

# AVX2 を用いた倍々精度演算の反復解法への適用と評価

菱沼 利彰<sup>1</sup>

<sup>1</sup> 筑波大学

e-mail : hishinuma@slis.tsukuba.ac.jp

## 1 はじめに

物理シミュレーションの核である Krylov 部分空間法は，丸め誤差の影響により収束に影響を受ける．収束の改善には高精度演算が有効だが，高精度演算は計算コストが高い [1]．

高精度演算をする手法の一つに，倍精度変数を 2 つ用いて 4 倍精度演算を実行する倍々精度演算という手法がある [2]．

我々は，入力行列を倍精度，ベクトルを倍々精度とした疎行列ベクトル積 (DD-SpMV) や，その転置演算 (DD-TSpMV) を対象に，Intel の SIMD 拡張命令である Advanced vector extensions2 (AVX2)[3] を用いて高速化している．

AVX2 を用いた CRS 形式 [4] の DD-SpMV や DD-TSpMV では端数の処理などが性能劣化要因となる．

疎行列の格納形式の 1 つである Block CRS (BCRS) 形式 [4] は，疎行列を零要素を含む  $r \times c$  の小密行列 (ブロック) の集合として格納する形式で，ブロックサイズを SIMD のレジスタサイズに合わせることで上記の CRS 形式の性能劣化要因を無くすることができる [5]．

これまで，SIMD 化の効果に着目して評価を行ってきた．本論文では，実際にアプリケーションとして利用することを想定し，AVX2 を用いた BCRS 形式の DD-SpMV や DD-TSpMV を反復解法に適用し，倍精度と比較することで，その高速化効果を示す．

## 2 AVX2 を用いた倍々精度演算

実際の反復解法ライブラリにおいて多くの場合，入力行列は倍精度で与えられ，反復計算中に値は変更されない．そのため，倍精度の疎行列と倍々精度のベクトルの積を行った．これにより，データサイズを約半分にし，メモリへのデータ要求を減らすことができる．DD-SpMV の核である倍精度と倍々精度の積和演算の演算量は 19 flops である．

AVX2 は 1 回の演算で必ず 4 つの演算を実行しなければならないため，CRS 形式では各行で端数の処理が必要になる．また，ベクトル  $x$  へ

表 1. 対象問題

|                  | N      | nnz       |
|------------------|--------|-----------|
| bcsstk39         | 46,772 | 2,060,662 |
| TSOPF_FS_b162_c4 | 40,798 | 2,398,220 |
| nasasrb          | 54,840 | 2,677,324 |

のアクセスがインデックスを用いた間接参照になる．これらは，性能を低下させる要因となる．

ブロックサイズを SIMD のレジスタサイズに合わせた BCRS4x1 形式は，CRS 形式の問題をなくし，メモリアクセスをスムーズにすることで，SIMD 化による高速化効果を効率的に引き出すことができるが，ブロックに零要素を含むため，演算量が最大で 4 倍に増加する．BCRS4x1 を用いた実装は，文献 [5] を参照されたい．

## 3 数値実験

実験の対象として，双共役勾配法 (BiCG 法) を用いた．BiCG 法は，SpMV1 回，TSpMV1 回，内積などのベクトル演算を 12 回含む．

対象とした問題を表 1 に示す．BCRS 化による効果が文献 [5] の中で平均的な 3 問である．それぞれ順に，BCRS4x1 形式にすることで演算量が 1.6, 1.4, 1.1 倍に増加する．

使用した CPU は Intel core i7 4770 K 3.4GHz 4core 16 GB，OS は CentOS6.4 で，コンパイラは Intel C/C++ Compiler 13.0.1，コンパイラオプションは “-O3 -xCORE-AVX2 -openmp -fp-model precise” を用い，実験は 4 スレッドで行った．

倍精度 (CRS 形式)，倍々精度 (CRS 形式)，倍々精度 (BCRS4x1 形式) のそれぞれにおける BiCG 法 100 反復の実行時間を図 1 に示す．

倍々精度 (CRS 形式) は倍精度と比べて，

- 全体の実行時間は 1.9 から 2.7 倍，
- ベクトル演算は 2.0 倍，
- DD-SpMV は 2.1 から 2.8 倍，
- DD-TSpMV は 2.1 から 3.0 倍

の時間がかかる．また，倍々精度 (BCRS4x1 形式) は倍精度と比べて，

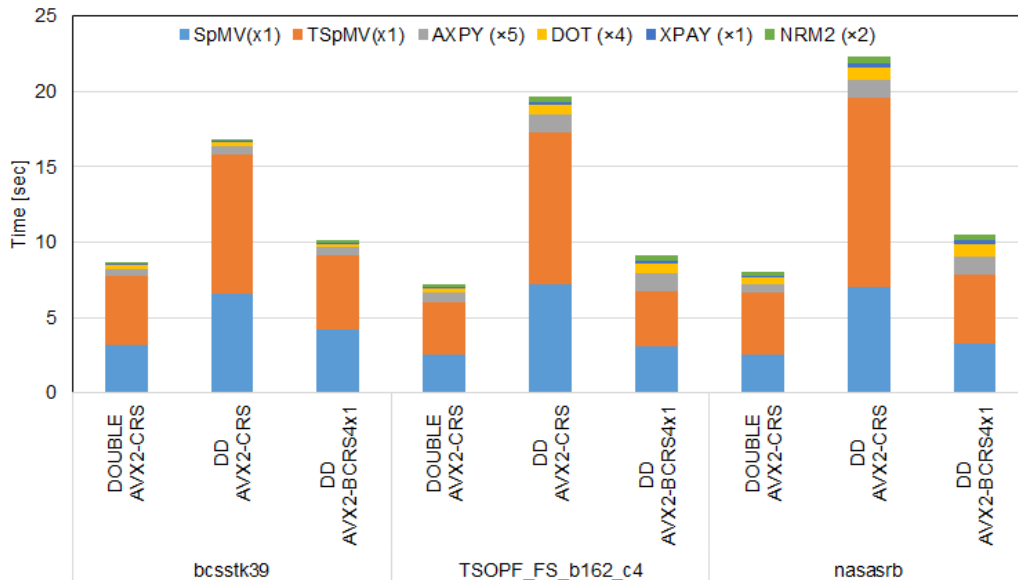


図 1. BiCG 法 100 反復における実行時間 [sec.]

- 全体の実行時間は 1.2 から 1.3 倍，
- ベクトル演算は 2.0 倍，
- DD-SpMV は 1.1 から 1.2 倍，
- DD-TSpMV は 1.05 から 1.1 倍

の時間がかかる．

倍々精度では，BCRS4x1 を用いることで，倍精度と比べて DD-SpMV や DD-TSpMV の性能差は 1.05 から 1.2 倍になり，これは小さい．BCRS4x1 形式を用いたことで SIMD 化の効果を効率的に引き出せていることがわかる．

それに対し全体の実行時間が最大で 1.3 倍かかるのは，倍精度は行列計算が全体の 90% を占めているが，倍々精度は行列計算が全体の 73 から 84% となるためである．ベクトル演算はすべて倍々精度で行うため，メモリ性能に制約を受けて 2 倍の時間がかかるが，DD-SpMV や DD-TSpMV は疎行列を倍精度にすることで，反復計算全体では，実行時間を倍精度の 2 倍以下にできる．

#### 4 結論

AVX2 を用いた BCRS 形式の DD-SpMV や DD-TSpMV を反復解法に対し，同一反復回数における計算時間を倍精度と比較した．

BCRS4x1 を用いて SIMD 命令を効率的に利用することの効果は高く，DD-SpMV や DD-TSpMV は倍精度と比べて 1.05 から 1.2 倍で実行できている．しかし，ベクトル演算が倍精度と比べて 2 倍の時間がかかるため，全体の時間

は 1.2 から 1.3 倍となった．

今後の課題として，より多くの応用例に対して，どのような場合で，どの程度の収束改善効果が得られれば，倍々精度演算を用いることで倍精度と比べて優位になるかについて，検討していく必要がある．

#### 参考文献

- [1] Tomonori Kouya.: A Highly Efficient Implementation of Multiple Precision Sparse Matrix-Vector Multiplication and Its Application to Product-type Krylov Subspace Methods, IJNMA, Vol. 7, Issue 2, pp. 107-119, 2012.
- [2] Bailey, D ,H.: High-Precision Floating-Point Arithmetic in Scientific Computation, computing in Science and Engineering, pp.54-61, 2005.
- [3] Intel: Intrinsics Guide, <http://software.intel.com/en-us/articles/intel-intrinsics-guide>
- [4] Barrett, R., et al.: Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, SIAM pp. 57-65 (1994)
- [5] T. Hishinuma, et. al.: SIMD Parallel Sparse Matrix-Vector and Transposed-Matrix-Vector Multiplication in DD Precision, VECPAR2016, pp.1-8, 2016.