

**4 倍精度 BLAS ルーチン
QPBLAS 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.....	3
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.....	4
6	インストール方法.....	5
7	検証用・性能評価用プログラム.....	7
8	サンプルプログラム.....	8
9	サブルーチン詳細.....	11

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 BLAS の全 40 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、本ライブラリ群を「QPBLAS」として公開している。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

となる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。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. High-Precision Software Directory.
<http://crd-legacy.lbl.gov/~dhbailey/mpdist>.

山田進, 佐々成正, 今村俊幸, 町田昌彦. “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 [情報処理学会研究報告] 137, 第 23 [2012]: 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造になる。

—— ddblas	double-double 4倍精度化BLASルーチン
—— qblas	128bit 4倍精度BLASルーチン
—— 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

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-version.tar.gz
cd QPBLAS-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。

FC=f90_compiler	Fortran コンパイラの指定
FCFLAGS= <i>options</i>	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。
LDFLAGS= <i>options</i>	リンクオプションの指定
--prefix= <i>path</i>	インストールディレクトリの変更
--disable-openmp	OpenMP による並列化を無効にする 指定されなかった場合は、OpenMP による並列化を行う
--without-dblas	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、 LDFLAGS 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
```

例：Intel Fortran を利用する場合

```
./configure FC=ifort
```

例：Intel Fortran および MKL を利用する場合

```
./configure --without-dblas FC=ifort LDFLAGS=-mkl
```

例：Intel Fortran および MKL を利用し、OpenMP による並列化を行わない場合

```
./configure --without-dblas --disable-openmp ¥  
FC=ifort LDFLAGS="-mkl=sequential"
```

例：PGI Fortran を利用する場合

```
./configure FC=pgfortran
```

例：PGI Fortran とその BLAS ライブラリを利用する場合

```
./configure --without-dblas FC=pgfortran LDFLAGS=-lblas
```

例：Parallelnavi Language Package Fortran を利用する場合

```
./configure FC=frt FCFLAGS=-Am
```

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。

lib/libddblas.a	QPBLAS ライブラリ
lib/libqblas.a	4 倍精度 BLAS ライブラリ (作成できる場合のみ)
lib/libdblas.a	倍精度 BLAS ライブラリ (必要な場合のみ)
test/level1	Level 1 の検証用プログラム
test/level2	Level 2 の検証用プログラム
test/level3	Level 3 の検証用プログラム
sample/ddnrm2_sample	DDNRM2 ルーチンを用いたサンプルプログラム
sample/ddgemv_sample	DDGEMV ルーチンを用いたサンプルプログラム
sample/ddtrsv_sample	DDTRSV ルーチンを用いたサンプルプログラム

sample/zzdotu_sample	ZZDOTC ルーチンを用いたサンプルプログラム
sample/zzhemm_sample	ZZHEMM ルーチンを用いたサンプルプログラム
sample/zzhemv_sample	ZZHEMV ルーチンを用いたサンプルプログラム

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 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 DDAXPY 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 のサンプルプログラムがいくつか用意されている。

ddnrm2_sample.f90	ベクトルの 2 ノルム
ddtrsv_sample.f90	三角行列を含む式を解く
ddgemm_sample.f90	一般行列と行列の演算
zzdotu_sample.f90	複素数ベクトルの内積
zzhemv_sample.f90	エルミート行列とベクトルの演算
zzhemm_sample.f90	エルミート行列と行列の演算

8.1 ベクトルの 2 ノルム (ddnrm2_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。

8.2 三角行列を解く (ddtrsv_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。

8.3 一般行列と行列の演算 (ddgemm_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 α および β を乱数で与えて、 $C = \alpha AB + \beta C$ を計算する。

8.4 複素数ベクトルの内積 (zzdotu_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。

8.5 エルミート行列とベクトルの演算 (zzhemv_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 α および β を乱数で与えて、 $y = \alpha Ax + \beta y$ を計算する。

8.6 エルミート行列と行列の演算 (zzhemm_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 α および β を乱数で与えて、 $C = \alpha AB + \beta C$ を計算する。

8.7 サンプルプログラムの実行方法

これらのサンプルプログラムは `make` の際にコンパイルされているため、例えば `zzdotu_sample` の場合は次のように入力することで実行できる。

```
cd sample
./zzdotu_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 -fopenmp zzdotu_sample.f90 ../lib/libddblas.a
```

なお、QPBLAS に含まれるいくつかのルーチンは BLAS のユーティリティ関数に依存しているため、リンク時に次のようなエラーが生じることがある。

```
gfortran -fopenmp ddgemv_sample.f90 ../lib/libddblas.a
...
undefined reference to `lsame_'
undefined reference to `xerbla_'
```

この場合は、BLAS へのリンクを追加すればよい。

```
gfortran -fopenmp -lblas ddgemv_sample.f90 ../lib/libddblas.a
```

サンプルプログラム `zzdotu_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 zzdotu(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(:) = cmplx(re, im)

end subroutine

end program

```

9 サブルーチン詳細

QPBLAS 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。

Level 1	DDSWAP	ベクトルの入れ替え	$y \leftrightarrow x$
	DDCOPY	ベクトルのコピー	$y \rightarrow x$
	DDSCAL	ベクトルのスケーリング	$y \rightarrow \alpha x$
	DDAXPY	ベクトルの演算	$y \rightarrow \alpha x + y$
	DDDOT	ベクトルの内積	$x^T y$
	DDNRM2	ベクトルの 2-ノルム	$\sum x_i^2$
	DDASUM	ベクトルの和 (1-ノルム)	$\sum x_i $
	IDDAMAX	絶対値が最大の値を持つベクトルの番号	
	DDRROTG	ギブズ回転行列の作成	
	DDRROT	ギブズ回転の適用	
	DDRROTMG	修正ギブズ回転行列の作成	
	DDRROTM	修正ギブズ回転の適用	
ZZDOTC	複素数ベクトルの内積 (第 1 ベクトルを共役転置)	$\bar{x}^T y$	

	ZZDOTU	複素数ベクトルの内積 (第 1 ベクトルを転置)	$x^T y$
Level2	DDGEMV	一般行列とベクトルの演算	$y \rightarrow \alpha Ax + \beta y$ $y \rightarrow \alpha A^T x + \beta y$
	DDGBMV	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	$A \rightarrow \alpha xy^T + A$
	DDSYMV	対称行列とベクトルの演算	$y \rightarrow \alpha Ax + \beta y$
	DDSBMV	対称帯行列とベクトルの演算	$y \rightarrow \alpha Ax + \beta y$
	DDSYR	対称行列のランク 1 の更新	$A \rightarrow \alpha xx^T + A$
	DDSYR2	対称行列のランク 2 の更新	$A \rightarrow \alpha xy^T + \alpha yx^T + A$
	DDTRMV	三角行列とベクトルの演算	$x \rightarrow Ax$ $x \rightarrow A^T x$
	DDTBMV	三角帯行列とベクトルの演算	$x \rightarrow Ax$ $x \rightarrow A^T x$
	DDTRSV	三角行列を含む式を解く	$Ax = b$ $A^T x = b$
	DDTBSV	三角帯行列を含む式を解く	$Ax = b$ $A^T x = b$
	ZZGERC	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)	$A \rightarrow \alpha x \bar{y}^T + A$
	ZZGERU	一般複素数行列のランク 1 の更新 (第 2 ベクトルを転置)	$A \rightarrow \alpha xy^T + A$
	ZZHEMV	エルミート行列とベクトルの演算	$y \rightarrow \alpha Ax + \beta y$
	ZZHBMV	エルミート帯行列とベクトルの演算	$y \rightarrow \alpha Ax + \beta y$
ZZHER	エルミート行列のランク 1 の更新	$A \rightarrow \alpha x \bar{x}^T + A$	
ZZHER2	エルミート行列のランク 2 の更新	$A \rightarrow \alpha x \bar{y}^T + \alpha y \bar{x}^T + A$	
Level	DDGEMM	一般行列と行列の演算	$C \rightarrow \alpha AB + \gamma C$ $C \rightarrow \alpha A^T B + \gamma C$

			$C \rightarrow \alpha AB^T + \gamma C$ $C \rightarrow \alpha A^T B^T + \gamma C$
DDSYMM	対称行列と行列の演算		$\alpha AB + \gamma C$ $\alpha BA + \gamma C$
DDSYRK	対称行列のランク k の更新		$C \rightarrow \alpha AA^T + \beta C$ $C \rightarrow \alpha A^T A + \beta C$
DDSYR2K	対称行列のランク $2k$ の更新		$C \rightarrow \alpha AB^T + \alpha BA^T + \beta C$ $C \rightarrow \alpha A^T B + \alpha B^T A + \beta C$
DDTRMM	三角行列と行列の演算		$B \rightarrow \alpha A^* B$ $B \rightarrow \alpha B A^*$
DDTRSM	三角行列を含む式を解く		$Ax = b$ $A^T x = b$
ZZHEMM	エルミート行列と行列の演算		$C \rightarrow \alpha AB + \beta C$ $C \rightarrow \alpha BA + \beta C$
ZZHERK	エルミート行列のランク k の更新		$C \rightarrow \alpha A \bar{A}^T + \beta C$ $C \rightarrow \alpha \bar{A}^T A + \beta C$
ZZHER2K	エルミート行列のランク $2k$ の更新		$C \rightarrow \alpha A \bar{B}^T + \alpha \bar{B} A^T + \beta C$ $C \rightarrow \alpha \bar{A}^T B + \alpha \bar{B}^T A + \beta C$

DDSWAP

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

DDCOPY

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

DDSCAL

概要：

ベクトルのスケールリング

$x := a * x$

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

DDAXPY

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

DDDOT

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

DDNRM2

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

DDASUM

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

IDDAMAX

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
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
ret	ret	integer	絶対値が最大の値を持つベクトルの番号

DDROTG

概要：

ギブズ回転行列の作成

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	生成されたギブズ回転データの余弦成分の high word
CL	o	double precision	生成されたギブズ回転データの余弦成分の low word
SH	o	double precision	生成されたギブズ回転データの正弦成分の high word
SL	o	double precision	生成されたギブズ回転データの正弦成分の low word

DDROT

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

DDROTMG

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

DDROTM

概要：

修正ギブズ回転の適用

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	適用する修正ギブズ回転パラメータの high word
PARAML	i	double precision	適用する修正ギブズ回転パラメータの low word

ZZDOTC

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

ZZDOTU

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

DDGEMV

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

DDGBMV

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

DDSYMV

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 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

DDSBMV

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

DDSYR

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

DDSYR2

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

DDTRMV

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

DDTBMV

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

DDTRSV

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

DDTBSV

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

ZZGERC

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

ZZGERU

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

ZZHEMV

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

ZZHBMV

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

ZZHER

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

ZZHER2

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

DDGEMM

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

DDSYMM

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

DDSYRK

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

DDSYR2K

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

DDTRMM

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

DDTRSM

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)
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
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

ZZHEMM

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

ZZHERK

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

ZZHER2K

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