



柴田 直樹 (NAIST)

概要: 三角関数等を計算する標準ライブラリ(Libm)のSIMD版を
英国ARM社と連携して開発, 複数の商用製品・OSSで採用

背景

- SIMD命令が一般に使われるプロセッサで利用可能になった. (Intel SSE2, AVX2, AVX512, ARM NEON, SVE 等)
- コンパイラに, 自動ベクトル化機能(SIMD命令を利用しないコードをSIMD対応に変換)が搭載されるように
- 上記に伴い, 標準ライブラリ(特にlibm)もSIMD対応が求められるようになってきた

開発したライブラリの特徴

- C99規格におけるmath.hで定義された浮動小数点の関数全てをベクトル化して実装
 - 1ULP と 3.5ULP の精度を選択可能
- Intel の SVML に匹敵する性能 (精度, 計算速度の両方) を実現
- 移植性が良く, 違った命令セットへの対応が容易であり, なおかつ性能が良い
 - アーキテクチャ・命令セット独自の機能を利用可能
- SIMDによる処理に適した実装, アルゴリズムを (一部) 新規開発
 - 全パスをベクトル化: 条件分岐, 可変アドレスへのメモリアクセスを最小限に
 - ベクトル長が変化しても計算ステップ数は同じ
- 決定的演算モードを実装: 利用する命令セットやベクトル長が異なっても計算結果が完全に一致

新規開発のアルゴリズム - 三角関数の引数の Reduction(引数から関数値が同じ $0 \dots \pi/2$ までの値を求める)

- 既存の方法 1: Cody-Waite 高速だが, 対応できる引数の範囲が狭い
- 既存の方法 2: Payne-Hanek 対応できる引数の範囲が非常に広いが, 低速, テーブル読みだしが必要
- 提案手法: 完全にベクトル化された Payne-Hanek

新規開発のアルゴリズム - 浮動小数点数の剰余

- 既存の方法では内部的に整数演算を行っており, 原理的に $O(\log(\text{除数} / \text{被除数}))$ の計算時間が必要
- 提案手法では, 浮動小数点演算を利用し, SIMDに適した非常に高速なアルゴリズムを考案

下記の商用製品・OSSに採用

- Arm HPC Compiler <https://developer.arm.com/>
- Unity (Burst Compiler) <https://unity.com/>
- PyTorch <https://pytorch.org>
- Portable Computing Language <http://portablecl.org>

発表論文等

- Naoki Shibata(NAIST) and Francesco Petrogalli(Arm): "SLEEF: A Portable Vectorized Library of C Standard Mathematical Functions," IEEE Transactions on Parallel and Distributed Systems. DOI: 10.1109/TPDS.2019.2960333

