S. Oishi, S. M. Rump, Error-free transformations of matrix multiplication by using fast routines of matrix multiplication and its applications. Numer. Algorithms 59, 1 (2012), pp. 95–118.

[2] D. Mukunoki, K. Ozaki, T. Ogita, T. Imamura, DGEMM using Tensor Cores, and Its Accurate and Reproducible Versions. Proc. ISC2020, LNCS 12151 (2020). pp. 230–248.

[3] H. Ootomo, K. Ozaki, R. Yokota, DGEMM on integer matrix multiplication unit, Int. J. High Perform. Comput. Appl. 38, 4 (2024), pp. 297-313.

