5G-03

高精度計算ライブラリを利用した 多倍長精度行列ベクトル演算の性能評価

関谷　和明†　　荻野　正雄‡　　片桐　孝洋‡　　永井　亨‡

名古屋大学工学部†　　名古屋大学情報基盤センター‡

科学技術計算の分野では,主に 64 ビット構成の倍精度浮動小数点数が使われている.しかし,近年では計算の質の向上のために,それ以上の高精度な計算に対する需要も高まってきている.そのため,固定長の整数や固定精度の倍精度浮動小数点数を用いて多倍長精度浮動小数点数演算を実現する数値計算ライブラリが開発され,普及しつつある.さらに,4 倍精度演算については,GCC 4.6 以降の__float128 型によるソフトウェアサポートや IBM Power9 によるハードウェアサポートなど利用しやすい環境が提供されている.そこで本研究では,GCC __float128 型,倍精度演算ベースである QD ライブラリ,整数演算ベースである exflib ライブラリなどを用いて,基本線形代数サブルーチン BLAS における Level 1 相当のベクトル演算や Level 2 相当の行列ベクトル演算などを実装し,その性能の評価を行った.

1. はじめに

多倍長精度の演算は,反復解法などでの有用性が認められるものの,倍精度演算と比べると演算時間が多くかかる.そのため,より効率的に多倍長精度演算を行うために,多くの多倍長精度演算アルゴリズムや数値計算ライブラリが開発されている.そのアルゴリズムには,固定精度の倍精度浮動小数点数を複数用いて多倍長精度数を実現する方法と, 固定長の整数を用いる方法が存在する.また近年では, IBM Power 9 による 4 倍精度演算のハードウェアサポートも実現されており,数値計算に関するさまざまな分野で活用されている.

本稿では,まず多倍長精度演算の実装について述べ,次にスーパーコンピュータ等を用いて実施した数値実験の結果とその評価について述べる.

2. 多倍長精度演算の実装

2.1 擬似四倍精度

擬似四倍精度とは,Bailey や Briggs によって提案された,2 つの倍精度浮動小数点数を用いて四倍精度に近い精度を実現する手法である.擬似四倍精度の値は,2 つの倍精度浮動小数点数の和によって表される.その精度は仮数部が 2×53=106 ビットで,IEEE754 で規定されている真の四倍精度数の 113 ビットと比べるとわずかに少ない.

2.2 QD ライブラリ

QD ライブラリ[1,2]では,前述の擬似四倍精度演算および 4 つの倍精度浮動小数点数で構成される擬似八倍精度の演算をサポートしている.擬似四倍精度数を dd_real 型,擬似八倍精度数を qd_real 型として定義する.四則演算だけでなく,平方根や初等関数等も実装されている.複素数演算は C++の complex クラスを用いて std::complex<dd_real>などとすれば良い.

2.3 exflib

exflib[3]は,C++および Fortran90/95 に対応している任意精度計算ライブラリである.ある整数 N を指定することで 10 進 N 桁での演算を行うことができる.ただし厳密には, ユーザが指定した精度とライブラリ内で実際に行われる演算の精度は異なる.

2.4 その他の四倍精度演算サポート

GCC の Ver.4.6 以降では四倍精度浮動小数点数に関する quadmath ライブラリが追加されている.四倍精度の実数を__float128 型,複素数を__complex128 型として定義しており,この精度に対する四則演算ほか種々の関数も提供されている.

Intel C/C++でも独自に四倍精度演算機能が提供されている.四倍精度浮動小数点数を_Quad 型として定義している.ただし,入出力については,実装が不完全で十分なサポートがなされていないことに注意が必要である.

3. 数値実験

3.1 実験環境

倍々精度の QD ライブラリ, 任意精度の exflib, 及び 4 倍精度の quadmath ライブラリについて演算性能を比較するために, Intel Core i7-4770 プロセッサで構成される計算機(PC)を用いた. QD ライブラリと quadmath ライブラリには gcc-5.4.0 コンパイラ, exflib には Visual C++ 2015 コンパイラを用いた. QD ライブラリは qd-2.3.17,exflib は exflib-win32-dll-20160808 を使用した.

また, 実行環境の違いによる QD ライブラリの性能比較と Intel _Quad 型の性能評価を行うために, 名古屋大学情報基盤センターのスーパーコンピュ

Performance Evaluation of Multiple Precision Matrix-Vector Multiplication Using High Precision Arithmetic Library
† Kazuaki Sekiya / Nagoya University
‡ Masao Ogino, Takahiro Katagiri, Toru Nagai / Information Technology Center, Nagoya University

ータ[4]である，Fujitsu PRIMEHPC FX100 と Fujitsu PRIMERGY CX400 を用いた．FX100 では，Fujitsu C/C++ Ver.2.0.0 コンパイラ，CX400 では Fujitsu C/C++ Ver.1.2.0 コンパイラと Intel C++ Ver.14.0.3.174 コンパイラが利用可能である．

3.2 問題設定

2 章に記したライブラリおよび四倍精度演算の機能を用いて,実数および複素数について行列ベクトル積の計算を行い,その時間の計測を行った.問題設定として，行列およびベクトルの次数は 4,000 とし，その値は乱数により生成した．また，コンパイラ最適化オプションは"-O2"を指定した．

3.3 実行結果

PC を用いた QD，exflib，並びに quadmath の演算性能比較結果を表 1-2 に示す．表において，double は倍精度演算，pseudo quad は擬似四倍精度演算，quad は 128 ビットの四倍精度演算，pseudo oct は擬似八倍精度演算を表す．ここで，便宜上 QD のところに double の結果を示しているが，この計算にはライブラリを使用していない．また，gcc が GCC コンパイラ，cl は Visual C++コンパイラを利用したことを表す．表 1-2 より,固定精度の数値計算ライブラリではより高い精度を求めようとすると計算時間も多くかかってしまうことが分かる.また,QD ライブラリの擬似四倍精度と GCC の提供している四倍精度とを比べた場合,GCC の提供している四倍精度の方が,精度の面でも速度の面でも良いと言える.さらに,八倍に近い精度になると固定精度の計算ライブラリよりも整数演算ベースである任意精度の計算ライブラリの方が有効であることも分かる.

表 1 PC での実数行列ベクトル積時間[sec]

Library/ Compiler	double	pseudo quad	quad	pseudo oct
QD/gcc	0.04466	0.72863	-	6.7336
exflib/cl	2.2222	2.2262	-	3.1544
quadmath/gcc	-	-	0.65523	-

表 2 PC での複素数行列ベクトル積時間[sec]

Library/ Compiler	double	pseudo quad	quad	pseudo oct
QD/gcc	0.33443	3.1555	-	26.233
exflib/cl	15.124	15.142	-	19.856
quadmath/gcc	-	-	2.0041	-

次に，スパコン上で実行した場合の結果を表 3-4 に示す．表において，FX は FX100，CX は CX400，fcc は Fujistu コンパイラ，icc は Intel コンパイラ，_Quad は Intel _Quad の四倍精度演算を表す．表 3-4 より,同じマシン上で QD ライブラリを富士通コンパイラと Intel コンパイラを用いてコンパイルする場合,Intel コンパイラの方が 3〜4 倍速く,相性が良いと言える.しかし,Intel の提供する四倍精度は,QD ライブラリの擬似四倍精度に比べて計算時間が多くかかり,効率が良いとは言えない.

表 3 スパコンでの実数行列ベクトル積時間[sec]

Computer/ Library/ Compiler	double	pseudo quad	quad	pseudo oct
FX/QD/fcc	0.056	0.31867	-	1.6667
CX/QD/fcc	0.032667	0.50933	-	5.3143
CX/QD/icc	0.010333	0.16767	-	1.314
CX/_Quad/icc	-	-	0.54667	-

表 4 スパコンでの複素数行列ベクトル積時間[sec]

Computer/ Library/ Compiler	double	pseudo quad	quad	pseudo oct
FX/QD/fcc	0.078667	1.2887	-	7.179
CX/QD/fcc	0.139	2.4077	-	21.459
CX/QD/icc	0.038667	1.4597	-	4.5167
CX/_Quad/icc	-	-	2.9407	-

4. まとめ

複数のマシンで多倍長精度の計算を行ったが,やはり倍精度での計算と比べるとその計算時間は多くなる.実用性を高めるためには，SIMD 演算への最適化などが必要になると言える．また,八倍精度に近い精度では,整数演算ベースのライブラリが有効であったが，まだ実用性は十分とは言えない.今後は,さらに大きな行列に対し,基本的な行列ベクトル演算だけでなく,疎行列ベクトル積演算並びにクリロフ部分空間法に適用し,その反復回数や計算時間を測定,評価していく予定である.

謝辞

本研究の一部は，JST-CREST「ポストペタスケールシミュレーションのための階層分割型数値解法ライブラリ開発」の支援を受けたものである.

参考文献

[1] Hida, Y., Li, X.S., Bailey, D.H., “Double-Double(DD)および Quad-Double(QD)演算ライブラリ,” https://na-inet.jp/na/qd_ja.pdf
[2] High-Precision Software Directory http://crd-legacy.lbl.gov/~dhbailey/mpdist/
[3] exflib - extend precision floating-point arithmetic library http://www-an.acs.i.kyoto-u.ac.jp/~fujiwara/exflib/
[4]名古屋大学情報基盤センタースーパーコンピュータシステム http://www.icts.nagoya-u.ac.jp/ja/sc/