

	表紙	
対象OS名 又は プロジェクト名	複素対称行列用 ハウスホルダ3重対角化変換機能の開発	
文書名	MPI対応の複素対称行列用 ハウスホルダ3重対角化法プログラム 内部データ仕様書	
作成日時	2012年10月24日	

## 目次

1. 概要 .....	2
2. 複素対称行列用ハウスホルダ 3 重対角化変換プログラム PZSYTRD .....	3
2. 1. PZSYTRD 機能概要 .....	3
2. 2. j5initzc 機能概要 .....	4
2. 3. j5pahous 機能概要 .....	5
2. 4. j5mvlowp 機能概要 .....	6
2. 5. j5ranknb 機能概要 .....	7
2. 6. j5parank 機能概要 .....	8
2. 7. j5serial 機能概要 .....	9
2. 8. j5rngcom 機能概要 .....	10
3. 固有ベクトルのハウスホルダ逆変換プログラム PZSYGHR .....	11
3. 1. PZSYGHR 機能概要 .....	11
3. 2. j5inivec 機能概要 .....	13
3. 3. j5makvec 機能概要 .....	14

全 14 頁

## 1. 概要

本稿は複素対称行列用ハウスホルダ 3 重対角化変換プログラム(PZSYTRD)と固有ベクトルのハウスホルダ逆変換プログラム(PZSYGHR)の内部機能について説明します。図 1-1 に PZSYTRD の機能ブロック図を、図 1-2 に PZSYGHR の機能ブロック図を示します。

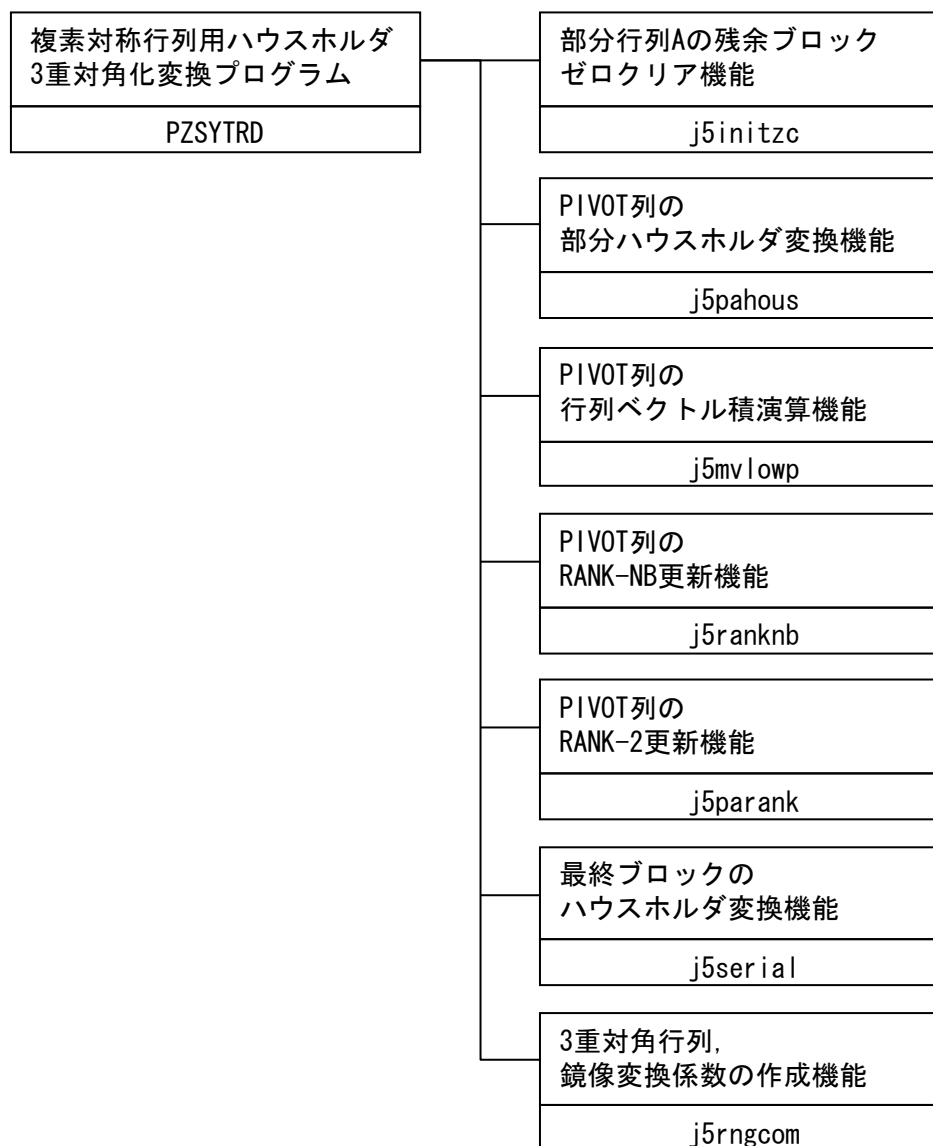


図 1-1 複素対称行列用ハウスホルダ 3 重対角化変換プログラム(PZSYTRD)機能ブロック図

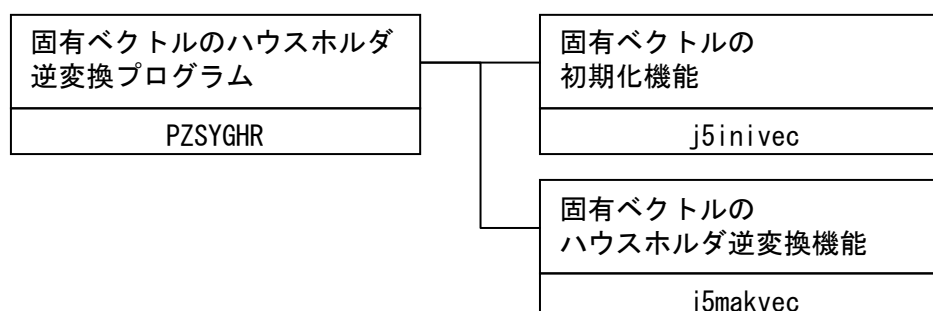


図 1-2 固有ベクトルのハウスホルダ逆変換プログラム(PZSYGHR)機能ブロック図

## 2. 複素対称行列用ハウスホルダ 3 重対角化変換プログラム PZSYTRD

複素対称行列用ハウスホルダ 3 重対角化変換プログラム PZSYTRD は、複素対称行列  $C$  をハウスホルダ変換  $T = HCH^T$  によって 3 重対角行列  $T$  へ変換するプログラムです。本プログラムではハウスホルダ変換の係数行列作成  $H = (\prod_{k=1}^n H_k)$ ,  $H_k = (I - \alpha u_k u_k^T)$  に多段変換型のブロック化アルゴリズムを採用しています。PZSYTRD は 8 つのサブルーチンで構成されており、2. 1. 節以降で各サブルーチンの機能について説明します。

### 2. 1. PZSYTRD 機能概要

PZSYTRD は複素対称行列用ハウスホルダ 3 重対角化変換プログラムのインターフェイスです。部分行列  $A$  を入力すると、 $A$  にハウスホルダ 3 重対角化で使用したリフレクタ情報が、 $D$  に 3 重対角行列の対角成分が、 $E$  に 3 重対角行列の副対角成分が、 $TAU$  にハウスホルダ変換の鏡像変換係数が出力されます。

PZSYTRD の引数仕様を表 2-1 に、リターンコード一覧を表 2-2 に示します。表 2-1 中の  $NB$ ,  $NLB$  は以下の式で定義します。

- ・  $NB = DESCA(6)$
- ・  $NLB = (N - 1) / NB + 1$

表 2-1 PZSYTRD 引数仕様

引数	型	種別	引数の説明
UPLO	Character*1	INPUT	入力行列の格納形式。UPLO='L'のみ指定可能
N	Integer	INPUT	全体行列の次元数。 $N \geq 2$
A(*)	Complex*16	INPUT / OUTPUT	(INPUT) : プロセスマップに従って分割した部分行列 (OUTPUT) : 3 重対角化で使用したリフレクタ情報
IA	Integer	INPUT	部分行列 $A$ の行方向先頭アドレス。IA=1 のみ指定可能
JA	Integer	INPUT	部分行列 $A$ の列方向先頭アドレス。JA=1 のみ指定可能
DESCA(9)	Integer	INPUT	ディスクリプタ <ul style="list-style-type: none"> <li>・ DESCA(6) 部分行列のブロックサイズ <math>NB</math> <math>NB \geq 8</math> かつ <math>MOD(NB,2)=0</math></li> <li>・ DESCA(9) 部分行列 <math>A</math> の整合寸法 <math>DESCA(9) \geq NLB * NB</math></li> <li>・ DESCA(1)~DESCA(5), DESCA(7), DESCA(8) 値入力の必要なし</li> </ul>
D(N)	Complex*16	OUTPUT	3 重対角行列の対角成分。全プロセスで同じ値が入る
E(N)	Complex*16	OUTPUT	3 重対角行列の副対角成分。全プロセスで同じ値が入る
TAU(N)	Complex*16	OUTPUT	ハウスホルダ変換の鏡像変換係数。全プロセスで同じ値が入る
WORK	Complex*16	WORK	値入力の必要なし (作業領域)
LWORK	Integer	INPUT	複素数型作業領域配列 WORK のサイズ $LWORK \geq 12 * NLB * NB + 4 * NB + 4 * NB * NLB * NB$
INFO	Integer	OUTPUT	リターンコード
MAJOR	Character*1	INPUT	プロセスマップ方向。MAJOR='C'または MAJOR='R'のみ指定可能

表 2-2 PZSYTRD リターンコード一覧

値	内容
0	正常終了したことを表す
-1	UPLO='L'が設定されていないことを表す
-2	$N \geq 2$ が設定されていないことを表す
-4	IA=1 が設定されていないことを表す
-5	JA=1 が設定されていないことを表す
-66	DESCA(6) $\geq 8$ かつ MOD(DESCA(6),2)=0 が設定されていないことを表す
-69	DESCA(9) $\geq$ NLB*NB が設定されていないことを表す
-11	LWORK $\geq 12 * \text{NLB} * \text{NB} + 4 * \text{NB} + 4 * \text{NB} * \text{NLB} * \text{NB}$ が設定されていないことを表す
-13	MAJOR='C'または MAJOR='R'が設定されていないことを表す
100	プロセスマップが正方かつ 1 辺が 2 べき数でないことを表す

## 2. 2. j5initzc 機能概要

j5initzc は部分行列 A を入力すると、対角ブロックの上三角部分と係数行列の N より大きい要素に相当する部分行列の要素をゼロクリアする機能です。j5initzc の引数仕様を表 2-3 に示します。

表 2-3 j5initzc 引数仕様

引数	型	種別	引数の説明
A(LDA,*)	Complex*16	INPUT / OUTPUT	(INPUT) : 部分行列 A (OUTPUT) : 対角ブロックの上三角部分と、部分行列 A の N より大きい部分をゼロクリアした行列
LDA	Integer	INPUT	部分行列 A の整合寸法 ( $LDA \geq \text{NLB} * \text{NB}$ )
NB	Integer	INPUT	ブロックサイズ ( $\text{NB} \geq 8$ , かつ MOD(NB,2)=0)
NLB	Integer	INPUT	MPI プロセスが保持する部分行列 A のブロック数
IPROW	Integer	INPUT	プロセスマップ中の MPI プロセス行番号
IPCOL	Integer	INPUT	プロセスマップ中の MPI プロセス列番号
KBPIV	Integer	INPUT	係数行列の最終ブロックを保持する MPI プロセスの列番号
NMOD	Integer	INPUT	係数行列の最終ブロック要素数

## 2. 3. j5pahous 機能概要

j5pahous は列 PIVOT ブロックを保持するプロセスで第 K 列のハウスホルダ変換を行う機能です。列 PIVOT ブロックを保持する MPI プロセスの部分行列 A, 列 PIVOT ブロック PQ と行 PIVOT ブロック QTPT を入力すると, 部分行列 A の第 K 列にハウスホルダ変換を適用し, 3 重対角行列の対角要素 DBUF, 副対角要素 EBUF, ハウスホルダ変換の鏡像変換係数 TBUF, 第 K 列の列 PIVOT P1 と第 K 列の鏡像変換係数 GAMMA を出力します。j5pahous の引数仕様を表 2-4 に示します。

表 2-4 j5pahous 引数仕様

引数	型	種別	引数の説明
A(LDA,*)	Complex*16	INPUT / OUTPUT	(INPUT) : 部分行列 A (OUTPUT) : ブロック内の K 列について部分ハウスホルダ変換を適用した結果
LDA	Integer	INPUT	部分行列 A の整合寸法 ( $LDA \geq NLB * NB$ )
NB	Integer	INPUT	ブロックサイズ ( $NB \geq 8$ , かつ $MOD(NB, 2) = 0$ )
NLB	Integer	INPUT	MPI プロセスが保持する部分行列 A のブロック数
JBLSTART2(NLB)	Integer	INPUT	部分行列 A の行方向開始位置
K	Integer	INPUT	ブロック内の部分ハウスホルダ変換対象列番号
NPROW	Integer	INPUT	プロセスマップの行方向 MPI プロセス数
IPROW	Integer	INPUT	プロセスマップ中の MPI プロセス行番号
IPCOL	Integer	INPUT	プロセスマップ中の MPI プロセス列番号
KBLK	Integer	INPUT	部分行列 A の開始ブロック番号
KBPIV	Integer	INPUT	列 PIVOT ブロックを保持する MPI プロセスの列番号
DBUF(NLB*NB)	Complex*16	OUTPUT	各プロセスが保持する 3 重対角行列の対角要素
EBUF(NLB*NB)	Complex*16	OUTPUT	各プロセスが保持する 3 重対角行列の副対角要素
TBUF(NLB*NB)	Complex*16	OUTPUT	各プロセスが保持するハウスホルダ変換の鏡像変換係数
P1(NLB*NB)	Complex*16	OUTPUT	第 K 列の列 PIVOT
PQ(2*NB,NLB*NB)	Complex*16	INPUT	列 PIVOT ブロック P, Q
QTPT(2*NB,NLB*NB)	Complex*16	INPUT	行 PIVOT ブロック $Q^T, P^T$
GAMMA	Complex*16	OUTPUT	ブロック内の K 列で求められる鏡像変換係数
MPI_COMM_ROW	Integer	INPUT	行方向の MPI 通信を行うためのコミュニケータ
MPI_COMM_COL	Integer	INPUT	列方向の MPI 通信を行うためのコミュニケータ

## 2. 4. j5mvlwp 機能概要

j5mvlwp は部分行列 A と列 PIVOT P1 で行列ベクトル積を計算し、列 PIVOT Q と行 PIVOT  $Q^T$  を作成する機能です。列 PIVOT ブロックを保持する MPI プロセスから列 PIVOT P1 を送信し、部分行列 A と P1 の行列ベクトル積演算を行います。また、この結果と P1 と行 PIVOT ブロック QTPT の内積結果を使用して列 PIVOT Q を作成し、列 PIVOT ブロック PQ と行 PIVOT ブロック QTPT の更新を行います。j5mvlwp の引数仕様を表 2-5 に示します。

表 2-5 j5mvlwp 引数仕様

引数	型	種別	引数の説明
A(LDA,*)	Complex*16	INPUT	部分行列 A。j5pahous の結果を入れればよい
LDA	Integer	INPUT	部分行列 A の整合寸法 ( $LDA \geq NLB * NB$ )
NB	Integer	INPUT	ブロックサイズ ( $NB \geq 8$ , かつ $MOD(NB,2)=0$ )
JBLSTART(NLB)	Integer	INPUT	対角ブロックを含まない部分行列 A の行方向開始位置
GAMMA	Complex*16	INPUT	ブロック内の K 列で求められる鏡像変換係数。j5pahous の結果を入れればよい
K	Integer	INPUT	ブロック内の部分ハウスホルダ変換対象列番号
NPROW	Integer	INPUT	プロセスマップの行方向 MPI プロセス数
IPROW	Integer	INPUT	プロセスマップ中の MPI プロセス行番号
IPCOL	Integer	INPUT	プロセスマップ中の MPI プロセス列番号
NLB	Integer	INPUT	MPI プロセスが保持する部分行列 A のブロック数
KBPIV	Integer	INPUT	列 PIVOT ブロックを保持する MPI プロセスの列番号
P1(NLB*NB)	Complex*16	INPUT	第 K 列の列 PIVOT。j5pahous の結果を入れればよい
P2(NLB*NB)	Complex*16	WORK	行 PIVOT $P^T$ を格納する作業領域
Q1(NLB*NB)	Complex*16	WORK	列 PIVOT Q を格納する作業領域
Q2(NLB*NB)	Complex*16	WORK	行 PIVOT $Q^T$ を格納する作業領域
PQ(2*NB,NLB*NB)	Complex*16	INPUT / OUTPUT	(INPUT) : 列 PIVOT ブロック P, Q (OUTPUT) : 第 K 列の列 PIVOT P, Q が格納される
QTPT(2*NB,NLB*NB)	Complex*16	INPUT / OUTPUT	(INPUT) : 行 PIVOT ブロック $Q^T$ , $P^T$ (OUTPUT) : 第 K 列の行 PIVOT $Q^T$ , $P^T$ が格納される
U(NLB*NB)	Complex*16	WORK	行列ベクトル積の最終集計を行う作業領域
U1(NLB*NB)	Complex*16	WORK	行列ベクトル積の列方向部分積 UL1 を集計する作業領域
U2(NLB*NB)	Complex*16	WORK	行列ベクトル積の行方向部分積 UL2 を集計する作業領域
UL1(NLB*NB)	Complex*16	WORK	行列ベクトル積の列方向部分積を格納する作業領域
UL2(NLB*NB)	Complex*16	WORK	行列ベクトル積の行方向部分積を格納する作業領域
C1(NB)	Complex*16	WORK	PIVOT 内積部分積 CL1 を集計するための作業領域
C2(NB)	Complex*16	WORK	PIVOT 内積部分積 CL2 を集計するための作業領域
CL1(NB)	Complex*16	WORK	PIVOT 内積 $\langle Q^T \cdot P \rangle$ の部分積を格納する作業領域
CL2(NB)	Complex*16	WORK	PIVOT 内積 $\langle P^T \cdot P \rangle$ の部分積を格納する作業領域
LDIAG(NPROW)	Integer	INPUT	対角ブロックを保持する MPI プロセスリスト
MPI_COMM_ROW	Integer	INPUT	行方向の MPI 通信を行うためのコミュニケーター
MPI_COMM_COL	Integer	INPUT	列方向の MPI 通信を行うためのコミュニケーター
MPI_COMM_DIG	Integer	INPUT	対角ブロックを保持する MPI プロセスで通信を行うためのコミュニケーター

## 2. 5. j5ranknb 機能概要

j5ranknb は列 PIVOT ブロック PQ と行 PIVOT ブロック QTPT を使用し, 部分行列 A に対して RANK-NB 更新を行う機能です。j5ranknb の引数仕様を表 2-6 に示します。

表 2-6 j5ranknb 引数仕様

引数	型	種別	引数の説明
A(LDA,*)	Complex*16	INPUT / OUTPUT	(INPUT) : 部分行列 A (OUTPUT) : 部分行列 A に RANK-NB 更新を適用した結果
LDA	Integer	INPUT	部分行列 A の整合寸法 ( $LDA \geq NLB * NB$ )
NB	Integer	INPUT	ブロックサイズ ( $NB \geq 8$ , かつ $MOD(NB,2)=0$ )
NLB	Integer	INPUT	MPI プロセスが保持する部分行列 A のブロック数
JBLSTART2(NLB)	Integer	INPUT	部分行列 A の行方向開始位置
IPCOL	Integer	INPUT	プロセスマップ中の MPI プロセス列番号
KTMP	Integer	INPUT	部分行列 A の列方向開始位置
KBPIV	Integer	INPUT	列 PIVOT ブロックを保持する MPI プロセスの列番号
PQ(2*NB,NLB*NB)	Complex*16	INPUT	列 PIVOT ブロック P, Q
QTPT(2*NB,NLB*N B)	Complex*16	INPUT	行 PIVOT ブロック $Q^T$ , $P^T$



## 2. 6. j5parank 機能概要

j5parank は係数行列の最終ブロックが 1 となる場合に動作します。動作するプロセスが保持する PIVOT ブロックの K 列に rank-2 更新を行い、部分行列 A, 3 重対角行列の対角要素 DBUF, 副対角要素 EBUF を更新する機能です。j5parank の引数仕様を表 2-7 に示します。

表 2-7 j5parank 引数仕様

引数	型	種別	引数の説明
A(LDA,*)	Complex*16	INPUT / OUTPUT	(INPUT) : 部分行列 A (OUTPUT) : ブロック内の K 列について RANK-2 更新を適用した結果
LDA	Integer	INPUT	部分行列 A の整合寸法 ( $LDA \geq NLB * NB$ )
NB	Integer	INPUT	ブロックサイズ ( $NB \geq 8$ , かつ $MOD(NB,2)=0$ )
NLB	Integer	INPUT	MPI プロセスが保持する部分行列 A のブロック数
JBLSTART2(NLB)	Integer	INPUT	部分行列 A の行方向開始位置
K	Integer	INPUT	ブロック内の部分ハウスホルダ変換対象列番号
NPROW	Integer	INPUT	プロセスマップの行方向 MPI プロセス数
IPROW	Integer	INPUT	プロセスマップ中の MPI プロセス行番号
IPCOL	Integer	INPUT	プロセスマップ中の MPI プロセス列番号
KBLK	Integer	INPUT	部分行列 A の開始ブロック番号
KBPIV	Integer	INPUT	列 PIVOT ブロックを保持する MPI プロセスの列番号
DBUF(NLB*NB)	Complex*16	OUTPUT	各プロセスが保持する 3 重対角行列の対角要素
EBUF(NLB*NB)	Complex*16	OUTPUT	各プロセスが保持する 3 重対角行列の副対角要素
PQ(2*NB,NLB*NB)	Complex*16	INPUT	列 PIVOT ブロック P, Q
QTPT(2*NB,NLB*NB)	Complex*16	INPUT	行 PIVOT ブロック $Q^T$ , $P^T$
MPI_COMM_ROW	Integer	INPUT	行方向の MPI 通信を行うためのコミュニケーター

## 2. 7. j5serial 機能概要

j5serial は係数行列の最終対角ブロックを保持するプロセスのみが動作します。動作するプロセスが保持する最終対角ブロックにハウスホルダ変換を適用し、部分行列 A, 3 重対角行列の対角要素 DBUF, 副対角要素 EBUF と鏡像変換係数 TBUF を更新する機能です。j5serial の引数仕様を表 2-8 に示します。

表 2-8 j5serial 引数仕様

引数	型	種別	引数の説明
A(LDA,*)	Complex*16	INPUT / OUTPUT	(INPUT) : 部分行列 A (OUTPUT) : 最終ブロックにハウスホルダ変換を適用した結果
LDA	Integer	INPUT	部分行列 A の整合寸法 ( $LDA \geq NLB * NB$ )
NB	Integer	INPUT	ブロックサイズ ( $NB \geq 8$ , かつ $MOD(NB,2)=0$ )
NLB	Integer	INPUT	MPI プロセスが保持する部分行列 A のブロック数
KS	Integer	INPUT	部分行列 A の最終ブロック始点
NMOD	Integer	INPUT	係数行列の最終ブロック要素数
DBUF(NLB*NB )	Complex*16	OUTPUT	各プロセスが保持する 3 重対角行列の対角要素
EBUF(NLB*NB )	Complex*16	OUTPUT	各プロセスが保持する 3 重対角行列の副対角要素
TBUF(NLB*NB )	Complex*16	OUTPUT	各プロセスが保持するハウスホルダ変換の鏡像変換係数

## 2. 8. j5rngcom 機能概要

j5rngcom は対角ブロックを保持するプロセスのみで動作します。j5rngcom が動作するプロセスが部分的に保持する 3 重対角行列の対角要素 DBUF, 副対角要素 EBUF と鏡像変換係数 TBUF から, 次元数  $N$  の 3 重対角要素  $D$ , 副対角要素  $E$ , 鏡像変換係数  $TAU$  を作成する機能です。j5rngcom の引数仕様を表 2-9 に示します。

表 2-9 j5rngcom 引数仕様

引数	型	種別	引数の説明
N	Integer	INPUT	全体行列の次元数。 $N \geq 2$
A(LDA,*)	Complex*16	INPUT	ハウスホルダ変換を適用した部分行列
LDA	Integer	INPUT	部分行列 A の整合寸法 ( $LDA \geq NLB * NB$ )
D(N)	Complex*16	OUTPUT	3 重対角行列の対角成分
E(N)	Complex*16	OUTPUT	3 重対角行列の副対角成分
TAU(N)	Complex*16	OUTPUT	ハウスホルダ変換の鏡像変換係数
NB	Integer	INPUT	ブロックサイズ ( $NB \geq 8$ , かつ $MOD(NB,2)=0$ )
NLB	Integer	INPUT	MPI プロセスが保持する部分行列 A のブロック数
NPROW	Integer	INPUT	プロセスマップの行方向 MPI プロセス数
NPCOL	Integer	INPUT	プロセスマップの列方向 MPI プロセス数
IPROW	Integer	INPUT	プロセスマップ中の MPI プロセス行番号
IPCOL	Integer	INPUT	プロセスマップ中の MPI プロセス列番号
KBPIV	Integer	INPUT	係数行列の最終ブロックを保持する MPI プロセスの列番号
NMOD	Integer	INPUT	係数行列の最終ブロック要素数
IDIAG	Integer	INPUT	対角ブロックを保持する MPI プロセス番号
DBUF(NLB*NB)	Complex*16	INPUT	各プロセスが保持する 3 重対角行列の対角要素
EBUF(NLB*NB)	Complex*16	INPUT	各プロセスが保持する 3 重対角行列の副対角要素
TBUF(NLB*NB)	Complex*16	INPUT	各プロセスが保持するハウスホルダ変換の鏡像変換係数
DSEN(NLB*NB)	Complex*16	WORK	RING 通信で DBUF を送信するための作業領域
ESEN(NLB*NB)	Complex*16	WORK	RING 通信で EBUF を送信するための作業領域
TSEN(NLB*NB)	Complex*16	WORK	RING 通信で TBUF を送信するための作業領域
MPI_COMM_IG	Integer	INPUT	対角ブロックを保持する MPI プロセスで通信を行うためのコミュニケータ

### 3. 固有ベクトルのハウスホルダ逆変換プログラム PZSYGHR

固有ベクトルのハウスホルダ逆変換プログラム PZSYGHR は、QR 法による固有値・固有ベクトル計算において必要となる初期固有ベクトルを作成するために、単位行列にハウスホルダ逆変換を施すプログラムです。PZSYGHR は 3 つのサブルーチンで構成されており、3. 1. 節以降で各サブルーチンの機能について説明します。

#### 3. 1. PZSYGHR 機能概要

PZSYGHR は固有ベクトルのハウスホルダ逆変換プログラムのインターフェイスです。PZSYTRD で作成されるリフレクタ情報 **A** とハウスホルダ変換の鏡像変換係数 **TAU** を入力すると、**V** に固有ベクトルのハウスホルダ逆変換結果が出力されます。

PZSYGHR の引数仕様を表 3-1 に示します。表 3-1 中の **NB**, **NLB**, **NP**, **NPROW**, **JVNUM** は以下の式で定義します。

- $NB = DESCA(6)$
- $NLB = (N - 1) / NB + 1$
- $NP = MPI$  プロセス数
- $NPROW =$  プロセスマップの 1 辺の長さ
- $JVNUM = (N - 1) / NP + 1$

表 3-1 PZSYGHR 引数仕様

引数	型	種別	引数の説明
UPLO	Character*1	INPUT	入力行列の格納形式。UPLO='L'のみ指定可能
N	Integer	INPUT	全体行列の次元数。N $\geq$ 2
A(*)	Complex*16	INPUT	3 重対角化で使したリフレクタ情報。PZSYTRD の結果を入れればよい
IA	Integer	INPUT	部分行列 A の行方向先頭アドレス。IA=1 のみ指定可能
JA	Integer	INPUT	部分行列 A の列方向先頭アドレス。JA=1 のみ指定可能
DESCA(9)	Integer	INPUT	ディスクリプタ <ul style="list-style-type: none"> <li>DESCA(6) 部分行列 A のブロックサイズ NB NB<math>\geq</math>8 かつ MOD(NB,2)=0</li> <li>DESCA(9) 部分行列 A の整合寸法 DESCA(9)<math>\geq</math>NLB*NB</li> <li>DESCA(1)~DESCA(5), DESCA(7), DESCA(8) 値入力の必要なし</li> </ul>
TAU(N)	Complex*16	INPUT	ハウスホルダ変換の鏡像変換係数。PZSYTRD の結果を入れればよい
V(N,*)	Complex*16	OUTPUT	固有ベクトル逆変換結果
WORK	Complex*16	WORK	値入力の必要なし (作業領域)
LWORK	Integer	INPUT	複素数型作業領域配列 WORK のサイズ LWORK $\geq$ NB*NPROW*NLB*NB+2*NLB*NB*NB+4*JVNUM
INFO	Integer	OUTPUT	リターンコード
MAJOR	Character*1	INPUT	プロセスマップ方向。MAJOR='C'または MAJOR='R'のみ指定可能

表 3-2 PZSYGHR リターンコード

値	内容
0	正常終了したことを表す
-1	UPLO='L'が設定されていないことを表す
-2	N $\geq$ 2 が設定されていないことを表す
-4	IA=1 が設定されていないことを表す
-5	JA=1 が設定されていないことを表す
-66	DESCA(6) $\geq$ 8 かつ MOD(DESCA(6),2)=0 が設定されていないことを表す
-69	DESCA(9) $\geq$ NLB*NB が設定されていないことを表す
-10	LWORK $\geq$ NB*NPROW*NLB*NB+2*NLB*NB*NB+4*JVNUM が設定されていないことを表す
-12	MAJOR='C'または MAJOR='R'が設定されていないことを表す
100	プロセスマップが正方かつ 1 辺が 2 べき数でないことを表す

## 3. 2. j5inivec 機能概要

j5inivec はベクトル  $V$  を入力するとブロック列分割された単位行列を出力する機能です。j5inivec の引数仕様を表 3-3 に示します。

表 3-3 j5inivec 引数仕様

引数	型	種別	引数の説明
N	Integer	INPUT	全体行列の次元数。 $N \geq 2$
V(N,*)	Complex*16	OUTPUT	ブロック列分割された単位行列
JVBLK(NP+1 )	Integer	INPUT	MPI プロセスが担当する固有ベクトルの開始位置
NP	Integer	INPUT	MPI プロセス数
IP	Integer	INPUT	MPI プロセス番号
JVS	Integer	INPUT	ブロック列分割された固有ベクトルの始点
JVE	Integer	INPUT	ブロック列分割された固有ベクトルの終点

### 3. 3. j5makvec 機能概要

j5makvec は PZSYTRD で出力されるリフレクタ情報 A, ハウスホルダ変換の鏡像変換係数 TAU とブロック列分割された単位行列 V を入力すると, V にハウスホルダ逆変換を適用した結果を出力する機能です。j5makvec の引数仕様を表 3-3 に示します。

表 3-4 j5makvec 引数仕様

引数	型	種別	引数の説明
N	Integer	INPUT	全体行列の次元数。N $\geq$ 2
A(LDA,*)	Complex*16	INPUT	3 重対角化で使用したリフレクタ情報。PZSYTRD の結果を入れればよい
LDA	Integer	INPUT	部分行列 A の整合寸法 (LDA $\geq$ NLB*NB)
V(N,*)	Complex*16	INPUT / OUTPUT	(INPUT): ブロック列分割された単位行列。j5inivec の結果を入れればよい (OUTPUT): 単位行列にハウスホルダ逆変換を適用した結果
TAU(N)	Complex*16	OUTPUT	ハウスホルダ変換の鏡像変換係数。PZSYTRD の結果を入れればよい
JVBLK(NP+1)	Integer	INPUT	MPI プロセスが担当する固有ベクトルの開始位置
NP	Integer	INPUT	MPI プロセス数
IP	Integer	INPUT	MPI プロセス番号
NB	Integer	INPUT	ブロックサイズ (NB $\geq$ 8, かつ MOD(NB,2)=0)
NMOD	Integer	INPUT	係数行列の最終ブロック要素数
NPROW	Integer	INPUT	プロセスマップの行方向 MPI プロセス数
NPCOL	Integer	INPUT	プロセスマップの列方向 MPI プロセス数
IPROW	Integer	INPUT	プロセスマップ中の MPI プロセス行番号
IPCOL	Integer	INPUT	プロセスマップ中の MPI プロセス列番号
NLB	Integer	INPUT	MPI プロセスが保持する部分行列 A のブロック数
JVE	Integer	INPUT	ブロック列分割された固有ベクトルの終点
NGB	Integer	INPUT	係数行列の総ブロック数
JVNUM	Integer	INPUT	ブロック列分割された固有ベクトルの列数
UF(NB,NPROW*NLB*NB)	Complex*16	WORK	RING 通信で受信した部分リフレクタ情報 UREC からリフレクタ情報を作成するための作業領域
USEN(NB,NLB*NB)	Complex*16	WORK	RING 通信で部分リフレクタ情報を送信するための作業領域
UREC(NB,NLB*NB)	Complex*16	WORK	RING 通信で部分リフレクタ情報を受信するための作業領域
WK(4,JVNUM)	Complex*16	WORK	ハウスホルダ逆変換を行うための作業領域
MPI_COMM_ROW	Integer	INPUT	行方向の MPI 通信を行うためのコミュニケーター
MPI_COMM_COL	Integer	INPUT	列方向の MPI 通信を行うためのコミュニケーター