

**GPU 用 4 倍精度  
BLAS ルーチン  
QPBLAS-GPU Ver.1.0  
ユーザーズマニュアル**

2013 年 7 月  
独立行政法人  
日本原子力研究開発機構



## 目次

1	概説.....	3
2	BLAS について.....	4
3	4 倍精度化アルゴリズムについて.....	4
4	参考文献.....	4
5	ディレクトリ構成.....	5
6	インストール方法.....	5
7	検証用・性能評価用プログラム.....	6
8	サンプルプログラム.....	7
9	サブルーチン詳細.....	10

# 1 概説

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「原子炉構造材料における劣化現象の解明、燃料関連アクチノイド化合物の物質特性の予測並びに高効率な熱電材料、電源材料及び超伝導材料の構造と機能の関係解明のための高精度シミュレーション技術を開発する」を達成するため、関係する計算手法の研究開発を行っている。上記計画では、材料物性を研究する数値計算手法の開発が共通の計算科学上の課題であり、量子力学（量子多体問題）に基づき正確に電子状態を計算する手法の開発が不可欠である。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があることから、大規模な数値計算を、精度を維持しながら実施しなければならない。現在、量子多体問題を解くような大規模計算は、並列計算機を利用することで実現可能であり、プロセッサ毎の利用可能メモリを集めることで全体として巨大なメモリ領域が確保されれば、計算資源が許す限りの大規模問題にアプローチできる。しかし、実際に大規模計算を行い、正確な量子状態を得るためには、膨大な数の演算が必要となるため、累積誤差が蓄積する事が知られてきた（計算機では、ある有効桁数での演算を行うため、一回の演算毎に丸め誤差が混入し、演算量が増加するほどその累積誤差は増大する）。

これまで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を実現するにあたり必要不可欠な研究開発項目であり、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。

当該研究室では平成 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、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ことから、平成 23 年度には「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開発作業を行った。この GPU 対応ライブラリを

QPBLAS-GPU と呼ぶ。以下ではこのライブラリの使い方の説明をする。

## 2 BLAS について

Basic Linear Algebra Subprograms (BLAS) とは、線形代数の分野において頻繁に必要な基本的な演算をまとめた高品質な数値計算ライブラリである。1979年に FORTRAN によって書かれたコードが発表されたが、その後、C/C++を含む各種プログラミング言語へのバインディングが行われたため、現在は多くの言語において BLAS に含まれる演算をほぼ共通のインターフェースで利用できる。なお、演算の追加や並列計算への最適化といった面で BLAS の拡張も行われてきた。これらの拡張された数値計算ライブラリとしては、LAPACK, ATLAS, Intel Kernel Math Library (MKL) などが知られている。

BLAS に含まれる演算は、その種類によって 3 つのレベルに分類されている。レベル 1 はベクトルとベクトルの演算、レベル 2 は行列とベクトルの演算、レベル 3 では行列と行列の演算である。

## 3 4 倍精度化アルゴリズムについて

IEEE 754 規格に基づく 4 倍精度(128 bit)の浮動小数点演算は、最新の Fortran を含む一部の言語ではすでに実装されている。しかし、ハードウェアがこの演算に対してまだ最適化されていないため、この演算は任意精度演算として実行される場合が多い。その結果、演算速度が倍精度演算と比べて非常に遅くなるという課題を抱えている。

QPBLAS では、計算精度と計算速度を両立させるために、D.H.Bailey が提案した 2 つの倍精度データを用いて 1 つの 4 倍精度データを表現する double-double アルゴリズムを採用した。この double-double アルゴリズムを用いた数値表現の範囲や精度は、指数部分のデータを 2 重に持つことになるため本来の 4 倍精度のそれよりは若干劣るものの、通常の倍精度数値表現と比べると約 2 倍になる。このような高精度な演算を倍精度演算の組み合わせのみで実現できるため、大規模で高精度なシミュレーションの実現のために非常に有効なアルゴリズムである。

## 4 参考文献

BaileyHD. (日付不明). High-Precision Software Directory. 参照先：  
<http://crd-legacy.lbl.gov/~dhbailey/mpdist>

山田進, 佐々成正, 今村俊幸, 町田昌彦. (2012). 4 倍精度基本線形代数ルーチン QPBLAS の紹介とアプリケーションへの応用. 2012-HPC, 137 (23), 1-6.

## 5 ディレクトリ構成

圧縮されたソースコードを展開すると、以下のようなディレクトリ構成になる。

├── cpu	ホストメモリに対するラッパー関数
├── cpu_multi	ホストメモリに対するラッパー関数（複数CPU）
├── gpu	double-double4倍精度化GPU対応BLAS
├── sample	サンプルプログラム
└── test	動作検証・性能評価プログラム

## 6 インストール方法

QPBLAS-GPU の検証用プログラムでは QPBLAS に含まれる 128bit4 倍精度化 BLAS を利用している。4 倍精度化 BLAS との比較を行う場合はあらかじめライブラリを作成しておくこと。

本ライブラリのインストールには Ver.4.0 の CUDA コンパイラ及び CUDA 標準ライブラリが必要である。

以下のコマンドを端末に入力することで、ユーザーの環境に合わせた **makefile** の作成とコンパイルが行われる。

```
tar xzf QPBLAS-GPU-version.tar.gz
cd QPBLAS-GPU-version
./configure options
make
```

**configure** の際に考慮すべき主なオプションとその意味は、以下の通りである。

<code>FC=f90_compiler</code>	Fortran コンパイラの指定（サンプルのビルドのため）
<code>FCFLAGS=options</code>	コンパイルオプションの指定並列計算や最適化に関する指定を行う。
<code>LDFLAGS=options</code>	リンクオプションの指定
<code>LIBS=options</code>	追加ライブラリの指定
<code>--prefix=path</code>	インストールディレクトリの変更

<code>--with-cuda=path</code>	CUDA がインストールされているディレクトリを指定する
<code>--without-dblas</code>	倍精度 BLAS ライブラリをコンパイルするかどうかの指定 指定された場合、テストプログラムの一部で用いる BLAS の倍精度ルーチンをコンパイルせずに、 <i>LDFLAGS</i> などによって指定された既存のライブラリを使用する。 デフォルトでは、倍精度 BLAS ライブラリをコンパイルする。

コンパイルのために最低限必要な `configure` オプションは、たとえば次のようになる。

例：GNU Fortran を利用する場合

```
./configure FC=gfortran LDFLAGS="-L../lib" ¥
LIBS="-lqblas -ldblas" --with-cuda=/usr/local/cuda-4.0
```

`make` が成功すると、次のライブラリやプログラムが作成される。

<code>libgddblas.a</code>	QPBLAS-GPU ライブラリ
<code>test/level1</code>	Level 1 の検証用プログラム
<code>test/level2</code>	Level 2 の検証用プログラム
<code>test/level3</code>	Level 3 の検証用プログラム

`libddblas.a` を既定の場所にインストールする場合は、次のコマンドを端末に入力する。ほかのライブラリやプログラムはインストールされない。

```
make install
```

## 7 検証用・性能評価用プログラム

6 に示した手順によってコンパイルを完了させたのち、次のコマンドを入力する。ただし、`n` は 1, 2, 3 のいずれかである。

```
cd test
./leveln
```

すると、QPBLAS-GPU に含まれるそのレベルのルーチンの一覧などを含むメニュー画面が表示される。このメニュー画面で一連の数字を入力することで、ライブラリの検証と性能評価を行うことができる。たとえば、level1 では以下のような命令を実行できる。

1 ... 14	該当するルーチンを QPBLAS, 倍精度 BLAS, 4 倍精度 BLAS で計算する (4 倍精度 BLAS は実行可能な場合のみ)
75	ベクトルの長さを変更する (一度 Enter を押し、新しい数値を聞かれたら入力して Enter を押す)
96	このレベルのすべてのルーチンを 1 度ずつ実行する (自動テスト用)
97	ルーチンの入出力の詳細表示の ON/OFF を切り替える (起動時は OFF)
98	計算結果の比較表示の ON/OFF を切り替える (起動時は OFF)
99	計算時間の結果表示の ON/OFF を切り替える (起動時は OFF)
0	プログラムを終了する

たとえば、次の数字を順に入力すると、長さ 100 のベクトルにおける GDDAXPY の計算結果の比較と計算時間の測定を行い、端末に出力する。(改行ごとに Enter キーを押す)

```
75
100
98
99
4
```

## 8 サンプルプログラム

sample ディレクトリに、Fortran 90 で書かれた QPBLAS-GPU のサンプルプログラムがいくつか用意されている。

gddnrm2_sample.f90	ベクトルの 2 ノルム
gddtrsv_sample.f90	三角行列を含む式を解く
gddgemm_sample.f90	一般行列と行列の演算

gzzdotu_sample.f90	複素数ベクトルの内積
gzzhemv_sample.f90	エルミート行列とベクトルの演算
gzzhemm_sample.f90	エルミート行列と行列の演算

これらのサンプルプログラムをコンパイルして、実行するには次のように入力する。

```
cd sample
make
./gzzdotu_sample
```

このプログラムを自分でコンパイルする場合は、たとえば次のように行う。

```
gfortran gzzdotu_sample.f90 ../libddblas.a -fopenmp ¥
-L/usr/local/cuda-4.0/lib64 ¥
-L/usr/local/cuda-4.0/NVIDIA_GPU_Computing_SDK/C/lib ¥
-lcudart -lcublas -lcuti1_x86_64
```

サンプルプログラムの内容は以下の通り。

### 1.1 ベクトルの2ノルム (gddnrm2\_sample.f90)

4倍精度実数の2000次のベクトル  $x(=x_h + x_l)$  として high word  $x_h$  を乱数で、low word  $x_l$  を 0.0d で与えて、その2-ノルム  $d(=d_h + d_l) = \sum x_i^2$  を計算する。

### 1.2 三角行列を解く (gddtrsv\_sample.f90)

4倍精度の行列  $A(=A_h + A_l)$  を

$$A_h = \begin{bmatrix} 1.0 & 0.1 & 0.2 \\ & 1.0 & 0.3 \\ & & 1.0 \end{bmatrix}$$

および  $A_l = 0$  で与えられる3次の上三角行列とし、4倍精度実数のベクトル  $b(=b_h + b_l)$  を high word を乱数で、low word を 0.0d で与えて、方程式  $Ax = b$  を解く。

### 1.3 一般行列と行列の演算 (gddgemm\_sample.f90)

4倍精度の2行4列の行列  $A(=A_h + A_l)$ 、4行3列の行列  $B(=B_h + B_l)$  および2行3列の行列  $C(=C_h + C_l)$  について、それぞれの high word を乱数で与え、low word を 0.0d で与える。係数  $\alpha$  および  $\beta$  を乱数で与えて、 $C = \alpha AB + \beta C$  を計算する。

#### 1.4 複素数ベクトルの内積 (gzzdotu\_sample.f90)

4 倍精度複素数の 2000 次のベクトル  $x(= x_h + x_l)$  と  $y(= y_h + y_l)$  について、それぞれ high word を乱数で、low word を 0.0d で与えて、その内積  $z(= z_h + z_l) = x^T y$  を計算する。

#### 1.5 エルミート行列とベクトルの演算 (gzzhemv\_sample.f90)

4 倍精度 3 次のエルミート行列  $A(= A_h + A_l)$ 、4 倍精度 3 次の複素数ベクトル  $x(= x_h + x_l)$  と  $y(= y_h + y_l)$  を high word を乱数で与え、low word を 0.0d で与える。係数  $\alpha$  および  $\beta$  を乱数で与えて、 $y = \alpha Ax + \beta y$  を計算する。

#### 1.6 エルミート行列と行列の演算 (gzzhemm\_sample.f90)

4 倍精度 3 次エルミート行列  $A(= A_h + A_l)$ 、3 行 2 次の複素数行列  $B(= B_h + B_l)$  および  $C(= C_h + C_l)$  について、high word を乱数で与え、low word を 0.0d で与える。係数  $\alpha$  および  $\beta$  を乱数で与えて、 $C = \alpha AB + \beta C$  を計算する。

#### 1.7 サンプルソースコード

サンプルプログラム gzzdotu\_sample.f90 のソースコードは以下の通り。

```
program main

  implicit none

  integer :: n
  double complex, allocatable :: xh(:), xl(:)
  integer :: incx
  double complex, allocatable :: yh(:), yl(:)
  integer :: incy
  double complex :: zh, zl

  n = 2000
  incx = 1
  incy = 1

  allocate(xh(1+(n-1)*abs(incx)))
  allocate(xl(1+(n-1)*abs(incx)))
  allocate(yh(1+(n-1)*abs(incy)))
```

```

allocate (yl (1+(n-1)*abs (incy)))
call random_complex(xh)
xl = 0d0
call random_complex(yh)
yl = 0d0

call gzdotu_host_multigpu(n, xh, xl, incx, yh, yl, incy, zh, zl)

print *, "zh =", zh
print *, "zl =", zl

contains

subroutine random_complex(z)

    double complex, intent(out) :: z(:)
    double precision, allocatable :: re(:), im(:)

    allocate(re(size(z,1)))
    allocate(im(size(z,1)))
    call random_number(re)
    call random_number(im)
    z(:) = dcplx(re, im)

end subroutine

end program

```

## 9 サブルーチン詳細

QPBLAS-GPU は、BLAS に含まれるルーチンのうち、とくによく用いられる 40 個のルーチンについて実装されている。

Lev	GDDSWAP	ベクトルの入れ替え	$y \leftrightarrow x$
-----	---------	-----------	-----------------------

el1	GDDCOPY	ベクトルのコピー	$y \rightarrow x$
	GDDSCAL	ベクトルのスケーリング	$y \rightarrow \alpha x$
	GDDAXPY	ベクトルの演算	$y \rightarrow \alpha x + y$
	GDDDOT	ベクトルの内積	$x^T y$
	GDDNRM2	ベクトルの2-ノルム	$\sum x_i^2$
	GDDASUM	ベクトルの和 (1-ノルム)	$\sum  x_i $
	GIDDAMAX	絶対値が最大の値を持つベクトルの番号	
	GDDRROTG	ギブンス回転行列の作成	
	GDDRROT	ギブンス回転の適用	
	GDDRROTMG	修正ギブンス回転行列の作成	
	GDDRROTM	修正ギブンス回転の適用	
	GZZDOTC	複素数ベクトルの内積 (第1ベクトルを共役転置)	$\bar{x}^T y$
	GZZDOTU	複素数ベクトルの内積 (第1ベクトルを転置)	$x^T y$
Lev el2	GDDGEMV	一般行列とベクトルの演算	$y \rightarrow \alpha Ax + \beta y$ $y \rightarrow \alpha A^T x + \beta y$
	GDDGBMV	一般帯行列とベクトルの演算	$y \rightarrow \alpha Ax + \beta y$ $y \rightarrow \alpha A^T x + \beta y$
	DDGER	一般行列のランク1の更新	$A \rightarrow \alpha xy^T + A$
	GDDSYMV	対称行列とベクトルの演算	$y \rightarrow \alpha Ax + \beta y$
	GDDSBMV	対称帯行列とベクトルの演算	$y \rightarrow \alpha Ax + \beta y$
	GDDSYR	対称行列のランク1の更新	$A \rightarrow \alpha xx^T + A$
	GDDSYR2	対称行列のランク2の	$A \rightarrow \alpha xy^T + \alpha yx^T + A$

		更新	
	GDDTRMV	三角行列とベクトルの演算	$x \rightarrow Ax$ $x \rightarrow A^T x$
	GDDTBMV	三角帯行列とベクトルの演算	$x \rightarrow Ax$ $x \rightarrow A^T x$
	GDDTRSV	三角行列を含む式を解く	$Ax = b$ $A^T x = b$
	GDDTBSV	三角帯行列を含む式を解く	$Ax = b$ $A^T x = b$
	GZZGERC	一般複素数行列のランク1の更新 (第2ベクトルを共役転置)	$A \rightarrow \alpha \bar{y}^T + A$
	GZZGERU	一般複素数行列のランク1の更新 (第2ベクトルを転置)	$A \rightarrow \alpha xy^T + A$
	GZZHEMV	エルミート行列とベクトルの演算	$y \rightarrow \alpha Ax + \beta y$
	GZZHBMV	エルミート帯行列とベクトルの演算	$y \rightarrow \alpha Ax + \beta y$
	GZZHER	エルミート行列のランク1の更新	$A \rightarrow \alpha x \bar{x}^T + A$
	GZZHER2	エルミート行列のランク2の更新	$A \rightarrow \alpha x \bar{y}^T + \bar{\alpha} y \bar{x}^T + A$
Level3	GDDGEMM	一般行列と行列の演算	$C \rightarrow \alpha AB + \gamma C$ $C \rightarrow \alpha A^T B + \gamma C$ $C \rightarrow \alpha AB^T + \gamma C$ $C \rightarrow \alpha A^T B^T + \gamma C$
	GDDSYMM	対称行列と行列の演算	$\alpha AB + \gamma C$ $\alpha BA + \gamma C$
	GDDSYRK	対称行列のランクkの更新	$C \rightarrow \alpha AA^T + \beta C$ $C \rightarrow \alpha A^T A + \beta C$
	GDDSYR2K	対称行列のランク2kの更新	$C \rightarrow \alpha AB^T + \alpha BA^T + \beta C$ $C \rightarrow \alpha A^T B + \alpha B^T A + \beta C$
	GDDTRMM	三角行列と行列の演算	$B \rightarrow \alpha A^* B$ $B \rightarrow \alpha BA^*$
	GDDTRSM	三角行列を含む式を解	$Ax = b$

		く	$A^T x = b$
	<b>GZZHEMM</b>	エルミート行列と行列の演算	$C \rightarrow \alpha AB + \beta C$ $C \rightarrow \alpha BA + \beta C$
	<b>GZZHERK</b>	エルミート行列のランクkの更新	$C \rightarrow \alpha A\bar{A}^T + \beta C$ $C \rightarrow \alpha \bar{A}^T A + \beta C$
	<b>GZZHER2K</b>	エルミート行列のランク 2k の更新	$C \rightarrow \alpha A\bar{B}^T + \bar{\alpha} B\bar{A}^T + \beta C$ $C \rightarrow \alpha \bar{A}^T B + \bar{\alpha} B^T A + \beta C$



# ***GDDSWAP***

概要：

ベクトルの入れ替え

引数	I/O	型	説明
N	i	integer	ベクトルの長さ
XH	i/o	double precision	4倍精度実数ベクトルのhigh word
XL	i/o	double precision	4倍精度実数ベクトルのlow word
INCX	i	integer	XH, XLの増分
YH	i/o	double precision	4倍精度実数ベクトルのhigh word
YL	i/o	double precision	4倍精度実数ベクトルのlow word
INCY	i	integer	YH, YLの増分

# ***GDDCOPY***

概要：

ベクトルのコピー

引数	I/O	型	説明
N	i	integer	ベクトルの長さ
XH	i	double precision	4倍精度実数ベクトルのhigh word
XL	i	double precision	4倍精度実数ベクトルのlow word
INCX	i	integer	XH, XLの増分
YH	o	double precision	4倍精度実数ベクトルのhigh word
YL	o	double precision	4倍精度実数ベクトルのlow word
INCY	i	integer	YH, YLの増分

# ***GDDSCAL***

概要：

ベクトルのスケーリング

$x := a * x$

引数	I/O	型	説明
N	i	integer	ベクトルの長さ
AH	i	double precision	4倍精度実数のhigh word
AL	i	double precision	4倍精度実数のlow word
XH	i/o	double precision	4倍精度実数ベクトルのhigh word
XL	i/o	double precision	4倍精度実数ベクトルのlow word
INCX	i	integer	XH, XLの増分

# ***GDDAXPY***

概要：

ベクトルの演算

$y := a * x + y$

引数	I/O	型	説明
N	i	integer	ベクトルの長さ
AH	i	double precision	4倍精度実数のhigh word
AL	i	double precision	4倍精度実数のlow word
XH	i	double precision	4倍精度実数ベクトルのhigh word
XL	i	double precision	4倍精度実数ベクトルのlow word
INCX	i	integer	XH, XLの増分
YH	i/o	double precision	4倍精度実数ベクトルのhigh word
YL	i/o	double precision	4倍精度実数ベクトルのlow word
INCY	i	integer	YH, YLの増分

# ***GDDDOT***

概要：

ベクトルの内積

引数	I/O	型	説明
N	i	integer	ベクトルの長さ
XH	i	double precision	4倍精度実数ベクトルのhigh word
XL	i	double precision	4倍精度実数ベクトルのlow word
INCX	i	integer	XH, XLの増分
YH	i	double precision	4倍精度実数ベクトルのhigh word
YL	i	double precision	4倍精度実数ベクトルのlow word
INCY	i	integer	YH, YLの増分
DDH	o	double precision	4倍精度実数のhigh word
DDL	o	double precision	4倍精度実数のlow word

## ***GDDNRM2***

概要：

ベクトルの2-ノルム

引数	I/O	型	説明
N	i	integer	ベクトルの長さ
XH	i	double precision	4倍精度実数ベクトルのhigh word
XL	i	double precision	4倍精度実数ベクトルのlow word
INCX	i	integer	XH, XLの増分
DDH	o	double precision	4倍精度実数のhigh word
DDL	o	double precision	4倍精度実数のlow word

# ***GDDASUM***

概要：

ベクトルの和 (1-ノルム)

引数	I/O	型	説明
N	i	integer	ベクトルの長さ
XH	i	double precision	4倍精度実数ベクトルのhigh word
XL	i	double precision	4倍精度実数ベクトルのlow word
INCX	i	integer	XH, XLの増分
DDH	o	double precision	4倍精度実数のhigh word
DDL	o	double precision	4倍精度実数のlow word

# ***GIDDAMAX***

概要：

絶対値が最大の値を持つベクトルの番号

引数	I/O	型	説明
<b>N</b>	i	integer	ベクトルの長さ
<b>XH</b>	i	double precision	4倍精度実数ベクトルのhigh word
<b>XL</b>	i	double precision	4倍精度実数ベクトルのlow word
<b>INCX</b>	i	integer	XH, XLの増分
<b>ret</b>	ret	integer	絶対値が最大の値を持つベクトルの番号

# ***GDDROTG***

概要：

ギブンス回転行列の作成

引数	I/O	型	説明
AH	i/o	double precision	4倍精度実数のhigh word
AL	i/o	double precision	4倍精度実数のlow word
BH	i/o	double precision	4倍精度実数のhigh word
BL	i/o	double precision	4倍精度実数のlow word
CH	o	double precision	生成されたギブンス回転データの余弦成分の high word
CL	o	double precision	生成されたギブンス回転データの余弦成分の low word
SH	o	double precision	生成されたギブンス回転データの正弦成分の high word
SL	o	double precision	生成されたギブンス回転データの正弦成分の low word

# ***GDDROT***

概要：

ギブンス回転の適用

引数	I/O	型	説明
N	i	integer	ベクトルの長さ
XH	i/o	double precision	4倍精度実数ベクトルのhigh word
XL	i/o	double precision	4倍精度実数ベクトルのlow word
INCX	i	integer	XH, XLの増分
YH	i/o	double precision	4倍精度実数ベクトルのhigh word
YL	i/o	double precision	4倍精度実数ベクトルのlow word
INCY	i	integer	YH, YLの増分
CH	i	double precision	適用するギブンス回転データの余弦成分の high word
CL	i	double precision	適用するギブンス回転データの余弦成分の low word
SH	i	double precision	適用するギブンス回転データの正弦成分の high word
SL	i	double precision	適用するギブンス回転データの正弦成分の low word

# ***GDDR0TMG***

概要：

修正ギブズ回転行列の作成

引数	I/O	型	説明
D1H	i/o	double precision	4倍精度実数のhigh word
D1L	i/o	double precision	4倍精度実数のlow word
D2H	i/o	double precision	4倍精度実数のhigh word
D2L	i/o	double precision	4倍精度実数のlow word
X1H	i/o	double precision	4倍精度実数のhigh word
X1L	i/o	double precision	4倍精度実数のlow word
Y1H	i	double precision	4倍精度実数のhigh word
Y1L	i	double precision	4倍精度実数のlow word
PARAMH	i/o	double precision	生成された修正ギブズ回転パラメータの high word
PARAML	i/o	double precision	生成された修正ギブズ回転パラメータの low word

# ***GDDROT***

概要：

修正ギブンス回転の適用

引数	I/O	型	説明
N	i	integer	ベクトルの長さ
XH	i/o	double precision	4倍精度実数ベクトルのhigh word
XL	i/o	double precision	4倍精度実数ベクトルのlow word
INCX	i	integer	XH, XLの増分
YH	i/o	double precision	4倍精度実数ベクトルのhigh word
YL	i/o	double precision	4倍精度実数ベクトルのlow word
INCY	i	integer	YH, YLの増分
PARAMH	i	double precision	適用する修正ギブンス回転パラメータのhigh word
PARAML	i	double precision	適用する修正ギブンス回転パラメータのlow word

# ***GZZDOTC***

概要：

複素数ベクトルの内積（第1ベクトルを共役転置）

引数	I/O	型	説明
N	i	integer	ベクトルの長さ
XH	i	double complex	4倍精度複素数ベクトルのhigh word
XL	i	double complex	4倍精度複素数ベクトルのlow word
INCX	i	integer	XH, XLの増分
YH	i	double complex	4倍精度複素数ベクトルのhigh word
YL	i	double complex	4倍精度複素数ベクトルのlow word
INCY	i	integer	YH, YLの増分
ZZH	o	double complex	4倍精度複素数のhigh word
ZZL	o	double complex	4倍精度複素数のlow word

# ***GZZDOTU***

概要：

複素数ベクトルの内積（第1ベクトルを転置）

引数	I/O	型	説明
N	i	integer	ベクトルの長さ
XH	i	double complex	4倍精度複素数ベクトルのhigh word
XL	i	double complex	4倍精度複素数ベクトルのlow word
INCX	i	integer	XH, XLの増分
YH	i	double complex	4倍精度複素数ベクトルのhigh word
YL	i	double complex	4倍精度複素数ベクトルのlow word
INCY	i	integer	YH, YLの増分
ZZH	o	double complex	4倍精度複素数のhigh word
ZZL	o	double complex	4倍精度複素数のlow word

# GDDGEMV

概要 :

一般行列とベクトルの演算

$y := \alpha * A * x + \beta * y$

or

$y := \alpha * A^T * x + \beta * y$

引数	I/O	型	説明
TRANS	i	character(1)	行列の転置操作の制御 TRANS = 'N' or 'n' → 転置しない TRANS = 'T' or 't' → 転置する TRANS = 'C' or 'c' → 共役転置する
M	i	integer	行列の行または列の数
N	i	integer	行列の行または列の数
ALPHAH	i	double precision	4倍精度実数のhigh word
ALPHAL	i	double precision	4倍精度実数のlow word
AH	i	double precision	4倍精度実数行列のhigh word
AL	i	double precision	4倍精度実数行列のlow word
LDA	i	integer	行列Aの第一次元
XH	i	double precision	4倍精度実数ベクトルのhigh word
XL	i	double precision	4倍精度実数ベクトルのlow word
INCX	i	integer	XH, XLの増分
BETAH	i	double precision	4倍精度実数のhigh word
BETAL	i	double precision	4倍精度実数のlow word
YH	i/o	double precision	4倍精度実数ベクトルのhigh word
YL	i/o	double precision	4倍精度実数ベクトルのlow word
INCY	i	integer	YH, YLの増分

# GDDGBMV

概要 :

一般帯行列とベクトルの演算

$y := \alpha * A * x + \beta * y$

or

$y := \alpha * A^T * x + \beta * y$

引数	I/O	型	説明
TRANS	i	character(1)	行列の転置操作の制御 TRANS = 'N' or 'n' → 転置しない TRANS = 'T' or 't' → 転置する TRANS = 'C' or 'c' → 共役転置する
M	i	integer	行列の行または列の数
N	i	integer	行列の行または列の数
KL	i	integer	下バンド幅
KU	i	integer	上バンド幅
ALPHAH	i	double precision	4倍精度実数のhigh word
ALPHAL	i	double precision	4倍精度実数のlow word
AH	i/o	double precision	4倍精度実数行列のhigh word
AL	i/o	double precision	4倍精度実数行列のlow word
LDA	i	integer	行列Aの第一次元
XH	i	double precision	4倍精度実数ベクトルのhigh word
XL	i	double precision	4倍精度実数ベクトルのlow word
INCX	i	integer	XH, XLの増分
BETAH	i	double precision	4倍精度実数のhigh word
BETAL	i	double precision	4倍精度実数のlow word
YH	i	double precision	4倍精度実数ベクトルのhigh word
YL	i	double precision	4倍精度実数ベクトルのlow word
INCY	i	integer	YH, YLの増分

# DDGER

概要：

一般行列のランク1の更新

$$A := \alpha * x * y^T + A$$

引数	I/O	型	説明
M	i	integer	行列の行または列の数
N	i	integer	行列の行または列の数
ALPHAH	i	double precision	4倍精度実数のhigh word
ALPHAL	i	double precision	4倍精度実数のlow word
XH	i	double precision	4倍精度実数ベクトルのhigh word
XL	i	double precision	4倍精度実数ベクトルのlow word
INCX	i	integer	XH, XLの増分
YH	i	double precision	4倍精度実数ベクトルのhigh word
YL	i	double precision	4倍精度実数ベクトルのlow word
INCY	i	integer	YH, YLの増分
AH	i/o	double precision	4倍精度実数行列のhigh word
AL	i/o	double precision	4倍精度実数行列のlow word
LDA	i	integer	行列Aの第一次元

# ***GDDSYMV***

概要：

対称行列とベクトルの演算

$$x := \alpha * A * x + \beta * y$$

引数	I/O	型	説明
UPLO	i	character(1)	三角行列の種類の指定 UPLO = 'U' or 'u' → 上三角行列 (右上成分を計算に用いる) UPLO = 'L' or 'l' → 下三角行列 (左下成分を計算に用いる)
N	i	integer	行列の行または列の数
ALPHAH	i	double precision	4倍精度実数のhigh word
ALPHAL	i	double precision	4倍精度実数のlow word
AH	i	double precision	4倍精度実数行列のhigh word
AL	i	double precision	4倍精度実数行列のlow word
LDA	i	integer	行列Aの第一次元
XH	i	double precision	4倍精度実数ベクトルのhigh word
XL	i	double precision	4倍精度実数ベクトルのlow word
INCX	i	integer	XH, XLの増分
BETAH	i	double precision	4倍精度実数のhigh word
BETAL	i	double precision	4倍精度実数のlow word
YH	i/o	double precision	4倍精度実数ベクトルのhigh word
YL	i/o	double precision	4倍精度実数ベクトルのlow word
INCY	i	integer	YH, YLの増分

# ***GDDSBMV***

概要：

対称帯行列とベクトルの演算

$$y := \alpha * A * x + \beta * y$$

引数	I/O	型	説明
UPLO	i	character(1)	三角行列の種類 UPLO = 'U' or 'u' → 上三角行列 (右上成分を計算に用いる) UPLO = 'L' or 'l' → 下三角行列 (左下成分を計算に用いる)
N	i	integer	行列の行または列の数
K	i	integer	バンド幅
ALPHAH	i	double precision	4倍精度実数のhigh word
ALPHAL	i	double precision	4倍精度実数のlow word
AH	i/o	double precision	4倍精度実数行列のhigh word
AL	i/o	double precision	4倍精度実数行列のlow word
LDA	i	integer	行列Aの第一次元
XH	i	double precision	4倍精度実数ベクトルのhigh word
XL	i	double precision	4倍精度実数ベクトルのlow word
INCX	i	integer	XH, XLの増分
BETAH	i	double precision	4倍精度実数のhigh word
BETAL	i	double precision	4倍精度実数のlow word
YH	i	double precision	4倍精度実数ベクトルのhigh word
YL	i	double precision	4倍精度実数ベクトルのlow word
INCY	i	integer	YH, YLの増分

# ***GDDSYR***

概要：

対称行列のランク1の更新

$$A := \alpha * x * x^T + A$$

引数	I/O	型	説明
UPLO	i	character(1)	三角行列の種類 UPLO = 'U' or 'u' → 上三角行列 (右上成分を計算に用いる) UPLO = 'L' or 'l' → 下三角行列 (左下成分を計算に用いる)
N	i	integer	行列の行または列の数
ALPHAH	i	double precision	4倍精度実数のhigh word
ALPHAL	i	double precision	4倍精度実数のlow word
XH	i	double precision	4倍精度実数ベクトルのhigh word
XL	i	double precision	4倍精度実数ベクトルのlow word
INCX	i	integer	XH, XLの増分
AH	i/o	double precision	4倍精度実数行列のhigh word
AL	i/o	double precision	4倍精度実数行列のlow word
LDA	i	integer	行列Aの第一次元

## ***GDDSYR2***

概要：

対称行列のランク2の更新

$$A := \alpha * x * y^T + \alpha * y * x^T + A$$

引数	I/O	型	説明
UPLO	i	character(1)	三角行列の種類 UPLO = 'U' or 'u' → 上三角行列 (右上成分を計算に用いる) UPLO = 'L' or 'l' → 下三角行列 (左下成分を計算に用いる)
N	i	integer	行列の行または列の数
ALPHAH	i	double precision	4倍精度実数のhigh word
ALPHAL	i	double precision	4倍精度実数のlow word
XH	i	double precision	4倍精度実数ベクトルのhigh word
XL	i	double precision	4倍精度実数ベクトルのlow word
INCX	i	integer	XH, XLの増分
YH	i	double precision	4倍精度実数ベクトルのhigh word
YL	i	double precision	4倍精度実数ベクトルのlow word
INCY	i	integer	YH, YLの増分
AH	i/o	double precision	4倍精度実数行列のhigh word
AL	i/o	double precision	4倍精度実数行列のlow word
LDA	i	integer	行列Aの第一次元

# ***GDDTRMV***

概要 :

三角行列とベクトルの演算

$x := A * x$

or

$x := A^T * x$

引数	I/O	型	説明
UPLO	i	character(1)	三角行列の種類 UPLO = 'U' or 'u' → 上三角行列 (右上成分を計算に用いる) UPLO = 'L' or 'l' → 下三角行列 (左下成分を計算に用いる)
TRANS	i	character(1)	行列の転置操作の制御 TRANS = 'N' or 'n' → 転置しない TRANS = 'T' or 't' → 転置する TRANS = 'C' or 'c' → 共役転置する
DIAG	i	character(1)	三角行列の対角成分の指定 DIAG = 'U' or 'u' → 単位三角行列である DIAG = 'N' or 'n' → 単位三角行列でない
N	i	integer	行列の行または列の数
AH	i	double precision	4倍精度実数行列のhigh word
AL	i	double precision	4倍精度実数行列のlow word
LDA	i	integer	行列Aの第一次元
XH	i/o	double precision	4倍精度実数ベクトルのhigh word
XL	i/o	double precision	4倍精度実数ベクトルのlow word
INCX	i	integer	XH, XLの増分

# ***GDDTBMV***

概要 :

三角帯行列とベクトルの演算

$x := A * x$

or

$x := A^T * x$

引数	I/O	型	説明
UPLO	i	character(1)	三角行列の種類 UPLO = 'U' or 'u' → 上三角行列 (右上成分を計算に用いる) UPLO = 'L' or 'l' → 下三角行列 (左下成分を計算に用いる)
TRANS	i	character(1)	行列の転置操作の制御 TRANS = 'N' or 'n' → 転置しない TRANS = 'T' or 't' → 転置する TRANS = 'C' or 'c' → 共役転置する
DIAG	i	character(1)	三角行列の対角成分の指定 DIAG = 'U' or 'u' → 単位三角行列である DIAG = 'N' or 'n' → 単位三角行列でない
N	i	integer	行列の行または列の数
K	i	integer	バンド幅
AH	i/o	double precision	4倍精度実数行列のhigh word
AL	i/o	double precision	4倍精度実数行列のlow word
LDA	i	integer	行列Aの第一次元
XH	i	double precision	4倍精度実数ベクトルのhigh word
XL	i	double precision	4倍精度実数ベクトルのlow word
INCX	i	integer	XH, XLの増分

# ***GDDTRSV***

概要 :

三角行列を含む式を解く

$$x := A^{-1} * x$$

or

$$x := (A^T)^{-1} * x$$

引数	I/O	型	説明
UPLO	i	character(1)	三角行列の種類 UPLO = 'U' or 'u' → 上三角行列 (右上成分を計算に用いる) UPLO = 'L' or 'l' → 下三角行列 (左下成分を計算に用いる)
TRANS	i	character(1)	行列の転置操作の制御 TRANS = 'N' or 'n' → 転置しない TRANS = 'T' or 't' → 転置する TRANS = 'C' or 'c' → 共役転置する
DIAG	i	character(1)	三角行列の対角成分の指定 DIAG = 'U' or 'u' → 単位三角行列である DIAG = 'N' or 'n' → 単位三角行列でない
N	i	integer	行列の行または列の数
AH	i	double precision	4倍精度実数行列のhigh word
AL	i	double precision	4倍精度実数行列のlow word
LDA	i	integer	行列Aの第一次元
XH	i/o	double precision	4倍精度実数ベクトルのhigh word
XL	i/o	double precision	4倍精度実数ベクトルのlow word
INCX	i	integer	XH, XLの増分

# ***GDDTBSV***

概要：

三角帯行列を含む式を解く

$$x := A^{-1} * x$$

or

$$x := (A^T)^{-1} * x$$

引数	I/O	型	説明
UPLO	i	character(1)	三角行列の種類 UPLO = 'U' or 'u' → 上三角行列 (右上成分を計算に用いる) UPLO = 'L' or 'l' → 下三角行列 (左下成分を計算に用いる)
TRANS	i	character(1)	行列の転置操作の制御 TRANS = 'N' or 'n' → 転置しない TRANS = 'T' or 't' → 転置する TRANS = 'C' or 'c' → 共役転置する
DIAG	i	character(1)	三角行列の対角成分の指定 DIAG = 'U' or 'u' → 単位三角行列である DIAG = 'N' or 'n' → 単位三角行列でない
N	i	integer	行列の行または列の数
K	i	integer	バンド幅
AH	i	double precision	4倍精度実数行列のhigh word
AL	i	double precision	4倍精度実数行列のlow word
LDA	i	integer	行列Aの第一次元
XH	i/o	double precision	4倍精度実数ベクトルのhigh word
XL	i/o	double precision	4倍精度実数ベクトルのlow word
INCX	i	integer	XH, XLの増分

# ***GZZGERC***

概要：

一般複素数行列のランク1の更新（第2ベクトルを共役転置）

$$A := \alpha * x * y^H + A$$

引数	I/O	型	説明
M	i	integer	行列の行または列の数
N	i	integer	行列の行または列の数
ALPHAH	i	double complex	4倍精度複素数のhigh word
ALPHAL	i	double complex	4倍精度複素数のlow word
XH	i	double complex	4倍精度複素数ベクトルのhigh word
XL	i	double complex	4倍精度複素数ベクトルのlow word
INCX	i	integer	XH, XLの増分
YH	i	double complex	4倍精度複素数ベクトルのhigh word
YL	i	double complex	4倍精度複素数ベクトルのlow word
INCY	i	integer	YH, YLの増分
AH	i/o	double complex	4倍精度複素数行列のhigh word
AL	i/o	double complex	4倍精度複素数行列のlow word
LDA	i	integer	行列Aの第一次元

# ***GZZGERU***

概要：

一般複素数行列のランク1の更新（第2ベクトルを転置）

$$A := \alpha * x * y^T + A$$

引数	I/O	型	説明
M	i	integer	行列の行または列の数
N	i	integer	行列の行または列の数
ALPHAH	i	double complex	4倍精度複素数のhigh word
ALPHAL	i	double complex	4倍精度複素数のlow word
XH	i	double complex	4倍精度複素数ベクトルのhigh word
XL	i	double complex	4倍精度複素数ベクトルのlow word
INCX	i	integer	XH, XLの増分
YH	i	double complex	4倍精度複素数ベクトルのhigh word
YL	i	double complex	4倍精度複素数ベクトルのlow word
INCY	i	integer	YH, YLの増分
AH	i/o	double complex	4倍精度複素数行列のhigh word
AL	i/o	double complex	4倍精度複素数行列のlow word
LDA	i	integer	行列Aの第一次元

# ***GZZHEMV***

概要：

エルミート行列とベクトルの演算

$$y := \alpha * A * x + \beta * y$$

引数	I/O	型	説明
UPLO	i	character(1)	三角行列の種類 UPLO = 'U' or 'u' → 上三角行列 (右上成分を計算に用いる) UPLO = 'L' or 'l' → 下三角行列 (左下成分を計算に用いる)
N	i	integer	行列の行または列の数
ALPHAH	i	double complex	4倍精度複素数のhigh word
ALPHAL	i	double complex	4倍精度複素数のlow word
AH	i	double complex	4倍精度複素数行列のhigh word
AL	i	double complex	4倍精度複素数行列のlow word
LDA	i	integer	行列Aの第一次元
XH	i	double complex	4倍精度複素数ベクトルのhigh word
XL	i	double complex	4倍精度複素数ベクトルのlow word
INCX	i	integer	XH, XLの増分
BETAH	i	double complex	4倍精度複素数のhigh word
BETAL	i	double complex	4倍精度複素数のlow word
YH	i/o	double complex	4倍精度複素数ベクトルのhigh word
YL	i/o	double complex	4倍精度複素数ベクトルのlow word
INCY	i	integer	YH, YLの増分

# ***GZZHBMV***

概要：

エルミート帯行列とベクトルの演算

$$y := \alpha * A * x + \beta * y$$

引数	I/O	型	説明
UPLO	i	character(1)	三角行列の種類 UPLO = 'U' or 'u' → 上三角行列 (右上成分を計算に用いる) UPLO = 'L' or 'l' → 下三角行列 (左下成分を計算に用いる)
N	i	integer	行列の行または列の数
K	i	integer	バンド幅
ALPHAH	i	double complex	4倍精度複素数のhigh word
ALPHAL	i	double complex	4倍精度複素数のlow word
AH	i	double complex	4倍精度複素数行列のhigh word
AL	i	double complex	4倍精度複素数行列のlow word
LDA	i	integer	行列Aの第一次元
XH	i	double complex	4倍精度複素数ベクトルのhigh word
XL	i	double complex	4倍精度複素数ベクトルのlow word
INCX	i	integer	XH, XLの増分
BETAH	i	double complex	4倍精度複素数のhigh word
BETAL	i	double complex	4倍精度複素数のlow word
YH	i/o	double complex	4倍精度複素数ベクトルのhigh word
YL	i/o	double complex	4倍精度複素数ベクトルのlow word
INCY	i	integer	YH, YLの増分

# ***GZZHER***

概要：

エルミート行列のランク1の更新

$$A := \alpha * x * x^H + A$$

引数	I/O	型	説明
UPLO	i	character(1)	三角行列の種類 UPLO = 'U' or 'u' → 上三角行列 (右上成分を計算に用いる) UPLO = 'L' or 'l' → 下三角行列 (左下成分を計算に用いる)
N	i	integer	行列の行または列の数
ALPHAH	i	double precision	4倍精度実数のhigh word
ALPHAL	i	double precision	4倍精度実数のlow word
XH	i	double complex	4倍精度複素数ベクトルのhigh word
XL	i	double complex	4倍精度複素数ベクトルのlow word
INCX	i	integer	XH, XLの増分
AH	i/o	double complex	4倍精度複素数行列のhigh word
AL	i/o	double complex	4倍精度複素数行列のlow word
LDA	i	integer	行列Aの第一次元

## ***GZZHER2***

概要：

エルミート行列のランク2の更新

$$A := \alpha * x * y^H + \text{conjg}(\alpha) * y * x^H + A$$

引数	I/O	型	説明
UPLO	i	character(1)	三角行列の種類 UPLO = 'U' or 'u' → 上三角行列 (右上成分を計算に用いる) UPLO = 'L' or 'l' → 下三角行列 (左下成分を計算に用いる)
N	i	integer	行列の行または列の数
ALPHAH	i	double complex	4倍精度複素数のhigh word
ALPHAL	i	double complex	4倍精度複素数のlow word
XH	i	double complex	4倍精度複素数ベクトルのhigh word
XL	i	double complex	4倍精度複素数ベクトルのlow word
INCX	i	integer	XH, XLの増分
YH	i	double complex	4倍精度複素数ベクトルのhigh word
YL	i	double complex	4倍精度複素数ベクトルのlow word
INCY	i	integer	YH, YLの増分
AH	i/o	double complex	4倍精度複素数行列のhigh word
AL	i/o	double complex	4倍精度複素数行列のlow word
LDA	i	integer	行列Aの第一次元

# ***GDDGEMM***

概要：

一般行列と行列の演算

$$C := \alpha * A * B + \beta * C$$

or

$$C := \alpha * A^T * B + \beta * C$$

or

$$C := \alpha * A * B^T + \beta * C$$

or

$$C := \alpha * A^T * B^T + \beta * C$$

引数	I/O	型	説明
TRANSA	i	character(1)	行列Aの転置操作の制御 TRANSA = 'N' or 'n' → 転置しない TRANSA = 'T' or 't' → 転置する TRANSA = 'C' or 'c' → 共役転置する
TRANSB	i	character(1)	行列Bの転置操作の制御 TRANSB = 'N' or 'n' → 転置しない TRANSB = 'T' or 't' → 転置する TRANSB = 'C' or 'c' → 共役転置する
M	i	integer	行列の行または列の数
N	i	integer	行列の行または列の数
K	i	integer	行列の行または列の数
ALPHAH	i	double precision	4倍精度実数のhigh word
ALPHAL	i	double precision	4倍精度実数のlow word
AH	i	double precision	4倍精度実数行列のhigh word
AL	i	double precision	4倍精度実数行列のlow word
LDA	i	integer	行列Aの第一次元
BH	i	double precision	4倍精度実数行列のhigh word
BL	i	double precision	4倍精度実数行列のlow word
LDB	i	integer	行列Bの第一次元
BETAH	i	double precision	4倍精度実数のhigh word

BETAL	i	double precision	4倍精度実数のlow word
CH	i/o	double precision	4倍精度実数行列のhigh word
CL	i/o	double precision	4倍精度実数行列のlow word
LDC	i	integer	行列Cの第一次元

# ***GDDSYMM***

概要：

対称行列と行列の演算

$C := \alpha * A * B + \beta * C$

or

$C := \alpha * B * A + \beta * C$

引数	I/O	型	説明
SIDE	i	character(1)	行列Aと行列Bの順番の指定 SIDE = 'L' or 'l' → A Bの順 SIDE = 'R' or 'r' → B Aの順
UPLO	i	character(1)	三角行列の種類指定 UPLO = 'U' or 'u' → 上三角行列 (右上成分を計算に用いる) UPLO = 'L' or 'l' → 下三角行列 (左下成分を計算に用いる)
M	i	integer	行列の行または列の数
N	i	integer	行列の行または列の数
ALPHAH	i	double precision	4倍精度実数のhigh word
ALPHAL	i	double precision	4倍精度実数のlow word
AH	i	double precision	4倍精度実数行列のhigh word
AL	i	double precision	4倍精度実数行列のlow word
LDA	i	integer	行列Aの第一次元
BH	i	double precision	4倍精度実数行列のhigh word
BL	i	double precision	4倍精度実数行列のlow word
LDB	i	integer	行列Bの第一次元
BETAH	i	double precision	4倍精度実数のhigh word
BETAL	i	double precision	4倍精度実数のlow word
CH	i/o	double precision	4倍精度実数行列のhigh word
CL	i/o	double precision	4倍精度実数行列のlow word
LDC	i	integer	行列Cの第一次元

# ***GDDSYRK***

概要：

対称行列のランクkの更新

$C := \alpha * A * A^T + \beta * C$

or

$C := \alpha * A^T * A + \beta * C$

引数	I/O	型	説明
SIDE	i	character(1)	行列Aと行列Bの順番の指定 SIDE = 'L' or 'l' → A Bの順 SIDE = 'R' or 'r' → B Aの順
UPLO	i	character(1)	三角行列の種類指定 UPLO = 'U' or 'u' → 上三角行列 (右上成分を計算に用いる) UPLO = 'L' or 'l' → 下三角行列 (左下成分を計算に用いる)
N	i	integer	行列の行または列の数
K	i	integer	行列の行または列の数
ALPHAH	i	double precision	4倍精度実数のhigh word
ALPHAL	i	double precision	4倍精度実数のlow word
AH	i	double precision	4倍精度実数行列のhigh word
AL	i	double precision	4倍精度実数行列のlow word
LDA	i	integer	行列Aの第一次元
BETAH	i	double precision	4倍精度実数のhigh word
BETAL	i	double precision	4倍精度実数のlow word
CH	i/o	double precision	4倍精度実数行列のhigh word
CL	i/o	double precision	4倍精度実数行列のlow word
LDC	i	integer	行列Cの第一次元

# GDDSYR2K

概要：

対称行列のランク2kの更新

$$C := \alpha * A * B^T + \alpha * B * A^T + \beta * C$$

or

$$C := \alpha * B^T * A + \alpha * A^T * B + \beta * C$$

引数	I/O	型	説明
UPLO	i	character(1)	三角行列の種類 UPLO = 'U' or 'u' → 上三角行列 (右上成分を計算に用いる) UPLO = 'L' or 'l' → 下三角行列 (左下成分を計算に用いる)
TRANS	i	character(1)	行列の転置操作の制御 TRANS = 'N' or 'n' → 転置しない TRANS = 'T' or 't' → 転置する TRANS = 'C' or 'c' → 共役転置する
N	i	integer	行列の行または列の数
K	i	integer	行列の行または列の数
ALPHAH	i	double precision	4倍精度実数のhigh word
ALPHAL	i	double precision	4倍精度実数のlow word
AH	i	double precision	4倍精度実数行列のhigh word
AL	i	double precision	4倍精度実数行列のlow word
LDA	i	integer	行列Aの第一次元
BH	i	double precision	4倍精度実数行列のhigh word
BL	i	double precision	4倍精度実数行列のlow word
LDB	i	integer	行列Bの第一次元
BETAH	i	double precision	4倍精度実数のhigh word
BETAL	i	double precision	4倍精度実数のlow word
CH	i/o	double precision	4倍精度実数行列のhigh word
CL	i/o	double precision	4倍精度実数行列のlow word
LDC	i	integer	行列Cの第一次元

# ***GDDTRMM***

概要 :

三角行列と行列の演算

$B := \alpha * A * B$

or

$B := \alpha * A^T * B$

or

$B := \alpha * B * A$

or

$B := \alpha * B * A^T$

引数	I/O	型	説明
SIDE	i	character(1)	行列Aと行列Bの順番の指定 SIDE = 'L' or 'l' → A Bの順 SIDE = 'R' or 'r' → B Aの順
UPLO	i	character(1)	三角行列の種類指定 UPLO = 'U' or 'u' → 上三角行列 (右上成分を計算に用いる) UPLO = 'L' or 'l' → 下三角行列 (左下成分を計算に用いる)
TRANS	i	character(1)	行列Aの転置操作の制御 TRANS = 'N' or 'n' → 転置しない TRANS = 'T' or 't' → 転置する TRANS = 'C' or 'c' → 共役転置する
DIAG	i	character(1)	三角行列の対角成分の指定 DIAG = 'U' or 'u' → 単位三角行列である DIAG = 'N' or 'n' → 単位三角行列でない
M	i	integer	行列の行または列の数
N	i	integer	行列の行または列の数
ALPHAH	i	double precision	4倍精度実数のhigh word
ALPHAL	i	double precision	4倍精度実数のlow word
AH	i	double precision	4倍精度実数行列のhigh word
AL	i	double precision	4倍精度実数行列のlow word

LDA	i	integer	行列Aの第一次元
BH	i/o	double precision	4倍精度実数行列のhigh word
BL	i/o	double precision	4倍精度実数行列のlow word
LDB	i	integer	行列Bの第一次元

# GDDTRSM

概要：

三角行列を含む式を解く

$B := \alpha * A^{-1} * B$

or

$B := \alpha * B * A^{-1}$

引数	I/O	型	説明
SIDE	i	character(1)	行列Aと行列Bの順番の指定 SIDE = 'L' or 'l' → A Bの順 SIDE = 'R' or 'r' → B Aの順
UPLO	i	character(1)	三角行列の種類指定 UPLO = 'U' or 'u' → 上三角行列 (右上成分を計算に用いる) UPLO = 'L' or 'l' → 下三角行列 (左下成分を計算に用いる)
TRANS	i	character(1)	行列Aの転置操作の制御 TRANS = 'N' or 'n' → 転置しない TRANS = 'T' or 't' → 転置する TRANS = 'C' or 'c' → 共役転置する
DIAG	i	character(1)	三角行列の対角成分の指定 DIAG = 'U' or 'u' → 単位三角行列である DIAG = 'N' or 'n' → 単位三角行列でない
M	i	integer	行列の行または列の数
N	i	integer	行列の行または列の数
ALPHAH	i	double precision	4倍精度実数のhigh word
ALPHAL	i	double precision	4倍精度実数のlow word
AH	i	double precision	4倍精度実数行列のhigh word
AL	i	double precision	4倍精度実数行列のlow word
LDA	i	integer	行列Aの第一次元
BH	i/o	double precision	4倍精度実数行列のhigh word
BL	i/o	double precision	4倍精度実数行列のlow word

LDB	i	integer	行列Bの第一次元
-----	---	---------	----------

# ***GZZHEMM***

概要：

エルミート行列と行列の演算

$C := \alpha * A * B + \beta * C$

or

$C := \alpha * B * A + \beta * C$

引数	I/O	型	説明
SIDE	i	character(1)	行列Aと行列Bの順番の指定 SIDE = 'L' or 'l' → A Bの順 SIDE = 'R' or 'r' → B Aの順
UPLO	i	character(1)	三角行列の種類指定 UPLO = 'U' or 'u' → 上三角行列 (右上成分を計算に用いる) UPLO = 'L' or 'l' → 下三角行列 (左下成分を計算に用いる)
M	i	integer	行列の行または列の数
N	i	integer	行列の行または列の数
ALPHAH	i	double precision	4倍精度実数のhigh word
ALPHAL	i	double precision	4倍精度実数のlow word
AH	i	double precision	4倍精度実数行列のhigh word
AL	i	double precision	4倍精度実数行列のlow word
LDA	i	integer	行列Aの第一次元
BH	i	double precision	4倍精度実数行列のhigh word
BL	i	double precision	4倍精度実数行列のlow word
LDB	i	integer	行列Bの第一次元
BETAH	i	double precision	4倍精度実数のhigh word
BETAL	i	double precision	4倍精度実数のlow word
CH	i/o	double precision	4倍精度実数行列のhigh word
CL	i/o	double precision	4倍精度実数行列のlow word
LDC	i	integer	行列Cの第一次元

# ***GZZHERK***

概要：

エルミート行列のランクkの更新

$C := \alpha * A * A^H + \beta * C$

or

$C := \alpha * A^H * A + \beta * C$

引数	I/O	型	説明
UPLO	i	character(1)	三角行列の種類 UPLO = 'U' or 'u' → 上三角行列 (右上成分を計算に用いる) UPLO = 'L' or 'l' → 下三角行列 (左下成分を計算に用いる)
TRANS	i	character(1)	行列の転置操作の制御 TRANS = 'N' or 'n' → 転置しない TRANS = 'C' or 'c' → 共役転置する
N	i	integer	行列の行または列の数
K	i	integer	行列の行または列の数
ALPHAH	i	double precision	4倍精度実数のhigh word
ALPHAL	i	double precision	4倍精度実数のlow word
AH	i	double complex	4倍精度複素数行列のhigh word
AL	i	double complex	4倍精度複素数行列のlow word
LDA	i	integer	行列Aの第一次元
BETAH	i	double precision	4倍精度実数のhigh word
BETAL	i	double precision	4倍精度実数のlow word
CH	i/o	double complex	4倍精度複素数行列のhigh word
CL	i/o	double complex	4倍精度複素数行列のlow word
LDC	i	integer	行列Cの第一次元

# GZZHER2K

概要：

エルミート行列のランク2kの更新

$C := \alpha * A * B^H + \alpha * B * A^H + \beta * C$

or

$C := \alpha * B^H * A + \alpha * A^H * B + \beta * C$

引数	I/O	型	説明
UPLO	i	character(1)	三角行列の種類 UPLO = 'U' or 'u' → 上三角行列 (右上成分を計算に用いる) UPLO = 'L' or 'l' → 下三角行列 (左下成分を計算に用いる)
TRANS	i	character(1)	行列の転置操作の制御 TRANS = 'N' or 'n' → 転置しない TRANS = 'C' or 'c' → 共役転置する
N	i	integer	行列の行または列の数
K	i	integer	行列の行または列の数
ALPHAH	i	double complex	4倍精度複素数のhigh word
ALPHAL	i	double complex	4倍精度複素数のlow word
AH	i	double complex	4倍精度複素数行列のhigh word
AL	i	double complex	4倍精度複素数行列のlow word
LDA	i	integer	行列Aの第一次元
BH	i	double complex	4倍精度複素数行列のhigh word
BL	i	double complex	4倍精度複素数行列のlow word
LDB	i	integer	行列Bの第一次元
BETAH	i	double precision	4倍精度実数のhigh word
BETAL	i	double precision	4倍精度実数のlow word
CH	i/o	double complex	4倍精度複素数行列のhigh word
CL	i/o	double complex	4倍精度複素数行列のlow word
LDC	i	integer	行列Cの第一次元

