

QPBLAS-GPU 利用マニュアル

目次

1. プログラム概要.....	1
2. ディレクトリ構成.....	2
3. コンパイル方法.....	3
3.1. GDDBLAS ライブラリ.....	3
3.2. 動作確認プログラム.....	4
3.3. 演算時間測定プログラム.....	5
4. 使用方法.....	6
4.1. GDDBLAS ライブラリ.....	6
4.2. 動作確認プログラム.....	6
4.3. 演算時間測定プログラム.....	7
5. サブルーチン一覧.....	7

1. プログラム概要

原子力研究開発機構システム計算科学センターシミュレーション技術開発室では、第二期中期計画に定めた研究開発テーマ「高効率な熱電材料、電源材料及び熱伝導材料の構造と機能の関係解明」を達成するため、関連する計算手法の研究開発を行ってきた。特に、一つの主たる課題である超伝導体材料の超伝導発現機構を調べるため、量子力学(量子多体問題)に基づき正確に電子状態を求めるための計算手法開発を進めている。

一般に、量子多体問題を正確に解くためには、極めて大きな自由度を近似なしに扱う必要性から、大規模シミュレーションを行う必要がある。現在、このような大規模シミュレーションは並列計算機を利用することで原理的に実現可能であり、プロセッサ毎の利用可能メモリを集めれば、全体として巨大なメモリ領域が確保され、大規模な超並列シミュレーションが実行可能である。実際にこれを実施し、正確な量子状態を得るためには、膨大な数の演算を実施する必要がある(例えば、大規模行列の対角化を行い、固有値、固有ベクトルを得るためには、行列サイズの増大とともに計算実行時利用 CPU の数も増え、超並列計算となる)。この時、計算機上ではある有効桁数での演算を行うため、1回の演算ごとに丸め誤差が混入し、演算量が増加するほど累積誤差が増大する。従来規模のシミュレーションではその誤差が問題視されることはなかったが、京コンピュータのような超大規模計算機の性能を極限まで利用したシミュレーションを行うと、累積誤差は膨大になり、シミュレーション結果の有効桁がほとんどなくなる可能性があることを指摘してきた経緯がある(実際、前地球シミュレータの全ノード 4096CPU を用いて計算可能な固有値問題を解いた際、有効精度が数桁程度になることを観測している)。

このような状況の下、計算機シミュレーションにおいて頻繁に利用される基本演算ルーチン群である BLAS の 4 倍精度への拡張は、高精度な超大規模並列シミュレーションにおいては必要不可欠な研究開発項目であると同時に、すべての大規模シミュレーションに共通な普遍性を有するため優先度の高い研究開発項目である。機構では平成 21 年度「原子力デバイスシミュレーション用高精度計算ルーチンの開発」において基礎となる 16 個のルーチンについて 4 倍精度化を実施し、平成 22 年度「原子力材料シミュレーション用高精度計算ルーチンの開発」では残りの 24 ルーチンについて 4 倍精度化、加えて OpenMP による並列化を実施した(この 2 つの開発をまとめて「4 倍精度化 BLAS ルーチン」という)。さらに近年、GPU の演算性能の向上により、これを用いた大規模数値シミュレーション例が数多く報告されている。今後、GPU を搭載した並列クラスタによる大規模並列シミュレーションの増大が見込まれる。

そこで今年度は平成 21,22 年度に行った「4 倍精度化 BLAS ルーチン」を GPU 上で動作させるための一連の作業を行った。本作業で作成した GPU 対応ライブラリを GDDBLAS と呼ぶことにする。

2. ディレクトリ構成

GDDBLAS および関連するファイルを格納したディレクトリは GDDBLAS_V0.0 である。このディレクトリの構成を図 2.1 に示す。また、表 2.1 にそれぞれのディレクトリ及びファイルの説明を示す。

GDDBLAS_V0.0
├ Makefile
├ TESTING/
├ gddblas/
├ include/
├ lib/
├ make.inc
├ speedCheck/
├ src/
└ src_multi/

図 2.1 GDDBLAS_RELEASE のディレクトリ構成

表 2.1 ディレクトリおよびファイルの説明

ディレクトリ/ファイル名	説明
Makefile	ライブラリ作成用の Makefile
TESTING	動作確認計算用のソースファイルを格納
gddblas	GDDBLAS のソースファイルを格納
include	種々のインクルードファイルを格納
lib	ライブラリファイル libgddblas.a を格納 リリース時は empty
make.inc	コンパイル時の環境設定ファイル
speedCheck	演算時間測定用のソースファイルを格納
src	ホストメモリに対して GDDBLAS を使用する 場合のラッパー関数を格納(単数 GPU)
src_multi	ホストメモリに対して GDDBLAS を使用する 場合のラッパー関数を格納(複数 GPU)

3. コンパイル方法

3.1. GDDBLAS ライブラリ

GDDBLAS のコンパイル方法について説明する。GDDBLAS のコンパイルには CUDA コンパイラ及び CUDA 標準ライブラリが必要である。また、複数 GPU を使用する場合には OpenMP が必要である。make.inc の内容を図 3.1 に示す。make.inc の各変数の説明を表 3.1 に示す。GDDBLAS をコンパイルするには make.inc を使用する環境に合わせて書き換える必要がある。

図 3.1 make.inc の内容

```
CC = gcc
NVCC = nvcc
INCLUDES = -I/usr/local/cuda40/cuda/include
INCLUDES += -I/usr/local/cuda40/cuda/NVIDIA_GPU_Computing_SDK/C/common/inc
INCLUDES += -I./include
NVOPTS = -arch sm_20 --compiler-options -Wall -Xcompiler -fopenmp
CCOPTS = -Wall -O3 -fopenmp
.SUFFIXES: .cu
USEMULTI = yes
```

表 3.1 make.inc 内の各変数の説明

ディレクトリ/ファイル名	説明
CC	C 言語用のコンパイラ
NVCC	CUDA 言語用のコンパイラ
INCLUDE	CUDA 標準ライブラリの include ファイル 及び本作業により作成した include ファイル
NVOPTS	NVCC 言語コンパイル用のオプション
CCOPTS	C 言語コンパイル用のオプション
.SUFFIXES	.cu ファイルをサフィックスに追加
USEMULTI	複数 GPU の関数をライブラリに含めるかどうかのフラグ。yes の場合には OpenMP が必要。

コンパイル変数の修正が終わったら GDDBLAS_v0.0 の下に移動して、

```
make
```

として、make を実行すれば、GDDBLAS ライブラリ libdgdblas.a が生成される。

3.2. 動作確認プログラム

動作確認プログラムのコンパイルには倍精度の BLAS ライブラリを必要とする。これは動作確認プログラムでは、4 倍精度の GDDBLAS の結果と倍精度の BLAS の結果を比較するためである。

動作確認計算プログラムをコンパイルする際、倍精度の BLAS とリンクするように図 3.2 の TESTING/Makefile のライブラリのパスを書き換える必要がある。

図 3.2 動作確認関数用の Makefile の内容

```
include ../make.inc
CC=nvcc
CP = /bin/cp
LIBS =
LIBS += -lcudart -lcublas -lcutil_x86_64 -L../lib -lgddblas -lgfortran
LIBS += -L../lib -lblas -lg2c
CFLAGS=$(INCLUDES)
CFLAGS+= -Xcompiler -fopenmp
STRIP = strip
SOURCE=$(shell ls *.c)
.....
```

コンパイル変数の修正が終わったら GDDBLAS_v0.0/TESTING の下に移動して、

```
make
```

として、`make` を実行すれば、各関数に対する動作確認用の実行ファイルが生成される。

複数 GPU 用の実行ファイルは、GDDBLAS_v0.0/TESTING/ /common_test.h に以下の行を追記することで生成される。

```
#define MULTI
```

3.3. 演算時間測定プログラム

動作確認プログラムでは CPU と GPU の 4 倍精度演算時間を比較するため、コンパイルには 4 倍精度の BLAS ライブラリを必要とする。

動作確認計算プログラムをコンパイルする際、4 倍精度化 DDBLAS ライブラリとリンクするように図 3.2 の speedCheck/Makefile のライブラリのパスを書き換える必要がある。

図 3.3 動作確認関数用の Makefile の内容

```
CC = nvcc
CP = /bin/cp
INCLUDES = -I. -I.
INCLUDES+=-I/usr/local/cuda40/cuda/include
INCLUDES+= -I/usr/local/cuda40/cuda/NVIDIA_GPU_Computing_SDK/C/common/inc
LIBS = -L../lib -lblas -lg2c -lddbblas_21 -lgddbblas
LIBS+=-L/usr/local/cuda40/cuda/lib64 -lcudart -lcublas
LIBS+=-L/usr/local/cuda40/cuda/NVIDIA_GPU_Computing_SDK/C/lib -lcutil_x86_64
CCOPTS = -Wall -O3 -fopenmp
CFLAGS=$(INCLUDES)
CFLAGS+= -fopenmp
NVOPTS = -arch sm_20 -maxrregcount 32
STRIP = strip
FC = gfortran
CC = gcc
.....
```

コンパイル変数の修正が終わったら GDDBLAS_v0.0/speedCheck の下に移動して、

```
make
```

として、make を実行すれば、各関数に対する演算時間測定用の関数が生成される。

4. 使用方法

GDDBLAS ライブラリ及び動作確認計算プログラムの使用方法について説明する

4.1. GDDBLAS ライブラリ

GDDBLAS ライブラリとのリンク時には CUDA 標準ライブラリとのリンクを必要とする。

3.1 節に従い、lib/libgddblas.a が生成されたら、

```
gfortran -O3 -o foo foo.f -L./DDBLAS_RELEASE/ lib -lddbblas  
-L/usr/local/cuda40/cuda/lib64 -lcudart -lcublas  
-L/usr/local/cuda40/cuda/NVIDIA_GPU_Computing_SDK/C/lib -lcutil_x86_64
```

のようにコンパイルを実行すれば、GDDBLAS のルーチンを用いた実行ファイルをリンクすることができる。各ルーチンの呼び出し方や変数の説明に関しては、詳細設計書の各項目を参照。

4.2. 動作確認プログラム

3.2 節に従ってコンパイルを行うことで各関数名に対応する動作確認プログラムが作成される。gemm の動作確認計算を行うには GDDBLAS_v0.0/TESTING のディレクトリの下で

```
./gemm_test
```

とする。動作確認計算を実行すると、行列の行数、列数、転置の情報などが表示され、動作確認に通過した場合には Success の文字列が出力される。以下に動作確認計算の出力例を示す。

```
dgemm C(52,162) += A(150,52)^t * B(150,162)^N  
calling single-gpu function  
calling single-cpu function  
check  
Success
```

4.3. 演算時間測定プログラム

3.3 節に従ってコンパイルを行うことで各関数名に対応する演算時間測定プログラムが作成される。gemm の演算時間測定計算を行うには GDDBLAS_v0.0/speedCheck のディレクトリの下で

```
./speedcheck_gemm
```

とする。演算時間測定計算を実行すると、行列のサイズ、CPU の計算時間、GPU の計算時間の情報などが表示される。また、表示と同様の内容が演算時間測定プログラムと同じ名前で拡張子が csv のファイルに出力される。以下に演算時間測定計算の csv ファイル出力例を示す。

matsize,	CPU,	GPU
4,	5.00679E-06,	0.000860929
8,	2.88486E-05,	0.000853062
16,	0.000210047,	0.001013994
32,	0.00161314,	0.001042843
64,	0.012376785,	0.001155138
128,	0.097104073,	0.001870871
256,	0.770068884,	0.00423789
512,	6.14212108,	0.01710391
1024,	49.406682,	0.100292921

ここで、1 列目が行列のデータサイズ、2 列目が CPU の計算時間(秒)、3 列目が GPU の計算時間(秒)を示す。

5. サブルーチン一覧

Level	Subroutines	演算内容
Level 1	gddswap	$x \leftrightarrow y$
	gddscal	$x \rightarrow \alpha x$
	gddcopy	$y = x$
	gddaxpy	$y \rightarrow \alpha x + y$
	gdddot	$x^T y$

	gddnrm2	$\sqrt{\sum x_i^2}$
	gddasum	$\sum x_i $
	gddidamax	絶対値が最大の値を持つ配列の番号
	gddrot	Givens rotatio
	gddrotg	Matrix elements of Givens rotation
	gddrotm	Modified Givens rotatio
	gddrotmg	Matrix elements of modified Givens rotation
	gddzdotc	$\bar{x}^T y$
	gddzdotu	$x^T y$
Level 2	gddgemv	$y \rightarrow \alpha Ax + \beta y, y \rightarrow \alpha A^T x + \beta y$
	gddsylv	Symmetric matrix, $y = \alpha Ax + \beta y$
	gddtrmv	$x \rightarrow Ax, x \rightarrow A^T x$
	gddtrsv	上下三角行列に A に対し、 $Ax = b, A^T x = b$ を解く
	gddsyr	$A \rightarrow \alpha x x^T + A$
	gddsyr2	$A \rightarrow \alpha x y^T + \alpha y x^T + A$
	gddgbmv	General band matrix, $y \rightarrow \alpha Ax + \beta y, y \rightarrow \alpha A^T x + \beta y$
	gddger	$A \rightarrow \alpha x y^T + A$
	gddsmbv	General band matrix, $y \rightarrow \alpha Ax + \beta y$
	gddtbmv	三角帯行列用の $x \rightarrow Ax, x \rightarrow A^T x$
	gddtbsv	三角帯行列に A に対し、 $Ax = b, A^T x = b$ を解く
	gddzgerc	$A \rightarrow \alpha x \bar{y}^T + A$
	gddgzeru	$\alpha AB + \gamma C, \alpha A^T B + \gamma C, \alpha AB^T + \gamma C, \alpha A^T B^T + \gamma C$
	gddzhbmv	Hermite abnd matrix, $y \rightarrow \alpha Ax + \beta y$
	gddzhemv	Hermite matrix, $y \rightarrow \alpha Ax + \beta y$
	gddzher	Hermite matrix, $A \rightarrow \alpha x \bar{x}^T + A$
	gddzher2	Hermite matrix, $A \rightarrow \alpha x \bar{y}^T + \bar{\alpha} y \bar{x}^T + A$
Level 3	gddgemm	$\alpha AB + \gamma C, \alpha A^T B + \gamma C, \alpha AB^T + \gamma C, \alpha A^T B^T + \gamma C$
	gddsymm	Symmetric matrix, $\alpha AB + \gamma C, \alpha BA + \gamma C$
	gddsyr2k	$\alpha AB^T + \alpha BA^T + \gamma C, \alpha A^T B + \alpha B^T A + \gamma C$
	gddsyrk	Symmetric matrix, $C \rightarrow \alpha AA^T + \beta C, C \rightarrow \alpha A^T A + \beta C$

	gddtrmm	三角行列用の $B \rightarrow \alpha \text{op}(A)B, B \rightarrow \alpha B \text{op}(A)$
	gddtrsm	上下三角行列に A に対し、 $Ax = b, A^T x = b$ を解く
	gddzhemm	Hermite matrix, $C \rightarrow \alpha AB + \beta C, C \rightarrow \alpha BA + \beta C$
	gddzher2k	Hermite matrix, $C \rightarrow \alpha A\bar{B}^T + \bar{\alpha} B\bar{A}^T + \beta C, C \rightarrow \alpha \bar{A}^T B + \bar{\alpha} \bar{B}^T A + \beta C$
	gddzherk	Hermite matrix, $C \rightarrow \alpha A\bar{A}^T + \beta C, C \rightarrow \alpha \bar{A}^T A + \beta C$