科学技術計算ライブラリ
ASL ユーザーズガイド
＜基本機能編 第１分冊＞

は し が き

本書は, 科学技術計算ライブラリ ASL (Advanced Scientific Library) の概念, 機能, 利用方法などについて説明したものです.
当製品に対応する説明書は7分冊からなっており, 構成は次のとおりです. このうち本書は, 基本機能第1分冊について記述したものです.

基本機能 第1分冊

章	タイトル	内　　　容
1	使用の手引き	本説明書の構成, 各項目の見方, および使用上の制限事項などの説明
2	格納モードの変換	配列データの格納モードの変換に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明
3	基本行列演算	行列の基本演算に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明
4	固有値・固有ベクトル	実行列, 複素行列, 実対称行列, エルミート行列, 実対称バンド行列, 実対称3重対角行列, 実対称スパース行列, エルミートスパース行列の標準固有値問題および実行列, 実対称行列, エルミート行列, 実対称バンド行列の一般化固有値問題に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明

基本機能 第2分冊

章	タイトル	内　　　容
1	使用の手引き	本説明書の構成, 各項目の見方, および使用上の制限事項などの説明
2	連立1次方程式 (直接法)	実行列, 複素行列, 正値対称行列, 実対称行列, エルミート行列, 実バンド行列, 正値対称バンド行列, 実3重対角行列, 実上三角行列, 実下三角行列の連立1次方程式に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明

基本機能 第 3 分冊

章	タイトル	内　容
1	使用の手引き	本説明書の構成, 各項目の見方, および使用上の制限事項などの説明
2	フーリエ変換とその応用	1 次元, 2 次元および 3 次元の複素ならびに実フーリエ変換, 1 次元, 2 次元および 3 次元の畳み込み, 相関, パワー・スペクトル解析, ウェーブレット変換およびラプラス逆変換に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明

基本機能 第 4 分冊

章	タイトル	内　容
1	使用の手引き	本説明書の構成, 各項目の見方, および使用上の制限事項などの説明
2	微分方程式とその応用	〔常微分方程式初期値問題〕 連立高階, 陰的連立, 行列型, スティフ問題の連立高階, 連立 1 階, 高階常微分方程式 〔常微分方程式境界値問題〕 連立高階, 連立 1 階, 高階, 線形高階, 線形 2 階常微分方程式 〔積分方程式〕 第 2 種フレドホルム型, 第 1 種ボルテラ型積分方程式 〔偏微分方程式〕 2 次元および 3 次元の非同次ヘルムホルツ方程式 に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明
3	数値微分	1 変数関数および多変数関数の数値微分に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明
4	数値積分	有限区間, 半無限区間, 全無限区間, 2 次元有限区間, 多次元有限区間の数値積分に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明
5	補間・近似	補間, 曲面補間, 最小二乗近似, 最小二乗曲面近似, チェビシェフ近似に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明
6	スプライン関数	3 次スプライン, 双 3 次スプラインおよび B–スプラインを用いた補間, 平滑化, 数値微分, 数値積分に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明

基本機能 第5分冊

章	タイトル	内　　容
1	使用の手引き	本説明書の構成, 各項目の見方, および使用上の制限事項などの説明
2	特殊関数	ベッセル関数, 変形ベッセル関数, 球ベッセル関数, ベッセル関数に関連した関数, ガンマ関数, ガンマ関数に関連した関数, 楕円関数, 初等関数の不定積分, ルジャンドル陪関数, 直交多項式, その他の特殊関数に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明
3	ソート・順位付け	ソート, 順位付けに関するサブルーチンの使用方法および使用例の説明
4	方程式の根	代数方程式, 非線形方程式, 連立非線形方程式の根に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明
5	極値問題・最適化	制約なし関数の極小化, 制約なし関数二乗和の極小化, 制約付き1変数関数の極小化, 制約付き多変数関数の最小化, 最短路問題に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明

基本機能 第6分冊

章	タイトル	内　　容
1	使用の手引き	本説明書の構成, 各項目の見方, および使用上の制限事項などの説明
2	乱数の検定	一様乱数の検定, 分布乱数の検定に関するサブルーチンの使用方法および使用例の説明
3	確率分布	連続分布, 離散分布に関するサブルーチンの使用方法および使用例の説明
4	基礎統計量	基礎統計量, 分散共分散, 相関係数に関するサブルーチンの使用方法および使用例の説明
5	推定と検定	区間推定, 検定に関するサブルーチンの使用方法および使用例の説明
6	分散分析・実験計画	1元配置, 2元配置, 多元配置, 乱塊法, グレコ・ラテン方格法, 累積法に関するサブルーチンの使用方法および使用例の説明
7	ノンパラメトリック検定	χ^2 分布による検定, その他分布による検定に関するサブルーチンの使用方法および使用例の説明
8	多変量解析	主成分分析, 因子分析, 正準相関分析, 判別分析, クラスタ分析に関するサブルーチンの使用方法および使用例の説明
9	時系列分析	自己相関・相互相関, 自己共分散・相互共分散, 平滑化・需要予測に関するサブルーチンの使用方法および使用例の説明
10	回帰分析	線形回帰, 非線形回帰に関するサブルーチンの使用方法および使用例の説明

共有メモリ並列機能

章	タイトル	内　　　　容
1	使用の手引き	本説明書の構成, 各項目の見方, および使用上の制限事項などの説明
2	基本行列演算	実行列および複素行列の積を求めるサブルーチンのアルゴリズム, 使用方法の説明
3	連立 1 次方程式 (直接法)	実行列, 複素行列, 実対称行列, エルミート行列の連立 1 次方程式（直接法）に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明
4	連立 1 次方程式 (反復法)	実正値対称スパース行列, 実対称スパース行列, 実非対称スパース行列の連立 1 次方程式（反復法）に関するサブルーチンのアルゴリズム, 使用法および使用例の説明
5	固有値・固有ベクトル	実対称行列およびエルミート行列の固有値問題に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明
6	フーリエ変換とその応用	1 次元, 2 次元および 3 次元の複素ならびに実フーリエ変換, 2 次元および 3 次元の畳み込み, 相関, パワー・スペクトル解析に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明
7	ソート	ソートに関するサブルーチンの使用方法および使用例の説明

2023 年　3 月　ASL 付属説明書　3.0.0-230301

備考 (1) 本書に説明しているすべての機能は, プログラムプロダクトであり, ASL 1.1 に対応しています.

(2) 製品名などの固有名詞は, 各メーカーの登録商標または商標です.

(3) 本ライブラリは, 最新の数値計算技法を取り入れ, 開発されたものです. 従って, 最新の技術を維持する目的から, 改良または新しく追加されたサブルーチンが, 既存のサブルーチンの機能を包含し, かつ, これまで以上の高速性能が得られる場合には, 既存のサブルーチンを削除することもあります.

目 次

第 3 章 基本行列演算 45

第 1 章　使用の手引

1.1　概　説

1.1.1　科学技術計算ライブラリ ASL の概要

科学技術計算ライブラリ ASL（Advanced Scientific Library）は, 数値解析プログラムの作成を強力に支援する数学ライブラリである．ASL では広範な数値解析分野で頻出するプログラムを提供しており, それらは VE(Vector Engine) 上で優れた実行速度と精度を実現するための高度な最適化が適用されている． ASL を用いることによって, 難解な数値計算アルゴリズムの詳細に煩わされることなく高度な数値解析プログラムを作成することができ, 数値解析プログラム開発の生産性を大幅に改善することができる.

ASL は, 基本機能, 共有メモリ並列機能で構成される．機能分類と本マニュアルの分冊との対応を表 1−1 に示す.

表 1−1　ASL　の機能分類

機能分類	分冊
基本機能	第 1〜6 分冊
共有メモリ並列機能	第 7 分冊

1.1.2　ASL　の特長

ASL の特長は, 次のとおりである.

(1) ハードウェア性能を十分発揮できるように設計しており, コンパイラの最適化機能を用いて作成した.

(2) 行列を扱うサブルーチンでは, 行列の種類 (対称行列, エルミート行列など) に応じて最適に処理を行えるように, 専用のサブルーチンをそれぞれ提供している．一般に, 専用のサブルーチンを用いて処理を行った方が, 処理性能を向上したり, 必要なメモリ容量を節約したりすることができる.

(3) 処理手順に従ってモジュール化を行い, コンポーネントサブルーチンごとの信頼性向上に努めるとともに, システム全体の効率化, 信頼性向上を図った.

(4) サブルーチンを利用した後の エラーインディケータの番号が体系的に決めてあるので, エラー情報を把握しやすい.

1.2　ライブラリの種類

ASL には, 32 ビット整数型ライブラリと 64 ビット整数型ライブラリがある. 32 ビット整数型ライブラリに含まれるサブルーチンの整数型の引数は, 32 ビット (4 バイト) 整数型である. 一方, 64 ビット整数型ライブラリに含まれるサブルーチンの整数型の引数は, 64 ビット (8 バイト) 整数型である. また, サブルーチンの実数型の引数によってサブルーチン名が異なる. サブルーチン名については, 1.4 を参照のこと.

表 1–2　ASL で提供しているライブラリの種類

変数の大きさ (バイト)		引数の型宣言文	通称	ライブラリの種類
整数型	実数型			
4	8	INTEGER(4) REAL(8)	32 ビット整数型倍精度 サブルーチン	32 ビット整数型ライブラリ (リンクオプション：-lasl_sequential)
4	4	INTEGER(4) REAL(4)	32 ビット整数型単精度 サブルーチン	
8	8	INTEGER(8) REAL(8)	64 ビット整数型倍精度 サブルーチン	64 ビット整数型ライブラリ (リンクオプション：-lasl_sequential_i64)
8	4	INTEGER(8) REAL(4)	64 ビット整数型単精度 サブルーチン	

（注 1）機能によっては, 4 種類全てをサポートしているとは限らない. その場合, 個別の説明の注意事項の欄に記述するので注意されたい.

（注 2）INTEGER(4) および REAL(4) で型宣言する場合, “(4)” は省略可.

1.3 マニュアルについて

ここでは本マニュアルの第 2 章以降の構成について述べる.
第 2 章以降は ASL で用いられるサブルーチンとその機能, 使用方法の説明を行う.

1.3.1 『概　要』

各章の第 1 節では, 概要として各サブルーチンの効果的な使用法, 採用した手法およびそのアルゴリズム, 注意事項などについて述べてある.

1.3.2 サブルーチン説明文の構成

各章の第 2 節では, サブルーチンごとに以下の順で説明している.

(1) 機能

(2) 使用法

(3) 引数

(4) 制限条件

(5) エラーインディケータ

(6) 注意事項

(7) 使用例

各項目は次に述べる原則に従って記述されている.

1.3.3 各項目の内容

(1) **機 能**

この項目では, サブルーチンの目的とする機能について簡単に述べてある.

(2) **使用法**

この項目では, サブルーチン名とその引数の順序について記述してある.
引数の並べ方は, 原則として次のように決められている.

CALL サブルーチン名 (入力引数, 入出力引数, 出力引数, ISW, ワーク, IERR)

ここで, ISW は処理の手順を指定するための入力引数であり, IERR は エラーインディケータである. ただし, 入力引数と入出力引数の順序が逆の場合もある. さらに次の規則にしたがっている.

- 配列は重要度に応じてできるだけ左方によせる.
- 配列名に続けて配列の大きさをそえる. 同じ大きさをもつ配列が複数個あるときは, その最初の配列名に続けてその大きさを引数として与え, 2 番目以降の配列からは, その大きさは引数として与えない.

(3) 引 数

(2) 項で記述された引数について, 順番に説明されている. その形式は以下のように統一されている.

引数	**型**	**大きさ**	**入出力**	**内 容**
(a)	(b)	**(c)**	**(d)**	**(e)**

(a) 引数

引数が記載されている.

(b) 型

引数のデータの型を示す. 次の略記号のいずれかに示されている.

I ：整数型

D ：倍精度実数型

R ：単精度実数型

Z ：倍精度複素数型

C ：単精度複素数型

整数型の引数には 64 ビット整数型と 32 ビット整数型とがある. サブルーチンの整数型引数が 64 ビット整数型であるのか 32 ビット整数型であるのかは, そのサブルーチンが 64 ビット整数型であるか 32 ビット整数型であるか, つまりライブラリの種類によって決められる (1.2 参照). ユーザプログラムにおいて引数の型を宣言する際は, 32 ビット整数型の引数は`INTEGER(4)`, 64 ビット整数型の引数は`INTEGER(8)` を用いて宣言する必要がある.

(c) 大きさ

指定された引数の必要な大きさを示す. 2 以上を指定した場合には, このサブルーチンを利用したプログラム側で, その必要な領域を確保しなければならない.

1 ：変数であることを示す.

N ：要素が N 個の 1 次元配列であることを示す. この配列が指定された直後にその大きさを示す引数 N が定義される. ただし大きさ N が以前に定義された配列の大きさを規定している場合には省略される. このほかに数値のみにて指定する場合や, $3 \times \mathrm{N}$ や $\mathrm{N} + \mathrm{M}$ のように, 積または和の形で表記する場合もある.

M, N ：M 行 N 列の 2 次元の配列であることを示す. この配列が指定される前にこの M と N が定義されていない場合は, この配列の直後にその大きさを示す引数 M または N が定義される.

(d) 入出力

引数の内容説明が入力時であるか出力時であるかを示す.

i. 「入力」とだけある場合：

このサブルーチンを利用したプログラムに制御がもどったときに, 引数の入力時の情報は保存されている. 入力時の情報は特に断らない限り, 利用者が与えなければならない.

ii. 「出力」とだけある場合：

引数には, サブルーチン内で計算された結果が出力される. 入力時には何も入れなくてよい.

iii. 「入力」と「出力」の両方に説明がある場合：

サブルーチンに制御がわたる前とサブルーチンから制御がもどった後で, この引数の内容に変化がある場合である. 入力時の情報は特に断らない限り, 利用者が与えなければならない.

iv. 「ワーク」とある場合：

サブルーチン内で演算を行うときに利用する領域であること示す. サブルーチンを利用するプログラム側で, 指定された大きさの作業領域を確保しなければならない. なお, 次の計算に流用するために, 作業領域の内容を保存しておく必要がある場合がある.

(e) 内　容

入力時あるいは出力時に, 引数が保持している情報について説明される.

- 「引　数」の説明の例を次に示す.

例 実行列の LU 分解と条件数を求めるサブルーチン (DBGMLC, RBGMLC) の使用法は以下のとおりである.

倍精度サブルーチン:

CALL DBGMLC (A, LNA, N, IPVT, COND, W1, IERR)

単精度サブルーチン:

CALL RBGMLC (A, LNA, N, IPVT, COND, W1, IERR)

この場合の引数の説明は次のようになる.

表 1–3　引数の例

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$

項番	引数名	型	大きさ	入出力	内　　容
1	A	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}^{注}$	LNA, N	入 力	実行列 A(2 次元配列型)
				出 力	$A = LU$ と分解した時の単位上三角行列 U および下三角行列 L
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A の次数 n
4	IPVT	I	N	出 力	ピボッティング情報 IPVT(i): i 段目の処理において行 i と交換した行の番号
5	COND	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	1	出 力	条件の逆数
6	W1	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	N	ワーク	作業領域
7	IERR	I	1	出 力	エラーインディケータ

このサブルーチンを利用するには, まず, 引数として使用する配列 A, IPVT および W1 を, 呼び出し元の利用者プログラム側でアロケートする必要がある. それらはそれぞれ, $\left\{\begin{matrix}\text{倍精度}\\\text{単精度}\end{matrix}\right\}^{注}$ 実数型で大きさ (LNA, N), 整数型で大きさ N, $\left\{\begin{matrix}\text{倍精度}\\\text{単精度}\end{matrix}\right\}$ 実数型で大きさ N の配列である.

また, 64 ビット整数版を利用する場合には, 整数型引数 (LNA,N,IPVT,IERR) はすべて `INTEGER` ではなく `INTEGER(8)` を用いて宣言する必要がある.

注 DBGMLC のときには倍精度実数型 (略記号 D), RBGMLC のときには実数型 (略記号 R) で宣言することを意味する. 以下, 本文中で特に断らない限り中括弧 {} 等の使用法は, 同様の扱いとする.

このサブルーチンを使用するときには, A, LNA および N にデータを格納しておかなければならない. サブルーチン内では, 与えられた行列の LU 分解と条件数の算出が行われ, 結果が配列 A と変数 COND に格納される. また, 後続サブルーチンで利用するため, ピボッティング情報が IPVT に格納される.

IERR は, 入力データや処理途中の異常を利用者に知らせるための引数であり, 正常の場合は 0 にセットされる.

なお, W1 はサブルーチン内でのみ使用する作業領域であるので, 入力時および出力時の内容は特に意味をもたない.

(4) **制限条件**

サブルーチンの引数の制限範囲を明確にしてある.

(5) **エラーインディケータ**

各サブルーチンには, エラーインディケータが出力引数として設けられている. この エラーインディケータは, IERR という変数名に統一されており, 引数表の最後におかれている. 各サブルーチンはサブルーチン内でエラー検出を行い, その結果を IERR に設定する. IERR の値の意味は, 次の 5 段階に分かれている.

表 1–4 エラーインディケータの出力値区分

レベル	IERR の値	意　　味	処 理 内 容
正 常	0	正常終了した.	結果は保証される.
警 告	1000～2999	ある条件のもとで一応の処理が終了した.	条件付きで結果は保証される.
異 常	3000～3499	引数が制限条件に違反したために処理が打ち切られた.	結果は保証されない.
	3500～3999	得られた結果がある検定条件を満足しなかった.	得られた結果を返す (結果は保証されない).
	4000 以上	処理の途中で致命的なエラーが発見された. 通常は処理を打ち切る.	結果は保証されない.

(6) **注意事項**

サブルーチンを使用するときの注意点およびあいまいな点を明確にしてある.

(7) **使用例**

サブルーチンの使い方の一例を載せてある. なお複数のサブルーチンを組み合わせて一つの例としてある場合もあるので注意されたい. 出力結果は, 32 ビット整数版での結果であり, コンパイラや組み込み関数の変更などにより丸め誤差の範囲で異なる場合がある.

本説明書に記載されている使用例のプログラムはソースコードの形で「ASL ユーザーズガイド」に収録されている. 入力データも (もし存在する場合は)「ASL ユーザーズガイド」に収録されている. コンパイラを用いて使用例のソースコードから実行形式ファイルを作成する場合には, ライブラリ本体とリンクする必要がある.

1.4　サブルーチン名

ASL の基本機能のサブルーチン名は, 6 桁のアルファニューメリック記号の集まりである.
また, サブルーチン名の各記号にはそれぞれ意味を持ち, 図 1–1 で表される. 利用時においては, 計算用途に合わせてサブルーチン名を指定する必要がある.

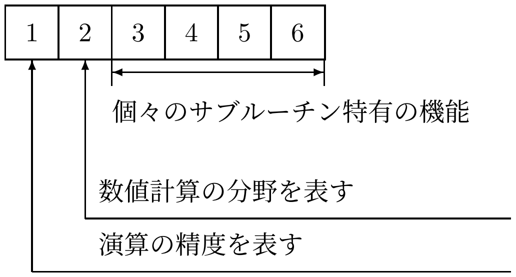


図 1–1　サブルーチン名の構成要素

図 1–1 の “1” ： 演算の精度を表す. 基本機能編で使用される文字は, 次の 8 種類である.

D, W	倍精度実数型演算
R, V	単精度実数型演算
Z, J	倍精度複素数型演算
C, I	単精度複素数型演算

ただし, 上記の複素数型とは必ずしも引数の型が複素数型であることを意味しない.

図 1–1 の “2” ： 計算の分野を表す. 現在, ASL では次の文字が使用されている.

文字	計算の分野	分冊
A	格納モードの変換	1
	基本行列演算	1, 7
B	連立 1 次方程式 (直接法)	2, 7
C	固有値・固有ベクトル	1, 7
F	フーリエ変換とその応用	3, 7
	時系列分析	6
G	スプライン関数	4
H	数値積分	4
I	特殊関数	5
J	乱数の検定	6
K	常微分方程式初期値問題	4
L	方程式の根	5
M	極値問題・最適化	5
N	近似・回帰分析	4, 6
O	常微分方程式境界値問題, 積分方程式, 偏微分方程式	4
P	補間	4
Q	数値微分	4

文字	計算の分野	分冊
S	ソート・順位付け	5, 7
X	基本行列演算	1
	連立1次方程式 (反復法)	7
1	確率分布	6
2	標本統計	6
3	推定と検定	6
4	分散分析・実験計画	6
5	ノンパラメトリック検定	6
6	多変量解析	6

図 1–1 **の “3” ～ “6”：** これらの文字で, 個々のサブルーチンに特有の機能を表す.

1.5 注意事項

(1) 単精度版ではなく, 倍精度版を標準として利用する方がよい. 精度が高いことに加え, 倍精度版の方が単精度版に比べて安定的に解が求まる場合 (特に固有値・固有ベクトル) が多い.

(2) 演算例外の抑止はメインプログラム側で行う必要がある. ASL のサブルーチンでは, コンパイラの演算例外の抑止に関して, ユーザのメインプログラムのコンパイルパラメータの指示に従うように設定してある.

(3) 扱う演算桁数を越える精度を期待することはできない. たとえば倍精度演算の (仮数部の) 演算桁数は 10 進 15 桁程度であるが, ここで数学的に 1 となるような値を計算した場合, 10^{-15} 程度の誤差は必ず発生する. これを抑制する方法として, 任意桁数演算のような多倍長演算のエミュレートが考えられるが, この場合, たとえば円周率のような定数や関数近似の定数なども都度計算する必要が生じるので, 通常の演算と比較して計算効率は悪くなる.

(4) 数学的に解が存在しないような問題の解を得ることはできない. たとえば, 数学的に特異な (または特異に近い) 行列を係数に持つ連立 1 次方程式の解を精度良く求めることは原理的にできない. なお, 数値計算上は, 数学的に特異な行列と特異に近い行列とを厳密に区別することはできない. もちろん, たとえば, 条件数の計算値が設定した基準値以上であれば特異とみなすというようなことはいつでも可能である.

(5) 浮動小数点例外 (オーバフローなど) をおこすようなデータを与えた場合, 正常な計算結果を期待することはできない. ただし, 反復計算で残差の加算等を行った場合に発生する浮動小数点アンダフローなどはこの限りではない.

(6) 数値計算で扱う問題 (特に反復法を計算手法とする問題) では, 与えるデータによっては解が精度良く求められない場合や全く求まらない場合がある. このような場合は, 問題自体を見直して, 解が求まるような問題に変更するなどの処置を講じる必要がある. たとえば, スパース行列を係数とする連立 1 次方程式を解く場合に, 専用のサブルーチンで解が得られないときでも, 密行列用のサブルーチンを用いることで解が得られる場合がある.

(7) 解が複数ある問題を解く場合, 実行するマシンや OS, 用いるコンパイラ等で実行結果が見掛け上異なる場合がある. たとえば, 固有値問題を解いた場合に得られる固有ベクトルがこれに相当する.

(8) “[非推奨]” と表示のあるサブルーチンは, 今後廃止予定の機能である. より高速な実装が **ASL 統合インタフェース**で提供されているので, そちらを利用されたい.

第 2 章　格納モードの変換

2.1　概 要

本章では行列の格納モードの変換を行うサブルーチンについて説明する.

本ライブラリにおいては, 行列の型, 性質ごとに異なった格納形式を採用しているため, 利用者は使用するサブルーチンに対応した格納形式で行列を格納しておかなければならない. 行列がすでに配列に格納されている場合には, 格納形式を変更する必要がある. この変換を容易に行うために, 格納モード変換のサブルーチンを提供している.

2.1.1 使用しているアルゴリズム

2.1.1.1 実バンド行列の格納モードの変換

実バンド行列の i 行 j 列の要素は次のように格納される.

$$\begin{array}{ccc} \text{行列} & & \text{バンド型} \\ A_{i,j} & \longleftrightarrow & \mathrm{A}(j-i+\mathrm{ML}+1,i) \end{array}$$

備 考

a. ML は下バンド幅である.

2.1.1.2 実対称バンド行列の格納モードの変換

実対称バンド行列の i 行 j 列の要素は次のように格納される.

$$\begin{array}{ccc} \text{行列} & & \text{対称バンド型} \\ A_{i,j} & \longleftrightarrow & \mathrm{A}(i-j+\mathrm{MB}+1,j) \end{array}$$

備 考

a. MB はバンド幅である.

2.1.1.3 スパース行列の列方向 1 次元リスト型格納

スパース行列の i 行 j 列の要素は次のように格納される.

$$\begin{array}{ccl} \text{行列} & & \text{列方向 1 次元リスト型} \\ A_{i,j} & \longrightarrow & \left\{ \begin{array}{rcl} k & = & \mathrm{IPONTR}(j)+m \\ i & = & \mathrm{IRWIND}(k) \\ A_{i,j} & = & \mathrm{VALUES}(k) \end{array} \right. \end{array}$$

備 考

a. 非対称行列の場合
ITYPE = 1.
m は j 列中の非零要素の各々に付けられる, 0 から始まる通し番号である.

b. 行列が対称で, 上三角部分を入力する場合
ITYPE = 2.
m は上三角行列部分における j 列中の非零要素の各々に付けられる, 0 から始まる通し番号である.

c. 行列が対称で, 下三角部分を入力する場合
ITYPE = 2.
m は下三角行列部分における j 列中の非零要素の各々に付けられる, 0 から始まる通し番号である.

2.1.1.4 スパース行列の ELLPACK 型格納

スパース行列の i 行 j 列の要素は次のように格納される.

$$\begin{array}{ccl} \text{行列} & & ELLPACK\ \text{型} \\ A_{i,j} & \longrightarrow & \left\{ \begin{array}{rcl} A_{i,j} & = & \mathrm{A}(i,m) \\ j & = & \mathrm{JA}(i,m) \end{array} \right. \end{array}$$

備 考

a. m は, 行列の i 行中の非零要素の各々に付けられる, 1 から始まる通し番号である. 最小値 $m=1$ は常に対角要素に付けるものとする. 対角要素以外の要素については値 $m=2,3,\cdots$ を付ける順序は任意でよい.

2.2 格納モードの変換

2.2.1 DABMCS, RABMCS

実バンド行列の格納モードの変換：(2 次元配列型) から (バンド型) へ

(1) **機 能**

実バンド行列 A の格納形式を (2 次元配列型) から (バンド型) に変換する.

(2) **使用法**

倍精度サブルーチン:

CALL DABMCS (A, LNA, N, MU, ML, B, LMB, IERR)

単精度サブルーチン:

CALL RABMCS (A, LNA, N, MU, ML, B, LMB, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型　I: $\left\{\begin{array}{l}\text{32 ビット整数版では INTEGER(4)}\\ \text{64 ビット整数版では INTEGER(8)}\end{array}\right\}$

R:単精度実数型　C:単精度複素数型

項番	引数名	型	大きさ	入出力	内容
1	A	$\left\{\begin{array}{c}\text{D}\\ \text{R}\end{array}\right\}$	LNA, N	入 力	実バンド行列 A(2 次元配列型)(付録 B 参照)
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A の次数
4	MU	I	1	入 力	行列 A の上バンド幅
5	ML	I	1	入 力	行列 A の下バンド幅
6	B	$\left\{\begin{array}{c}\text{D}\\ \text{R}\end{array}\right\}$	LMB, N	出 力	実バンド行列 A(バンド型)(付録 B 参照)
7	LMB	I	1	入 力	配列 B の整合寸法
8	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 \leq \mathrm{MU} < \mathrm{N}$

(b) $0 \leq \mathrm{ML} < \mathrm{N}$

(c) $0 < \mathrm{N} \leq \mathrm{LNA}$

(d) $\mathrm{MU} + \mathrm{ML} < \mathrm{LMB}$

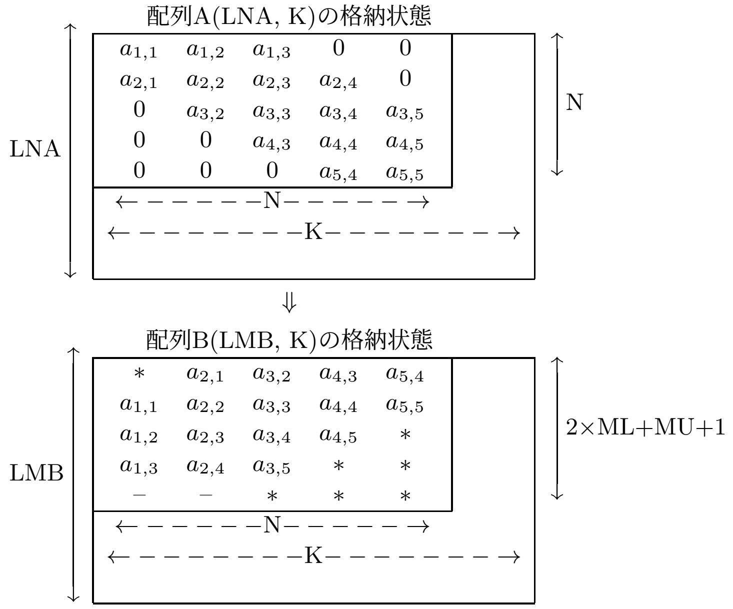
(5) **エラーインディケータ**

IERR の値	意味	処理内容
0	正常終了.	
1000	$\mathrm{N} = 1$ であった.	処理を続ける.
3000	制限条件 (a), (b), (c) または (d) を満足しなかった.	処理を打ち切る.

(6) **注意事項**

(a) 配列 B の要素のうち, 行列 A の要素に対応しない要素は, 引用時の値がそのまま保存される.

例



備　考

a. $*$ と–は, 入力時の値を保つ.
b. –は, 行列の LU 分解時に必要となる領域である.
c. MU は上バンド幅, ML は下バンド幅である.
d. LMB $>$ ML $+$ MU, K $\geq$ N を満たさなければならない. (ただし, 変換後 LU 分解を行う場合には, LMB $\geq 2 \times$ ML $+$ MU $+ 1$, K $\geq$ N).

(b) 変換後に LU 分解を行う場合には, LMB が次の条件を満たすように配列 B を宣言しておく必要がある.

$$\mathrm{LMB} \geq \min(2 \times \mathrm{ML} + \mathrm{MU} + 1, \mathrm{N} + \mathrm{ML})$$

(7) **使用例**

実バンド行列 A の格納形式を (2 次元配列型) から (バンド型) に変換し, 変換した形式で行列を保持する配列 AC を用いて連立 1 次方程式 $A\boldsymbol{x} = \boldsymbol{b}$ の解を求める. (ただし, MU は上バンド幅, ML は下バンド幅である).

```
IMPLICIT REAL(8) (A-H, O-Z)
PARAMETER (LNA=11, LMB=11)
DIMENSION A(LNA, LNA), AC(LMB, LMB), B(LNA), IPVT(LNA)
 ≀
CALL DABMCS(A, LNA, N, MU, ML, AC, LMB, IERR)
 ≀
CALL DBBDSL(AC, LMB, N, MU, ML, B, IPVT, JERR)
 ≀
```

2.2.2 DABMEL, RABMEL
実バンド行列の格納モードの変換 : (バンド型) から (2 次元配列型) へ

(1) **機 能**

実バンド行列 A の格納形式を (バンド型) から (2 次元配列型) に変換する.

(2) **使用法**

倍精度サブルーチン:

CALL DABMEL (A, LMA, N, MU, ML, B, LNB, IERR)

単精度サブルーチン:

CALL RABMEL (A, LMA, N, MU, ML, B, LNB, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\left\{\begin{array}{l}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{array}\right\}$

項番	引数名	型	大きさ	入出力	内 容
1	A	$\left\{\begin{array}{c}D\\R\end{array}\right\}$	LMA, N	入 力	実バンド行列 A(バンド型)(付録 B 参照)
2	LMA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A の次数
4	MU	I	1	入 力	行列 A の上バンド幅
5	ML	I	1	入 力	行列 A の下バンド幅
6	B	$\left\{\begin{array}{c}D\\R\end{array}\right\}$	LNB, N	出 力	実バンド行列 A(2 次元配列型)(付録 B 参照)
7	LNB	I	1	入 力	配列 B の整合寸法
8	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 \leq \mathrm{MU} < \mathrm{N}$

(b) $0 \leq \mathrm{ML} < \mathrm{N}$

(c) $0 < \mathrm{N} \leq \mathrm{LNB}$

(d) $\mathrm{MU} + \mathrm{ML} < \mathrm{LMA}$

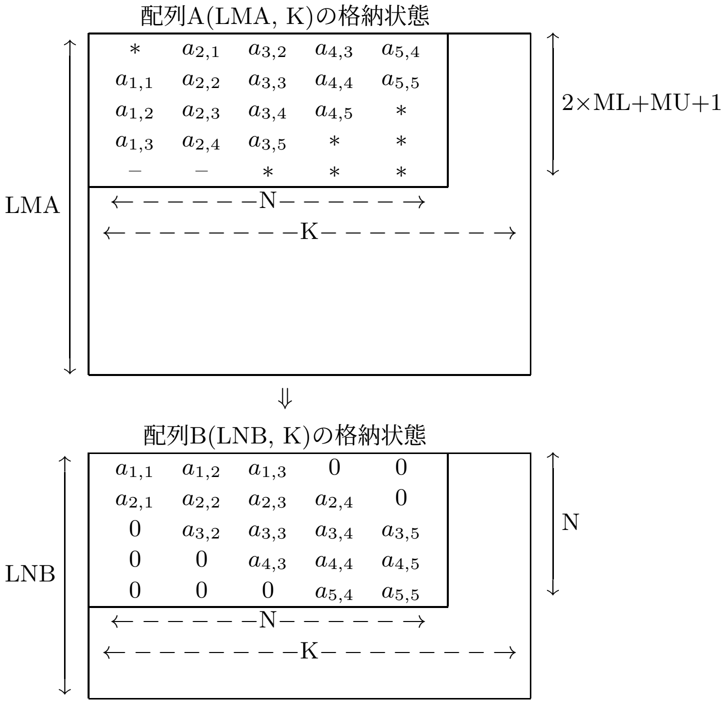
(5) **エラーインディケータ**

IERR の値	意 味	処 理 内 容
0	正常終了.	
1000	$\mathrm{N} = 1$ であった.	処理を続ける.
3000	制限条件 (a), (b), (c) または (d) を満足しなかった.	処理を打ち切る.

(6) 注意事項

(a) 変換後の行列のバンド幅よりも外側の部分には, 0.0 が入れられる.

例



備 考

a. ∗ は, 任意の値であることを示す.

b. –は, 行列の LU 分解時に必要となる領域である.

c. MU は上バンド幅, ML は下バンド幅である.

d. LMA $>$ ML $+$ MU, K $\geq$ N を満たさなければならない.

(7) 使用例

実バンド行列 A を (バンド型) で配列 AC に保持し, A を係数行列とする連立 1 次方程式 $A\boldsymbol{x} = \boldsymbol{b}$ の解を求め, A の LU 分解を (2 次元配列型) で配列 A に格納する. (ただし, MU は上バンド幅, ML は下バンド幅である).

```
IMPLICIT REAL(8) (A-H, O-Z)
PARAMETER (LNA=11, LMB=11)
DIMENSION A(LNA, LNA), AC(LMB, LMB), B(LNA), IPVT(LNA)
 ≀
CALL DBBDSL(AC, LMB, N, MU, ML, B, IPVT, JERR)
 ≀
CALL DABMEL(AC, LMB, N, MU, ML, A, LNA, KERR)
 ≀
```

2.2.3 DASBCS, RASBCS
実対称バンド行列の格納モードの変換：(2 次元配列型)(上三角型) から (対称バンド型) へ

(1) **機 能**

実対称バンド行列 A の格納形式を (2 次元配列型)(上三角型) から (対称バンド型) に変換する.

(2) **使用法**

倍精度サブルーチン:

CALL DASBCS (A, LNA, N, MB, B, LMB, IERR)

単精度サブルーチン:

CALL RASBCS (A, LNA, N, MB, B, LMB, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\left\{\begin{array}{l}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{array}\right\}$

項番	引数名	型	大きさ	入出力	内　容
1	A	$\left\{\begin{matrix}\text{D}\\\text{R}\end{matrix}\right\}$	LNA, N	入 力	実対称バンド行列 A(2 次元配列型)(上三角型) (付録 B 参照)
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A の次数
4	MB	I	1	入 力	行列 A のバンド幅
5	B	$\left\{\begin{matrix}\text{D}\\\text{R}\end{matrix}\right\}$	LMB, N	出 力	実対称バンド行列 A(対称バンド型) (付録 B 参照)
6	LMB	I	1	入 力	配列 B の整合寸法
7	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNA}$

(b) $0 \leq \mathrm{MB} < \mathrm{N}$

(c) $\mathrm{MB} < \mathrm{LMB}$

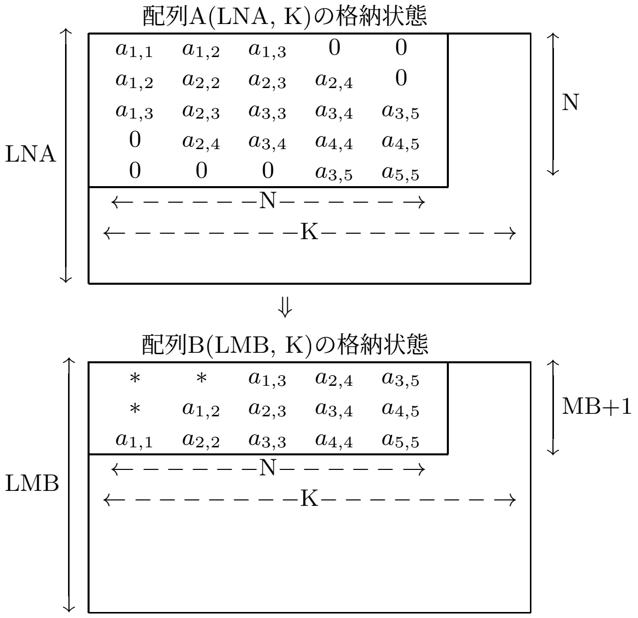
(5) **エラーインディケータ**

IERR の値	意　味	処 理 内 容
0	正常終了.	
1000	$\mathrm{N} = 1$ であった.	処理を続ける.
3000	制限条件 (a), (b) または (c) を満足しなかった.	処理を打ち切る.

(6) **注意事項**

(a) 行列 A の上三角部分のみが配列 B に格納される.

(b) 配列 B の要素のうち, 行列 A の要素に対応しない要素は, 引用時の値がそのまま保存される.
例



備　考
a.　$*$ は, 入力時の値を保つ.
b.　MB は, バンド幅である.
c.　LMB $>$ MB, K $\geq$ N を満たさなければならない.

(7) **使用例**

正値対称バンド行列 A の格納形式を (2 次元配列型)(上三角型) から (対称バンド型) に変換し, 変換した形式で行列を保持する配列 AC を用いて連立 1 次方程式 $A\boldsymbol{x} = \boldsymbol{b}$ の解を求める. ただし MB はバンド幅である.

```
IMPLICIT REAL(8) (A-H, O-Z)
PARAMETER (LNA=11, NC=11)
DIMENSION A(LNA, LNA), AC(LMB, LMB), B(NA)
  ≀
CALL DASBCS(A, LNA, N, MB, AC, LMB, IERR)
  ≀
CALL DBBPSL(AC, LMB, N, MB, B, JERR)
  ≀
```

2.2.4 DASBEL, RASBEL 実対称バンド行列の格納モードの変換：(対称バンド型) から (2 次元配列型)(上三角型) へ

(1) 機 能

実対称バンド行列 A の格納形式を (対称バンド型) から (2 次元配列型)(上三角型) に変換する.

(2) 使用法

倍精度サブルーチン:

CALL DASBEL (A, LMA, N, MB, B, LNB, IERR)

単精度サブルーチン:

CALL RASBEL (A, LMA, N, MB, B, LNB, IERR)

(3) 引 数

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$

項番	引数名	型	大きさ	入出力	内　容
1	A	$\left\{\begin{matrix}D\\R\end{matrix}\right\}$	LMA, N	入 力	実対称バンド行列 A(対称バンド型) (付録 B 参照)
2	LMA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	配列 A の次数
4	MB	I	1	入 力	行列 A のバンド幅
5	B	$\left\{\begin{matrix}D\\R\end{matrix}\right\}$	LNB, N	出 力	実対称バンド行列 A(2 次元配列型)(上三角型) (付録 B 参照)
6	LNB	I	1	入 力	配列 B の整合寸法
7	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a) $0 < \mathrm{N} \leq \mathrm{LNB}$

(b) $0 \leq \mathrm{MB} < \mathrm{N}$

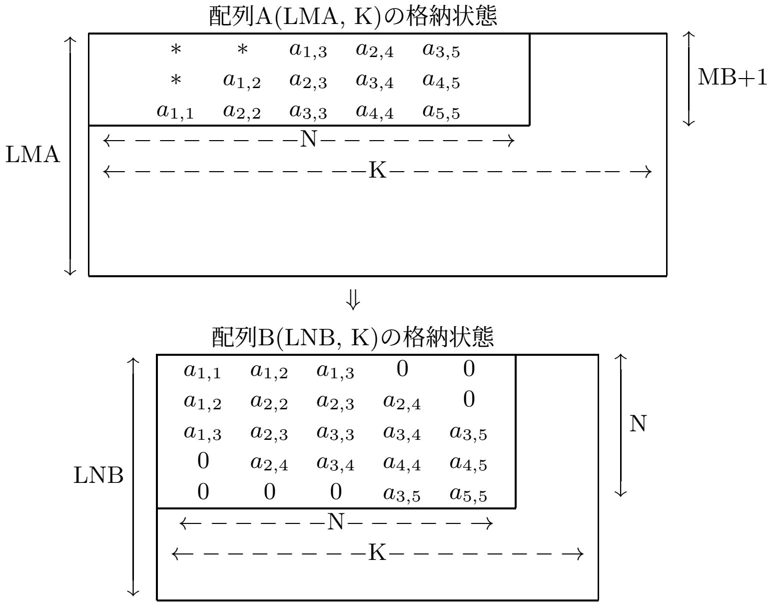
(c) $\mathrm{MB} < \mathrm{LMA}$

(5) エラーインディケータ

IERR の値	意　味	処 理 内 容
0	正常終了.	
1000	$\mathrm{N} = 1$ であった.	処理を続ける.
3000	制限条件 (a), (b) または (c) を満足しなかった.	処理を打ち切る.

(6) 注意事項

(a) 変換後の行列は下三角部分も復元される．また, バンド幅よりも外側の部分には 0.0 が入れられる．
例



備 考
a. ∗ は, 任意の値であることを示す.
b. MB は, バンド幅である.
c. LMA > MB, LNB ≥ N, K ≥ N を満たさなければならない.

(7) 使用例

正値対称バンド行列 A を (対称バンド型) で配列 AC に保持し, A を係数行列とする連立 1 次方程式 $A\boldsymbol{x} = \boldsymbol{b}$ の解を求め, A の LL^{T} 分解を (2 次元配列型)(上三角型) で配列 A に格納する．ただし MB はバンド幅である．

```
IMPLICIT REAL(8) (A-H, O-Z)
PARAMETER (LNA=11, LMB=11)
DIMENSION A(LNA, LNA), AC(LMB, LMB), B(LNA)
  ≀
CALL DBBPSL(AC, LMB, N, MB, B, JERR)
  ≀
CALL DASBEL(AC, LMB, N, MB, A, LNA, KERR)
  ≀
```

2.2.5 DARSJD, RARSJD
スパース行列の格納モードの変換：(実対称行方向１次元リスト型)(上三角型) から (JAD 格納型) へ

(1) **機 能**

実対称スパース行列 A の格納形式を (実対称行方向１次元リスト型)(上三角型) から (JAD 格納型) に変換する.

(2) **使用法**

倍精度サブルーチン:

CALL DARSJD (N, A, IA, JA, LXA, LXIA, MJAD, AJAD, IAJAD, JAJAD, JADORD, IW, IERR)

単精度サブルーチン:

CALL RARSJD (N, A, IA, JA, LXA, LXIA, MJAD, AJAD, IAJAD, JAJAD, JADORD, IW, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$

項番	引数名	型	大きさ	入出力	内　　容
1	N	I	1	入 力	配列 A の次数
2	A	$\left\{\begin{matrix}D\\R\end{matrix}\right\}$	内容参照	入 力	実対称スパース行列 A (実対称行方向１次元リスト型) **大きさ**: IA(N + 1) − 1 (付録 B 参照)
3	IA	I	N + 1	入 力	実対称スパース行列 A(実対称行方向１次元リスト型) のインデックス配列 (付録 B 参照)
4	JA	I	内容参照	入 力	実対称スパース行列 A(実対称行方向１次元リスト型) のインデックス配列 **大きさ**: IA(N + 1) − 1 (付録 B 参照)
5	LXA	I	1	入 力	配列 AJAD および配列 JAJAD に割り当てる大きさ
6	LXIA	I	1	入 力	配列 IAJAD に割り当てる大きさ
7	MJAD	I	1	出 力	行列 A の JAD 格納型における, jagged diagonal の本数
8	AJAD	$\left\{\begin{matrix}D\\R\end{matrix}\right\}$	LXA	出 力	スパース行列 A(JAD 格納型)(付録 B 参照)
9	IAJAD	I	LXIA	出 力	スパース行列 A(JAD 格納型) のインデックス配列 (付録 B 参照)
10	JAJAD	I	LXA	出 力	スパース行列 A(JAD 格納型) のインデックス配列 (付録 B 参照)
11	JADORD	I	N	出 力	スパース行列 A(JAD 格納型) のインデックス配列 (付録 B 参照)
12	IW	I	3×N+1	ワーク	作業配列
13	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a) $N > 0$

(b) $MJAD \leq N$

(c) $MJAD < LXIA$

(d) $IAJAD(MJAD + 1) - 1 \leq LXA$

(5) エラーインディケータ

IERRの値	意　　味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3100	制限条件 (b) を満足しなかった (N, A, IA, JA の入力値に矛盾がある).	
3200	制限条件 (c) を満足しなかった (出力配列 IAJAD の大きさが不足).	
3300	制限条件 (d) を満足しなかった (出力配列 AJAD, JAJAD の大きさが不足).	

(6) 注意事項

(a) 実対称行方向1次元リスト型で入力する行列データは上三角部分のみであるが, 変換後は下三角部分も含めた行列の全てのゼロでない要素をJAD格納型に格納してできる行列データが出力される.

(b) ゼロでない要素の個数が一致する相異なる行が存在する場合, JAD格納型に変換されるとこれらの行は上下方向に連続して配置される. その配置順は本来任意でかまわないが, 本サブルーチンでは入力時により小さい行番号を持っていた行が, 出力時にはより下の方に配置されるように格納される.
図2−1に, 格納形式の変換過程の例を示す.

図 2−1　JAD 型格納状態の例

行列 A

$$\text{N=7}\ \updownarrow\ \begin{array}{|ccccccc|}\hline 1.0 & 0.0 & 0.0 & 1.3 & 0.0 & 0.0 & 0.0 \\ 0.0 & 2.0 & 2.1 & 0.0 & 0.0 & 2.4 & 0.0 \\ 0.0 & 2.1 & 3.0 & 0.0 & 3.2 & 3.3 & 3.4 \\ 1.3 & 0.0 & 0.0 & 4.0 & 4.1 & 0.0 & 4.3 \\ 0.0 & 0.0 & 3.2 & 4.1 & 5.0 & 0.0 & 0.0 \\ 0.0 & 2.4 & 3.3 & 0.0 & 0.0 & 6.0 & 6.1 \\ 0.0 & 0.0 & 3.4 & 4.3 & 0.0 & 6.1 & 7.0 \\ \hline \end{array}$$

← − − − − − − − −N − − − − − − − − →

⇓ 実対称行方向 1 次元リスト型 (入力)

配列 IA の格納状態

1	3	6	10	13	14	16	17

← − − − − − (N + 1) − − − − − →

配列 A の格納状態

1.0	1.3			(→ 次行の先頭 (左端) へ続く)
2.0	2.1	2.4		(→ 次行の先頭 (左端) へ続く)
3.0	3.2	3.3	3.4	(→ 次行の先頭 (左端) へ続く)
4.0	4.1	4.3		(→ 次行の先頭 (左端) へ続く)
5.0				(→ 次行の先頭 (左端) へ続く)
6.0	6.1			(→ 次行の先頭 (左端) へ続く)
7.0				

N

=

1.0 1.3	2.0 2.1 2.4	3.0 3.2 3.3 3.4	4.0 4.1 4.3	5.0	6.0 6.1	7.0	

← − − − − − − − − − − − − − − − − − −(IA(N + 1) − 1) − − − − − − − − − − − − − − − − − − →

配列 JA の格納状態

1	4			(→ 次行の先頭 (左端) へ続く)
2	3	6		(→ 次行の先頭 (左端) へ続く)
3	5	6	7	(→ 次行の先頭 (左端) へ続く)
4	5	7		(→ 次行の先頭 (左端) へ続く)
5				(→ 次行の先頭 (左端) へ続く)
6	7			(→ 次行の先頭 (左端) へ続く)
7				

N

=

1 4	2 3 6	3 5 6 7	4 5 7	5	6 7	7

← − − − − − − − − − − − − (IA(N + 1) − 1) − − − − − − − − − − − − − →

⇓ JAD 型 (出力)

配列 JADORD の格納状態

N

7
6
1
4
5
3
2

配列 IAJAD の格納状態

1	8	15	21	25	26

← −(MJAD + 1)− →

配列 AJAD の格納状態

← − − −MJAD − −− →

(*a*)	(*b*)	(*c*)	(*d*)	(*e*)
2.1	3.0	3.2	3.3	3.4
3.4	4.3	6.1	7.0	
2.4	3.3	6.0	6.1	
1.3	4.0	4.1	4.3	
3.2	4.1	5.0	↓ (*e*)	
2.0	2.1	2.4		
1.0	1.3	↓ (*d*)		
↓ (*b*)	↓ (*c*)			

= | 2.1 3.4 2.4 1.3 3.2 2.0 1.0 | 3.0 4.3 3.3 4.0 4.1 2.1 1.3 | 3.2 6.1 6.0 4.1 5.0 2.4 | 3.3 7.0 6.1 4.3 | 3.4 |

← − − − − − − − − − − − − − − − − − − −(IAJAD(N + 1) − 1) − − − − − − − − − − − − − − − − − →

配列 JAJAD の格納状態

← − − MJAD − − →

(*a*)	(*b*)	(*c*)	(*d*)	(*e*)
2	3	5	6	7
3	4	6	7	
2	3	6	7	
1	4	5	7	
3	4	5	↓ (*e*)	
2	3	6		
1	4	↓ (*d*)		
↓ (*b*)	↓ (*c*)			

= | 2 3 2 1 3 2 1 | 3 4 3 4 4 3 4 | 5 6 6 5 5 6 | 6 7 7 7 | 7 |

← − − − − − − − −(IAJAD(N + 1) − 1) − − − − − − − − →

備 考

a. N は, 行列 A の次数

b. 行列 A の零でない全ての要素を行ごとに左方向に詰め, 零でない要素の数について行を降順にソートしたときにできるデータ配置を考える. このデータ配置における各列を **jagged diagonal** という. MJAD には, jagged diagonal の本数を格納する. このデータ配置において jagged diagonal に沿って左端列から右端列に向かって連続した順番に要素を取り出し, 配列 AJAD に格納する.

c. 配列 JAJAD には, 配列 AJAD に格納した各要素に対応する行番号を格納する.

d. 配列 IAJAD には, 配列 AJAD における各 jagged diagonal の先頭位置を格納する. ただし, MJAD+1 番目には AJAD に格納される要素数に 1 を足した値を格納する.

e. IAJAD (1) には値 1 が設定される.

f. (JAD 格納形式において格納される要素の数) = IAJAD (MJAD+1) −1

(7) **使用例**

実対称スパース行列 A を (実対称行方向 1 次元リスト型) で配列 ACSR に保持し, 内部的に (JAD 格納型) で配列 AJAD に格納してから, A を係数行列とする固有値問題 $A\boldsymbol{x} = \lambda\boldsymbol{b}$ を解く.

```
IMPLICIT REAL(8) (A-H, O-Z)
PARAMETER (N=7, NZ=11, LXA=NZ*2, LXIA=4)
DIMENSION ACSR(NZ), JACSR(NZ), IACSR(N+1)
DIMENSION AJAD(LXA), JACSR(LXA), IAJAD(LXIA), JADORD(N), IW(N*3+1)
  ≀
CALL DARSJD(   N, ACSR, IACSR, JACSR, LXA, LXIA, &
            MJAD, AJAD, IAJAD, JAJAD, JADORD, IW, KERR)
  ≀
NA = IAJAD( MJAD + 1 ) - IAJAD(1)
CALL DCSJSS( MJAD, AJAD, NA, IAJAD, JAJAD, JADORD, &
             N, X, LDA, E, M, TR, IX, IS, ITM, IPREC, &
             NDIA, ITJD, ITQMR, IW, WK, IERR)
  ≀
```

2.2.6 DARGJM, RARGJM スパース行列の格納モードの変換：(実行方向1次元ブロックリスト型) から (MJAD格納型) へ

(1) **機能**

実不規則スパース行列 A の格納形式を (実1次元方向ブロックリスト型) から (MJAD格納型) に変換する.

(2) **使用法**

倍精度サブルーチン:

CALL DARGJM (NB, M, A, IA, JA, ISW, LXA, LXIA, MJAD, AJAD, IAJAD, JAJAD, JADORD, IW, IERR)

単精度サブルーチン:

CALL RARGJM (NB, M, A, IA, JA, ISW, LXA, LXIA, MJAD, AJAD, IAJAD, JAJAD, JADORD, IW, IERR)

(3) **引数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: { 32ビット整数版ではINTEGER(4)
64ビット整数版ではINTEGER(8) }

項番	引数名	型	大きさ	入出力	内容
1	NB	I	1	入力	行列 A をM×Mのブロック行列で区分けした場合の行のブロック数 (または列のブロック数)
2	M	I	1	入力	ブロックの次数
3	A	{D R}	内容参照	入力	不規則スパース行列 A (実行方向1次元ブロックリスト型) **大きさ**: (IA (NB+1) −IA (1))×M×M (付録B参照)
4	IA	I	NB+1	入力	不規則スパース行列 A(実行方向1次元ブロックリスト型) のインデックス配列 (付録B参照)
5	JA	I	内容参照	入力	不規則スパース行列 A(実行方向1次元ブロックリスト型) のインデックス配列 **大きさ**: IA (NB+1) −1 (付録B参照)
6	ISW	I	1	入力	処理スイッチ ISW=0: ブロック内の要素がメモリ上で行方向に連続 (Row Major) ISW=1: ブロック内の要素がメモリ上で列方向に連続 (Column Major)
7	LXA	I	1	入力	配列AJADおよび配列JAJADに割り当てる大きさ
8	LXIA	I	1	入力	配列IAJADに割り当てる大きさ
9	MJAD	I	1	出力	行列 A のMJAD格納型における, jagged diagonalの本数
10	AJAD	{D R}	LXA×M×M	出力	スパース行列 A(MJAD格納型) (付録B参照)
11	IAJAD	I	LXIA	出力	スパース行列 A(MJAD格納型) のインデックス配列 (付録B参照)

項番	引数名	型	大きさ	入出力	内　　容
12	JAJAD	I	LXA	出 力	スパース行列 A(MJAD 格納型) のインデックス配列 (付録 B 参照)
13	JADORD	I	NB	出 力	スパース行列 A(MJAD 格納型) のインデックス配列 (付録 B 参照)
14	IW	I	$2\times\mathrm{NB}+1$	ワーク	作業配列
15	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a) $\mathrm{NB} > 0$

(b) $\mathrm{M} > 0$

(c) $\mathrm{ISW} \in \{0, 1\}$

(d) $\mathrm{MJAD} \le \mathrm{NB}$

(e) $\mathrm{MJAD} < \mathrm{LXIA}$

(f) $\mathrm{IAJAD}(\mathrm{MJAD}+1) - \mathrm{IAJAD}(1) \le \mathrm{LXA}$

(5) エラーインディケータ

IERR の値	意　　味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3100	制限条件 (b) を満足しなかった.	
3200	制限条件 (c) を満足しなかった.	
3300	制限条件 (d) を満足しなかった.	
3400	制限条件 (e) を満足しなかった (出力配列 IAJAD の大きさが不足).	
3500	制限条件 (f) を満足しなかった (出力配列 AJAD, JAJAD の大きさが不足).	

(6) 注意事項

(a) 以下の例は, 実行方向 1 次元ブロックリスト型から MJAD 格納型に変換する流れを示したものである.

例

図 2−2　MJAD 型の格納状態の例 $(\mathrm{M}=2, \mathrm{ISW}=0$ の場合$)$

行列 A

$$\text{NB=4}\updownarrow\begin{bmatrix} B & 0 & C & 0 \\ D & F & 0 & G \\ 0 & 0 & H & 0 \\ 0 & K & 0 & L \end{bmatrix}$$

$$\leftarrow - - \text{NB} - - \rightarrow$$

$$B=\begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \end{bmatrix},\quad C=\begin{bmatrix} c_1 & c_2 \\ c_3 & c_4 \end{bmatrix},\quad D=\overset{\leftarrow -\mathrm{M}- \rightarrow}{\begin{bmatrix} d_1 & d_2 \\ d_3 & d_4 \end{bmatrix}},\ \updownarrow\mathrm{M}$$

$$F=\begin{bmatrix} f_1 & f_2 \\ f_3 & f_4 \end{bmatrix},\quad G=\begin{bmatrix} g_1 & g_2 \\ g_3 & g_4 \end{bmatrix},\quad H=\begin{bmatrix} h_1 & h_2 \\ h_3 & h_4 \end{bmatrix},$$

$$K=\begin{bmatrix} k_1 & k_2 \\ k_3 & k_4 \end{bmatrix},\quad L=\begin{bmatrix} l_1 & l_2 \\ l_3 & l_4 \end{bmatrix},\quad 0=\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$\Downarrow$　　実行方向 1 次元ブロックリスト型（入 力）

配列 IA の格納状態

$$\begin{array}{|c|c|c|c|c|}\hline 1 & 3 & 6 & 7 & 9 \\ \hline \end{array}$$

$$\leftarrow -(\mathrm{NB}+1)- \rightarrow$$

配列 A の格納状態 (ISW=0の場合)

$$\text{NB}\updownarrow\begin{array}{ll} \boxed{B\quad C} & (\rightarrow \text{次行の先頭 (左端) へ続く}) \\ \boxed{D\quad F\quad G} & (\rightarrow \text{次行の先頭 (左端) へ続く}) \\ \boxed{H} & (\rightarrow \text{次行の先頭 (左端) へ続く}) \\ \boxed{K\quad L} & \end{array}$$

$$=\begin{array}{|c|c|c|c|}\hline B\quad C & D\quad F\quad G & H & K\quad L \\ \hline \end{array}$$

$$\leftarrow - - - - (\mathrm{IA}(\mathrm{NB}+1)-1) - - - - \rightarrow$$

$$=\begin{array}{|cccccccc|c|cccccccc|}\hline b_1 & b_2 & b_3 & b_4 & c_1 & c_2 & c_3 & c_4 & \cdots & k_1 & k_2 & k_3 & k_4 & l_1 & l_2 & l_3 & l_4 \\ \hline \end{array}$$

$$\leftarrow - - - - - - - - - - - ((\mathrm{IA}(\mathrm{NB}+1)-\mathrm{IA}(1))\times\mathrm{M}\times\mathrm{M}) - - - - - - - - - - - \rightarrow$$

配列 JA の格納状態

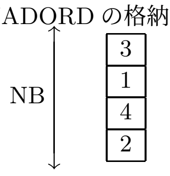
$$\text{NB}\updownarrow\begin{array}{ll} \boxed{1\quad 3} & (\rightarrow \text{次行の先頭 (左端) へ続く}) \\ \boxed{1\quad 2\quad 4} & (\rightarrow \text{次行の先頭 (左端) へ続く}) \\ \boxed{3} & (\rightarrow \text{次行の先頭 (左端) へ続く}) \\ \boxed{2\quad 4} & \end{array}$$

$$=\begin{array}{|c|c|c|c|}\hline 1\quad 3 & 1\quad 2\quad 4 & 3 & 2\quad 4 \\ \hline \end{array}$$

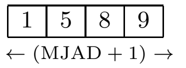
$$\leftarrow - - -(\mathrm{IA}(\mathrm{NB}+1)-1) - - - \rightarrow$$

⇓ MJAD 格納型（出 力）

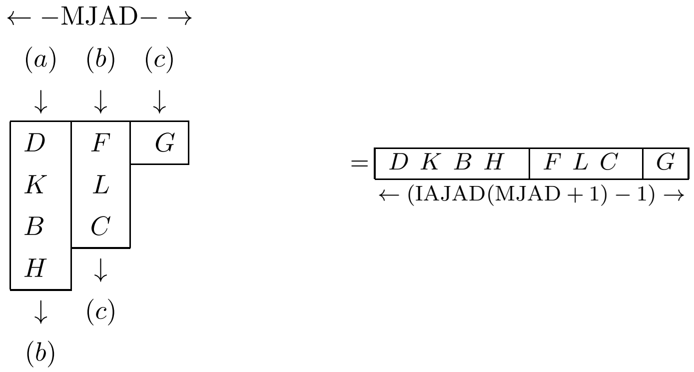
配列 JADORD の格納状態



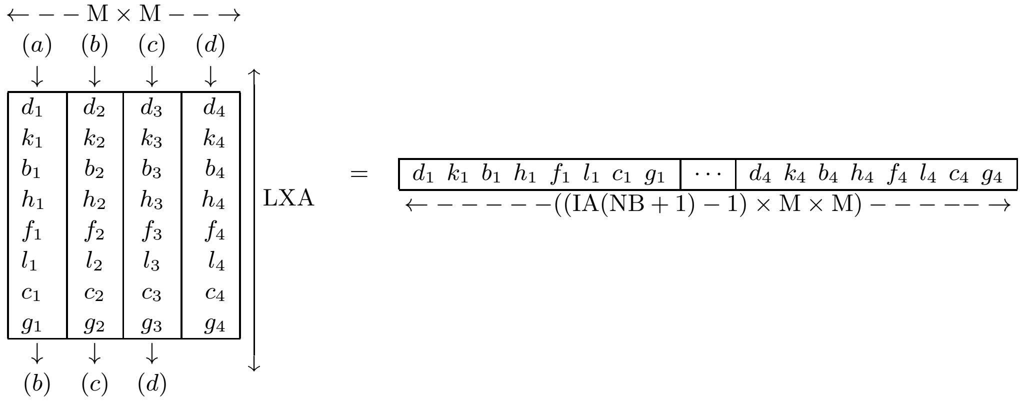
配列 IAJAD の格納状態



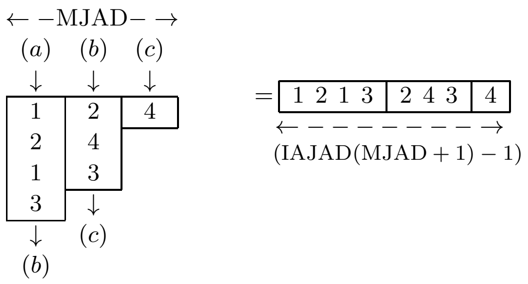
(⋆)Jagged diagonal のブロックの順番



(⋆⋆) 配列 AJAD の格納状態



配列 JAJAD の格納状態



備 考

a. NB は, 行列 A を M×M のブロック行列で区分けした場合の行のブロック数 (または列のブロック数)

b. 行列 A の零行列でない全てのブロック行列を行ごとに左方向に詰め, 零行列でないブロック行列の数について行を降順にソートしたときにできるデータ配置を考える $(\star)$. このデータ配置における各ブロック列を jagged diagonal という. MJAD には, jagged diagonal の本数を格納する. 配列 AJAD に格納する方法は次の通り. まず, 各ブロック行列 (D, K, B, H, F, C, G) から 1 行 1 列の要素を取り出す. ここで, 各ブロック行列から 1 行 1 列の要素を取り出す順番は, 上述の jagged diagonal に沿ってブロック行列が出てくる順番とする $(d_1, k_1, b_1, h_1, f_1, c_1, g_1)$. これを各ブロック行列の M 行 M 列の要素まで繰り返し ((a),(b),(c),(d)), 取り出した要素を配列 AJAD に格納する.

c. 配列 JAJAD には, 配列 AJAD に格納した各ブロック行列に対応するブロック列番号を格納する.

d. 配列 IAJAD には, 配列 AJAD における各 jagged diagonal の先頭位置を格納する. ただし, MJAD+1 番目には AJAD に格納されるブロック行列数に 1 を足した値を格納する.

e. IAJAD (1) には値 1 が設定される.

f. (MJAD 格納形式において格納される要素の数) = (IAJAD (MJAD+1) −1)×M×M

g. 零行列でないブロック行列の個数が一致する相異なるブロック行が存在する場合, MJAD 格納型に変換されるとこれらのブロック行は上下方向に連続して配置される. その配置順は本来任意でかまわないが, 本サブルーチンでは入力時により小さいブロック行番号を持っていたブロック行が, 出力時にはより下の方に配置されるように格納される.

h. 同じブロックにある各要素のメモリ上の位置は, LXA 個の飛びを持つ $(\star\star)$. 例えば, 同じブロック D にある要素 (d_1,d_2,d_3,d_4) のメモリ上の位置は, LXA 個の飛びを持つ.

i. 行列 A のブロック内の各要素の格納の順番が列方向に連続になる場合は, ISW = 1 とすればよい.

(b) 本サブルーチンを使った格納形式の変換は, なるべく回数を削減した方がよい. 例えば, スパース行列の連立 1 次方程式や固有値方程式の反復解法等で, 行列 A を変更せずに行列ベクトル積を繰り返し計算する場合は, 反復ループの外で本サブルーチンを用いて一度だけ格納形式の変換を行い, 反復ループ内で行列ベクトル積を繰り返し使えば効率良く計算できる.

(c) 本サブルーチンで得られる MJAD 形式のスパース行列ベクトル積を求める場合, ブロックの次数が $M = 1, 3, 4$ であれば, 3.2.24 $\left\{\begin{matrix}\text{DAMVJ1}\\\text{RAMVJ1}\end{matrix}\right\}$, 3.2.25 $\left\{\begin{matrix}\text{DAMVJ3}\\\text{RAMVJ3}\end{matrix}\right\}$, 3.2.26 $\left\{\begin{matrix}\text{DAMVJ4}\\\text{RAMVJ4}\end{matrix}\right\}$ を用いて計算できる. これらは, Vector Engine 向けに高度にアセンブリチューニングされているため, 効率良く計算できる.

(7) **使用例**

3 × 3 のブロック行列を要素に持つ不規則スパース行列 A を (実行方向 1 次元ブロックリスト型) で配列 A に保持し, 内部的に (MJAD 格納型) で配列 AJAD に格納してから, 行列ベクトル積 $\boldsymbol{y} = \beta\boldsymbol{y} + \alpha A\boldsymbol{x}$ を求める.

```
! *** EXAMPLE OF DARGJM AND DAMVJ1 ***
      INTEGER NB,NZ,LXA,LXIA,M
      PARAMETER( NB=4, M=3, NZ=8, ISW=0, LXA=NZ, LXIA=N+1 )
      INTEGER IA(NB+1),JA(LXA),MJAD,IAJAD(LXIA),JAJAD(LXA),JADORD(NB)
      INTEGER IW(NB*2+1),IERR
      INTEGER I,J,K,L
      REAL(8) A(NZ*M*M),AJAD(LXA*M*M),X(NB*M),Y(NB*M),W(NB*M)
      REAL(8) ALPHA,BETA
      PARAMETER (ALPHA=1.0D0, BETA=1.0D0)
       ≀
      CALL DARGJM &
      (NB,M,A,IA,JA,ISW,LXA,LXIA,MJAD,AJAD,IAJAD,JAJAD,JADORD,IW,IERR)
       ≀
      CALL DAMVJ3 &
      (AJAD,LXA,IAJAD,JAJAD,JADORD,NB,MJAD,ALPHA,BETA,X,Y,W,IERR)
       ≀
```

2.2.7 ZARSJD, CARSJD
スパース行列の格納モードの変換：(エルミート行方向1次元リスト型)(上三角型) から (JAD 格納型) へ

(1) **機 能**

エルミートスパース行列 A の格納形式を (エルミート行方向1次元リスト型)(上三角型) から (JAD 格納型) に変換する.

(2) **使用法**

倍精度サブルーチン:

CALL ZARSJD (N, A, IA, JA, LXA, LXIA, MJAD, AJAD, IAJAD, JAJAD, JADORD, IW, IERR)

単精度サブルーチン:

CALL CARSJD (N, A, IA, JA, LXA, LXIA, MJAD, AJAD, IAJAD, JAJAD, JADORD, IW, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\begin{Bmatrix} \text{32 ビット整数版では INTEGER(4)} \\ \text{64 ビット整数版では INTEGER(8)} \end{Bmatrix}$

項番	引数名	型	大きさ	入出力	内　　容
1	N	I	1	入 力	配列 A の次数
2	A	$\begin{Bmatrix} Z \\ C \end{Bmatrix}$	内容参照	入 力	エルミートスパース行列 A (エルミート行方向1次元リスト型) **大きさ**: IA (N+1) −1 (付録 B 参照)
3	IA	I	N+1	入 力	エルミートスパース行列 A(エルミート行方向1次元リスト型) のインデックス配列 (付録 B 参照)
4	JA	I	内容参照	入 力	エルミートスパース行列 A(エルミート行方向1次元リスト型) のインデックス配列 **大きさ**: IA (N+1) −1 (付録 B 参照)
5	LXA	I	1	入 力	配列 AJAD および配列 JAJAD に割り当てる大きさ
6	LXIA	I	1	入 力	配列 IAJAD に割り当てる大きさ
7	MJAD	I	1	出 力	行列 A の JAD 格納型における, jagged diagonal の本数
8	AJAD	$\begin{Bmatrix} Z \\ C \end{Bmatrix}$	LXA	出 力	スパース行列 A(JAD 格納型)(付録 B 参照)
9	IAJAD	I	LXIA	出 力	スパース行列 A(JAD 格納型) のインデックス配列 (付録 B 参照)
10	JAJAD	I	LXA	出 力	スパース行列 A(JAD 格納型) のインデックス配列 (付録 B 参照)
11	JADORD	I	N	出 力	スパース行列 A(JAD 格納型) のインデックス配列 (付録 B 参照)
12	IW	I	3×N+1	ワーク	作業配列
13	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $\mathrm{NB} > 0$

(b) $\mathrm{M} > 0$

(c) $\mathrm{ISW} \in \{0, 1\}$

(d) $\mathrm{MJAD} \leq \mathrm{NB}$

(e) $\mathrm{MJAD} < \mathrm{LXIA}$

(f) $\mathrm{IAJAD}(\mathrm{MJAD} + 1) - 1 \leq \mathrm{LXA}$

(5) **エラーインディケータ**

IERR の値	意　　味	処　理　内　容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3100	制限条件 (b) を満足しなかった.	
3200	制限条件 (b) を満足しなかった (N, A, IA, JA の入力値に矛盾がある).	
3200	制限条件 (c) を満足しなかった (出力配列 IAJAD の大きさが不足).	
3300	制限条件 (d) を満足しなかった (出力配列 AJAD, JAJAD の大きさが不足).	

(6) **注意事項**

(a) エルミート行方向 1 次元リスト型で入力する行列データは上三角部分のみであるが, 変換後は下三角部分も含めた行列の全てのゼロでない要素を JAD 格納型に格納してできる行列データが出力される.

(b) ゼロでない要素の個数が一致する相異なる行が存在する場合, JAD 格納型に変換されるとこれらの行は上下方向に連続して配置される. その配置順は本来任意でかまわないが, 本サブルーチンでは入力時により小さい行番号を持っていた行が, 出力時にはより下の方に配置されるように格納される.

例

行列 A

$$\text{N=7}\ \updownarrow\ \begin{array}{|ccccccc|} \hline a_{1,1} & 0. & 0. & a_{1,4} & 0. & 0. & 0. \\ 0. & a_{2,2} & a_{2,3} & 0. & 0. & a_{2,6} & 0. \\ 0. & \overline{a}_{2,3} & a_{3,3} & 0. & a_{3,5} & a_{3,6} & a_{3,7} \\ \overline{a}_{1,4} & 0. & 0. & a_{4,4} & a_{4,5} & 0. & a_{4,7} \\ 0. & 0. & \overline{a}_{3,5} & \overline{a}_{4,5} & a_{5,5} & 0. & 0. \\ 0. & \overline{a}_{2,6} & \overline{a}_{3,6} & 0. & 0. & a_{6,6} & a_{6,7} \\ 0. & 0. & \overline{a}_{3,7} & \overline{a}_{4,7} & 0. & \overline{a}_{6,7} & a_{7,7} \\ \hline \end{array}$$

$$\longleftarrow - - - - - - - - \text{N} - - - - - - - - \longrightarrow$$

$$\Downarrow$$

配列 IA の格納状態

1	3	6	10	13	14	16	17

$$\longleftarrow - - - - - (\text{N}+1) - - - - - \longrightarrow$$

配列 A の格納状態

N

$a_{1,1}\quad a_{1,4}$ (→ 次行の先頭 (左端) へ続く)
$a_{2,2}\quad a_{2,3}\quad a_{2,6}$ (→ 次行の先頭 (左端) へ続く)
$a_{3,3}\quad a_{3,5}\quad a_{3,6}\quad a_{3,7}$ (→ 次行の先頭 (左端) へ続く)
$a_{4,4}\quad a_{4,5}\quad a_{4,7}$ (→ 次行の先頭 (左端) へ続く)
$a_{5,5}$ (→ 次行の先頭 (左端) へ続く)
$a_{6,6}\quad a_{6,7}$ (→ 次行の先頭 (左端) へ続く)
$a_{7,7}$

=

$a_{1,1}$ $a_{1,4}$	$a_{2,2}$ $a_{2,3}$ $a_{2,6}$	$a_{3,3}$ $a_{3,5}$ $a_{3,6}$ $a_{3,7}$	$a_{4,4}$ $a_{4,5}$ $a_{4,7}$	$a_{5,5}$	$a_{6,6}$ $a_{6,7}$	$a_{7,7}$

$$\longleftarrow - (\text{IA}(\text{N}+1)-1) - \longrightarrow$$

配列 JA の格納状態

N

1 4 (→ 次行の先頭 (左端) へ続く)
2 3 6 (→ 次行の先頭 (左端) へ続く)
3 5 6 7 (→ 次行の先頭 (左端) へ続く)
4 5 7 (→ 次行の先頭 (左端) へ続く)
5 (→ 次行の先頭 (左端) へ続く)
6 7 (→ 次行の先頭 (左端) へ続く)
7

=

1 4	2 3 6	3 5 6 7	4 5 7	5	6 7	7

$$\longleftarrow - - - - - - - - - - - - - (\text{IA}(\text{N}+1)-1) - - - - - - - - - - - - - - \longrightarrow$$

⇓

配列 JADORD の格納状態

↑ N ↓
7
6
1
4
5
3
2

配列 IAJAD の格納状態

1	8	15	21	25	26

← −(MJAD + 1)− →

配列 AJAD の格納状態

← − − − − − MJAD − − − − − →

(*a*) ↓	(*b*) ↓	(*c*) ↓	(*d*) ↓	(*e*) ↓
$\overline{a}_{2,3}$	$a_{3,3}$	$a_{3,5}$	$a_{3,6}$	$a_{3,7}$
$\overline{a}_{3,7}$	$\overline{a}_{4,7}$	$\overline{a}_{6,7}$	$a_{7,7}$	
$\overline{a}_{2,6}$	$\overline{a}_{3,6}$	$a_{6,6}$	$a_{6,7}$	
$\overline{a}_{1,4}$	$a_{4,4}$	$a_{4,5}$	$a_{4,7}$	
$\overline{a}_{3,5}$	$\overline{a}_{4,5}$	$a_{5,5}$	↓ (*e*)	
$a_{2,2}$	$a_{2,3}$	$a_{2,6}$		
$a_{1,1}$	$a_{1,4}$	↓ (*d*)		
↓ (*b*)	↓ (*c*)			

=

$\overline{a}_{2,3}$ $\overline{a}_{3,7}$ $\overline{a}_{2,6}$ $\overline{a}_{1,4}$ $\overline{a}_{3,5}$ $a_{2,2}$ $a_{1,1}$	$a_{3,3}$ $\overline{a}_{4,7}$ $\overline{a}_{3,6}$ $a_{4,4}$ $\overline{a}_{4,5}$ $a_{2,3}$ $a_{1,4}$	$a_{3,5}$ $\overline{a}_{6,7}$ $a_{6,6}$ $a_{4,5}$ $a_{5,5}$ $a_{2,6}$	$a_{3,6}$ $a_{7,7}$ $a_{6,7}$ $a_{4,7}$	$a_{3,7}$

← − (IAJAD(N + 1) − 1) − →

配列 JAJAD の格納状態

← − − MJAD − − →

(*a*) ↓	(*b*) ↓	(*c*) ↓	(*d*) ↓	(*e*) ↓
2	3	5	6	7
3	4	6	7	
2	3	6	7	
1	4	5	7	
3	4	5	↓ (*e*)	
2	3	6		
1	4	↓ (*d*)		
↓ (*b*)	↓ (*c*)			

=

2 3 2 1 3 2 1	3 4 3 4 4 3 4	5 6 6 5 5 6	6 7 7 7	7

← − − − − − − − −(IAJAD(N + 1) − 1) − − − − − − − − →

備　考

a. x の複素共役を $\overline{x}$ で表している.

b. N は, 行列 A の次数

c. 行列 A の零でない全ての要素を行ごとに左方向に詰め, 零でない要素の数について行を降順にソートしたときにできるデータ配置を考える. このデータ配置における各列を **jagged diagonal** という. MJAD には, jagged diagonal の本数を格納する. このデータ配置において jagged diagonal に沿って左端列から右端列に向かって連続した順番に要素を取り出し, 配列 AJAD に格納する.

d. 配列 JAJAD には, 配列 AJAD に格納した各要素に対応する行番号を格納する.

e. 配列 IAJAD には, 配列 AJAD における各 jagged diagonal の先頭位置を格納する. ただし, MJAD+1 番目には AJAD に格納される要素数に 1 を足した値を格納する.

f. IAJAD (1) には値 1 が設定される.

g. (JAD 格納形式において格納される要素の数) = IAJAD (MJAD+1) −1

(7) **使用例**

エルミートスパース行列 A を (エルミート行方向 1 次元リスト型) で配列 ACSR に保持し, 内部的に (JAD 格納型) で配列 AJAD に格納してから, A を係数行列とする固有値問題 $A\boldsymbol{x} = \lambda\boldsymbol{b}$ を解く.

```
      IMPLICIT COMPLEX*16 (A-H, O-Z)
      PARAMETER (N=7, NZ=11, LXA=NZ*2, LXIA=4)
      DIMENSION ACSR(NZ), JACSR(NZ), IACSR(N+1)
      DIMENSION AJAD(LXA), JACSR(LXA), IAJAD(LXIA), JADORD(N), IW(N*3+1)
        ≀
      CALL ZARSJD(   N, ACSR, IACSR, JACSR, LXA, LXIA, &
                  MJAD, AJAD, IAJAD, JAJAD, JADORD, IW, KERR)
        ≀
      NA = IAJAD( MJAD + 1 ) - IAJAD(1)
      CALL ZCHJSS( MJAD, AJAD, NA, IAJAD, JAJAD, JADORD, &
                   N, X, LDA, E, M, TR, IX, IS, ITM, IPREC, &
                   NDIA, ITJD, ITQMR, IW, WK, IERR)
        ≀
```

2.2.8 ZARGJM, CARGJM スパース行列の格納モードの変換 : (複素行方向 1 次元ブロックリスト型) から (MJAD 格納型) へ

(1) 機 能

複素不規則スパース行列 A の格納形式を (複素行方向 1 次元ブロックリスト型) から (MJAD 格納型) に変換する.

(2) 使用法

倍精度サブルーチン:

CALL ZARGJM (NB, M, A, IA, JA, ISW, LXA, LXIA, MJAD, AJAD, IAJAD, JAJAD, JADORD, IW, IERR)

単精度サブルーチン:

CALL CARGJM (NB, M, A, IA, JA, ISW, LXA, LXIA, MJAD, AJAD, IAJAD, JAJAD, JADORD, IW, IERR)

(3) 引 数

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: { 32 ビット整数版では INTEGER(4)
　　 64 ビット整数版では INTEGER(8) }

項番	引数名	型	大きさ	入出力	内　　容
1	NB	I	1	入 力	行列 A を M×M のブロック行列で区分けした場合の行のブロック数 (または列のブロック数)
2	M	I	1	入 力	ブロックの次数
3	A	Z C	内容参照	入 力	不規則スパース行列 A (複素行方向 1 次元ブロックリスト型) **大きさ**: (IA (NB+1) −IA (1))×M×M(付録 B 参照)
4	IA	I	NB+1	入 力	不規則スパース行列 A(複素行方向 1 次元ブロックリスト型) のインデックス配列 (付録 B 参照)
5	JA	I	内容参照	入 力	不規則スパース行列 A(複素行方向 1 次元ブロックリスト型) のインデックス配列 **大きさ**: IA (NB+1) −1 (付録 B 参照)
6	ISW	I	1	入 力	処理スイッチ ISW=0: ブロック内の要素がメモリ上で行方向に連続 (Row Major) ISW=1: ブロック内の要素がメモリ上で列方向に連続 (Column Major)
7	LXA	I	1	入 力	配列 AJAD および配列 JAJAD に割り当てる大きさ
8	LXIA	I	1	入 力	配列 IAJAD に割り当てる大きさ
9	MJAD	I	1	出 力	行列 A の MJAD 格納型における, jagged diagonal の本数

項番	引数名	型	大きさ	入出力	内　　容
10	AJAD	$\left\{\begin{matrix}\mathrm{Z}\\\mathrm{C}\end{matrix}\right\}$	LXA×M×M	出 力	スパース行列 A(MJAD 格納型) (付録 B 参照)
11	IAJAD	I	LXIA	出 力	スパース行列 A(MJAD 格納型) のインデックス配列 (付録 B 参照)
12	JAJAD	I	LXA	出 力	スパース行列 A(MJAD 格納型) のインデックス配列 (付録 B 参照)
13	JADORD	I	NB	出 力	スパース行列 A(MJAD 格納型) のインデックス配列 (付録 B 参照)
14	IW	I	2×NB+1	ワーク	作業配列
15	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a) $\mathrm{NB} > 0$

(b) $\mathrm{M} > 0$

(c) $\mathrm{ISW} \in \{0, 1\}$

(d) $\mathrm{MJAD} \leq \mathrm{NB}$

(e) $\mathrm{MJAD} < \mathrm{LXIA}$

(f) $\mathrm{IAJAD(MJAD + 1) - IAJAD(1)} \leq \mathrm{LXA}$

(5) エラーインディケータ

IERR の値	意　　味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3100	制限条件 (b) を満足しなかった.	
3200	制限条件 (c) を満足しなかった.	
3300	制限条件 (d) を満足しなかった.	
3400	制限条件 (e) を満足しなかった (出力配列 IAJAD の大きさが不足).	
3500	制限条件 (f) を満足しなかった (出力配列 AJAD, JAJAD の大きさが不足).	

(6) 注意事項

(a) 零行列でないブロック行列の個数が一致する相異なるブロック行が存在する場合, MJAD 格納型に変換されるとこれらのブロック行は上下方向に連続して配置される. その配置順は本来任意でかまわないが, 本サブルーチンでは入力時により小さいブロック行番号を持っていたブロック行が, 出力時にはより下の方に配置されるように格納される. (2.2.6 図 2−2 参照)

(b) 本サブルーチンを使った格納形式の変換は, なるべく回数を削減した方がよい. 例えば, スパース行列の連立 1 次方程式や固有値方程式の反復解法等で, 行列 A を変更せずに行列ベクトル積を繰り返し計算する場合は, 反復ループの外で本サブルーチンを用いて一度だけ格納形式の変換を行い, 反復ループ内で行列ベクトル積を繰り返し使えば効率良く計算できる.

(c) 本サブルーチンで得られる MJAD 形式のスパース行列ベクトル積を求める場合, ブロックの次数が $M = 1$ であれば, 3.2.27 $\left\{\begin{matrix}\text{ZANVJ1}\\\text{CANVJ1}\end{matrix}\right\}$を用いて計算できる．これらは, Vector Engine 向けに高度にアセンブリチューニングされているため, 効率良く計算できる.

(7) **使用例**

不規則スパース行列 A を (複素行方向 1 次元ブロックリスト型) で配列 A に保持し, 内部的に (MJAD 格納型) で配列 AJAD に格納してから, 行列ベクトル積 $\boldsymbol{y} = \beta\boldsymbol{y} + \alpha A\boldsymbol{x}$ を求める.

```
! *** EXAMPLE OF ZARGJM AND ZANVJ1 ***
      INTEGER NB,NZ,LXA,LXIA,M
      PARAMETER( NB=4, M=1, NZ=8, ISW=0, LXA=NZ, LXIA=N+1 )
      INTEGER IA(NB+1),JA(LXA),MJAD,IAJAD(LXIA),JAJAD(LXA),JADORD(NB)
      INTEGER IW(NB*2+1),IERR
      INTEGER I,J,K,L
      COMPLEX(8) A(NZ*M*M),AJAD(LXA*M*M),X(NB*M),Y(NB*M),W(NB*M)
      COMPLEX(8) ALPHA,BETA
      PARAMETER (ALPHA=(1.0D0,0.D0), BETA=(1.0D0,0.D0) )
       ≀
      CALL ZARGJM &
      (NB,M,A,IA,JA,ISW,LXA,LXIA,MJAD,AJAD,IAJAD,JAJAD,JADORD,IW,IERR)
       ≀
      CALL ZANVJ1 &
      (AJAD,LXA,IAJAD,JAJAD,JADORD,NB,MJAD,ALPHA,BETA,X,Y,W,IERR)
       ≀
```

2.2.9 DXA005, RXA005
スパース行列の格納モードの変換 : (列方向 1 次元リスト型) から (ELLPACK 型) へ

(1) 機 能

スパース行列 A の格納形式を (列方向 1 次元リスト型) から (ELLPACK 型) へ変換する.

(2) 使用法

倍精度サブルーチン:

CALL DXA005 (ITYPE, N, VALUES, IPONTR, IRWIND, NNZ, A, LNA, M, JA, IWK, IERR)

単精度サブルーチン:

CALL RXA005 (ITYPE, N, VALUES, IPONTR, IRWIND, NNZ, A, LNA, M, JA, IWK, IERR)

(3) 引 数

D:倍精度実数型　Z:倍精度複素数型　I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$
R:単精度実数型　C:単精度複素数型

項番	引数名	型	大きさ	入出力	内　容
1	ITYPE	I	1	入 力	列方向 1 次元リスト型における行列データ参照方法の指定スイッチ ITYPE=　1 : 上三角部分と下三角部分の両方 (非対称行列用) 2 : 上三角部分または下三角部分のみ (対称行列用)
2	N	I	1	入 力	行列 A の次数
3	VALUES	$\left\{\begin{matrix}\text{D}\\\text{R}\end{matrix}\right\}$	NNZ	入 力	行列 A の非零要素 (列方向 1 次元リスト型) (付録 B 参照)
4	IPONTR	I	N+1	入 力	IRWIND における各列の開始位置 ただし, 入力する要素がない列 j では IPONTR(j) $=$ IPONTR($j+1$)　($j = 1, \cdots$, N)
5	IRWIND	I	NNZ	入 力	行列 A の非零要素の行番号 VALUES(i) に格納された行列 A の要素の行番号を IRWIND(i) に格納する　($i = 1, \cdots$, NNZ)
6	NNZ	I	1	入 力	配列 VALUES に格納される行列 A の要素数
7	A	$\left\{\begin{matrix}\text{D}\\\text{R}\end{matrix}\right\}$	LNA, M	出 力	行列 A の非零要素値の配列
8	LNA	I	1	入 力	配列 A および JA の整合寸法
9	M	I	1	入 力	配列 A および JA の列数
				出 力	必要な M の大きさ (IERR= 2000 が出力された時)
10	JA	I	LNA, M	出 力	行列 A の非零構造データを格納する配列
11	IWK	I	N	ワーク	作業領域
12	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a) $\mathrm{ITYPE} \in \{1, 2\}$

(b) $1 \leq \mathrm{N} \leq \mathrm{LNA}$

(c) $\mathrm{IPONTR}(1) = 1$
$1 \leq \mathrm{IPONTR}(j) \leq \mathrm{NNZ} + 1 \quad (j = 2, \ldots, \mathrm{N})$
$\mathrm{IPONTR}(\mathrm{N} + 1) = \mathrm{NNZ} + 1$

(d) $0 \leq \mathrm{IPONTR}(j + 1) - \mathrm{IPONTR}(j) \leq \mathrm{N} \quad (i = 1, \ldots, \mathrm{N})$

(e) $1 \leq \mathrm{IRWIND}(k) \leq \mathrm{N} \quad (k = 1, \ldots, \mathrm{NNZ})$

(f) 同じ列 j に格納されている要素に対する $\mathrm{IRWIND}(k) \quad (k = \mathrm{IPONTR}(j), \ldots, \mathrm{IPONTR}(j + 1) - 1)$ はすべて異なる.

(g) $\mathrm{NNZ} \geq 1$

(h) $\mathrm{M} \geq 1$

(5) エラーインディケータ

IERR の値	意　　味	処　理　内　容
0	正常終了.	
2000	M の値が小さい.	M に必要な大きさを出力して処理を打ち切る.
2200	対角要素の一部または全部が入力されなかった.	入力されなかった対角要素は 0.0 であるとして, 処理を続ける.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	
3030	制限条件 (d) を満足しなかった.	
3050	制限条件 (e) を満足しなかった.	
3060	制限条件 (f) を満足しなかった.	
3070	制限条件 (g) を満足しなかった.	
3080	制限条件 (h) を満足しなかった.	

(6) 注意事項

(a) 配列 VALUES, IPONTR, IRWIND, A および JA の格納方法は付録 B 参照のこと.

(b) M に設定した値が不十分だった場合, 処理が打ち切られ, 格納モードの変換は行われない (IERR = 2000). その場合は, 領域を一旦解放し, M に出力された値を用いて再度確保すれば格納モードの変換が行える.

```
 ≀
CALL DXA005(ITYPE, N, VALUES, IPONTR, IRWIND, A, LNA, M, JA, IWK, IERR)
IF( IERR .EQ. 2000 ) THEN
   DEALLOCATE(A, JA)
   ALLOCATE (A(LNA, M), JA(LNA, M))
   CALL DXA005(ITYPE, N, VALUES, IPONTR, IRWIND, A, LNA, M, JA, IWK, IERR)
ENDIF
 ≀
```

(7) **使用例**

(a) 問 題

行列 A の格納形式を (列方向 1 次元リスト型) から (ELLPACK 型) に変換する.

$$A = \begin{bmatrix} 1.0 & 1.1 & 0.0 & 1.2 & 1.4 \\ -2.1 & 2.0 & 2.1 & 0.0 & 0.0 \\ 0.0 & 0.0 & 3.0 & 0.0 & 3.5 \\ 4.5 & 0.0 & 0.0 & 4.0 & 4.1 \\ 5.0 & 0.0 & -5.4 & -5.1 & 5.0 \end{bmatrix}$$

(b) 入力データ

行列の要素値, 要素の位置, 要素の列番号, $\mathrm{ITYPE} = 1, \mathrm{N} = 5, \mathrm{LNA} = 5, \mathrm{M} = 1$

(c) 主プログラム

```
      PROGRAM BXA005
! *** EXAMPLE OF DXA005 ***
      IMPLICIT NONE
!
      INTEGER N,NNZ,LNA
      PARAMETER( N = 5, NNZ = 16, LNA = 5 )
      INTEGER ITYPE,IPONTR(N+1),IRWIND(NNZ),M,IWK(N),IERR
      REAL(8) VALUES(NNZ)
      INTEGER,ALLOCATABLE :: JA(:,:)
      REAL(8),ALLOCATABLE :: A(:,:)
!
      INTEGER I,J,KERR
!
      KERR  = 0
      ITYPE = 1
!
      DO 100 I= 1,NNZ
         READ(5,*) VALUES(I)
  100 CONTINUE
      DO 110 I= 1,N+1
         READ(5,*) IPONTR(I)
  110 CONTINUE
      DO 120 I= 1,NNZ
         READ(5,*) IRWIND(I)
  120 CONTINUE
!
      WRITE(6,6000) N,NNZ,LNA
      WRITE(6,6010)
      DO 130 I=1,NNZ
         WRITE(6,6020) VALUES(I)
  130 CONTINUE
      WRITE(6,6030)
      DO 140 I=1,N+1
         WRITE(6,6040) IPONTR(I)
  140 CONTINUE
      WRITE(6,6050)
      DO 150 I=1,NNZ
         WRITE(6,6040) IRWIND(I)
  150 CONTINUE
!
      M=1
      WRITE(6,6060) M
      ALLOCATE( A(LNA,M), STAT=KERR )
      IF( KERR .NE. 0 ) GOTO 160
      ALLOCATE( JA(LNA,M), STAT=KERR )
      IF( KERR .NE. 0 ) GOTO 170
      CALL DXA005&
      (ITYPE,N,VALUES,IPONTR,IRWIND,NNZ,A,LNA,M,JA,IWK,IERR)
      WRITE(6,6070) IERR
      IF( IERR .EQ. 2000 ) THEN
         WRITE(6,6080) M
         DEALLOCATE(JA)
         DEALLOCATE(A)
         ALLOCATE( A(LNA,M), STAT=KERR )
         IF( KERR .NE. 0) GOTO 160
         ALLOCATE( JA(LNA,M), STAT=KERR )
         IF( KERR .NE. 0) GOTO 170
         CALL DXA005&
         (ITYPE,N,VALUES,IPONTR,IRWIND,NNZ,A,LNA,M,JA,IWK,IERR)
         WRITE(6,6090) IERR
      ENDIF
!
      IF( IERR .GE. 2000 ) GOTO 9999
      WRITE(6,6100)
      DO 180 I=1,N
         WRITE(6,6110) (A(I,J),J=1,M)
  180 CONTINUE
      WRITE(6,6120)
      DO 190 I=1,N
         WRITE(6,6130) (JA(I,J),J=1,M)
  190 CONTINUE
```

```
!
 9999 CONTINUE
      DEALLOCATE(JA)
  170 CONTINUE
      DEALLOCATE(A)
  160 CONTINUE
!
      STOP
 6000 FORMAT(/,&
             1X,'***  DXA005  ***',/,/,&
             1X,' **  INPUT  **',/,/,&
             1X,'     N   = ',I5,/,&
             1X,'     NNZ = ',I5,/,&
             1X,'     LNA = ',I5)
 6010 FORMAT(/,&
             1X,'     VECTOR VALUES')
 6020 FORMAT(1X,'       ',F5.1)
 6030 FORMAT(/,&
             1X,'     VECTOR IPONTR')
 6040 FORMAT(1X,'       ',I5)
 6050 FORMAT(/,&
             1X,'     VECTOR IRWIND')
 6060 FORMAT(/,&
             1X,'     INPUT  M   = ',I5,/)
 6070 FORMAT(/,&
             1X,' **  OUTPUT  **',/,/,&
             1X,'     IERR( FIRST  CALL ) = ',I5,/)
 6080 FORMAT(1X,'     OUTPUT M   = ',I5,/)
 6090 FORMAT(1X,'     IERR( SECOND CALL ) = ',I5,/)
 6100 FORMAT(1X,'     MATRIX A')
 6110 FORMAT(1X,'    ',4(2X,F5.1))
 6120 FORMAT(/,&
             1X,'     MATRIX JA')
 6130 FORMAT(1X,'    ',4(2X,I5))
!
      END
```

(d) 出力結果

```
***  DXA005  ***

 **  INPUT  **

     N   =     5
     NNZ =    16
     LNA =     5

     VECTOR VALUES
         1.0
        -2.1
         4.5
         5.0
         1.1
         2.0
         2.1
         3.0
        -5.4
         1.2
         4.0
        -5.1
         1.4
         3.5
         4.1
         5.0

     VECTOR IPONTR
           1
           5
           7
          10
          13
          17

     VECTOR IRWIND
           1
           2
           4
           5
           1
           2
           2
           3
           5
           1
           4
           5
           1
           3
           4
           5

     INPUT  M   =     1


 **  OUTPUT  **

     IERR( FIRST  CALL ) =  2000

     OUTPUT M   =     4
```

```
IERR( SECOND CALL ) =     0

MATRIX A
   1.0    1.1    1.2    1.4
   2.0   -2.1    2.1    0.0
   3.0    3.5    0.0    0.0
   4.0    4.5    4.1    0.0
   5.0    5.0   -5.4   -5.1

MATRIX JA
     1      2      4      5
     2      1      3      0
     3      5      0      0
     4      1      5      0
     5      1      3      4
```

第 3 章　基本行列演算

3.1　概 要

本章では基本的な行列演算を行うサブルーチンについて説明する.

3.1.1　使用しているアルゴリズム

3.1.1.1　実行列の積の計算 (速度優先型)

A を $M \times N$ の実行列, B を $N \times L$ の実行列とし, その (i, j) 要素を それぞれ a_{ij}, b_{ij} とする. $C = A \cdot B$ を計算するのに行列の積の定義にしたがって,

$$c_{ik} = \sum_{j=1}^{N} a_{ij} \cdot b_{jk}$$

のように計算すると, $M \times L \times (2 \times N - 1)$ 回の浮動小数点演算が必要である. これを以下に述べるようにストラッセンのアルゴリズムを用いて計算することにより, 最大 12% 程度減らすことができる. いま, M, N, L がいずれも偶数である場合を考える. まず行列 A, B, C を行および列方向に 2 等分して次のように小行列に分割する.

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

ここで

$$M_1 = (A_{12} - A_{22})(B_{21} + B_{22})$$

$$M_2 = (A_{11} + A_{22})(B_{11} + B_{22})$$

$$M_3 = (A_{11} - A_{21})(B_{11} + B_{12})$$

$$M_4 = (A_{11} + A_{12})B_{22}$$

$$M_5 = A_{11}(B_{12} - B_{22})$$

$$M_6 = A_{22}(B_{21} - B_{11})$$

$$M_7 = (A_{21} + A_{22})B_{11}$$

と置くと, C の小行列が次のように計算される.

$$C_{11} = M_1 + M_2 - M_4 + M_6$$

$$C_{12} = M_4 + M_5$$

$$C_{21} = M_6 + M_7$$

$$C_{22} = M_2 - M_3 + M_5 - M_7$$

M, N, L のどれかが奇数の場合には, 初めに, 各行列に含まれる最大の偶数次の部分小行列について上記の方法で積を計算する. それから, その部分小行列に含まれない要素からの寄与を計算して補正を加える. また, 各小行列の積を求めるのにも上記の手続きを用いることにより, 演算回数をさらに減らすことができる. 本ライブラリでは最大 3 段までこの手続きを行っている.

3.2 基本演算

3.2.1 DAM1AD, RAM1AD

実行列 (2 次元配列型) の和 $(C = A + B)$

(1) **機能**

二つの実行列 A, B (2 次元配列型) の和 $(C = A + B)$ を求める.

(2) **使用法**

倍精度サブルーチン:

CALL DAM1AD (A, LMA, NM, NN, B, LMB, C, LMC, IERR)

単精度サブルーチン:

CALL RAM1AD (A, LMA, NM, NN, B, LMB, C, LMC, IERR)

(3) **引数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: $\left\{\begin{array}{l}\text{32 ビット整数版では INTEGER(4)}\\ \text{64 ビット整数版では INTEGER(8)}\end{array}\right\}$

項番	引数名	型	大きさ	入出力	内容
1	A	$\left\{\begin{matrix}\mathrm{D}\\ \mathrm{R}\end{matrix}\right\}$	LMA, NN	入力	実行列 A (2 次元配列型)
2	LMA	I	1	入力	配列 A の整合寸法
3	NM	I	1	入力	行列 A, B, C の行数
4	NN	I	1	入力	行列 A, B, C の列数
5	B	$\left\{\begin{matrix}\mathrm{D}\\ \mathrm{R}\end{matrix}\right\}$	LMB, NN	入力	実行列 B (2 次元配列型)
6	LMB	I	1	入力	配列 B の整合寸法
7	C	$\left\{\begin{matrix}\mathrm{D}\\ \mathrm{R}\end{matrix}\right\}$	LMC, NN	出力	行列 A, B の和 $(A + B)$ (2 次元配列型)
8	LMC	I	1	入力	配列 C の整合寸法
9	IERR	I	1	出力	エラーインディケータ

(4) **制限条件**

(a) $\mathrm{NN} > 0$

(b) $0 < \mathrm{NM} \leq \mathrm{LMA}, \mathrm{LMB}, \mathrm{LMC}$

(5) **エラーインディケータ**

IERR の値	意味	処理内容
0	正常終了.	
1000	NN = 1 であった.	処理を続ける.
3000	制限条件 (a) または (b) を満足しなかった.	処理を打ち切る.

(6) **注意事項**

なし

(7) **使用例**

(a) 問 題

行列 A, B が以下の時, $C = A + B$ を求める.

$$A = \begin{bmatrix} 1 & 2 & 0 & -1 \\ -3 & -5 & 1 & 2 \\ 1 & 3 & 2 & -2 \\ 0 & 2 & 1 & -1 \end{bmatrix}$$

$$B = \begin{bmatrix} -3 & -1 & 1 & -1 \\ -3 & -1 & 0 & 1 \\ -4 & -1 & 1 & 0 \\ -10 & -3 & 1 & 1 \end{bmatrix}$$

(b) 入力データ

行列 $A, B, \mathrm{LMA} = 11, \mathrm{LMB} = 11, \mathrm{LMC} = 11, \mathrm{NM} = 4, \mathrm{NN} = 4.$

(c) 主プログラム

```
      PROGRAM BAM1AD
! *** EXAMPLE OF DAM1AD ***
      IMPLICIT NONE
      INTEGER LMA,LMB,LMC,NM,NN
      PARAMETER( LMA=11, LMB=11, LMC=11 )
      PARAMETER( NM=4, NN=4 )
      INTEGER IERR,I,J
      REAL(8) A(LMA,NN),B(LMB,NN),C(LMC,NN)
!
      DO 100 I=1,NM
         READ(5,*) (A(I,J),J=1,NN)
  100 CONTINUE
      DO 110 I=1,NN
         READ(5,*) (B(I,J),J=1,NN)
  110 CONTINUE
!
      WRITE(6,6000) LMA,LMB,LMC,NM,NN
      DO 120 I=1,NM
         WRITE(6,6010) (A(I,J),J=1,NN)
  120 CONTINUE
      WRITE(6,6020)
      DO 130 I=1,NN
         WRITE(6,6010) (B(I,J),J=1,NN)
  130 CONTINUE
!
      CALL DAM1AD(A,LMA,NM,NN,B,LMB,C,LMC,IERR)
!
      WRITE(6,6030) IERR
      IF( IERR .GE. 3000 ) STOP
!
      WRITE(6,6040)
      DO 140 I=1,NM
         WRITE(6,6010) (C(I,J),J=1,NN)
  140 CONTINUE
      STOP
!
 6000 FORMAT(/,&
             1X,’***  DAM1AD  ***’,/,/,&
             1X,’ **  INPUT  **’,/,/,&
             1X,’     LMA=’,I2,’   LMB=’,I2,’   LMC=’,I2,/,/,&
             1X,’     NM =’,I2,’   NN =’,I2,/,/,&
             1X,’     INPUT MATRIX A’,/)
 6010 FORMAT(1X,6X,11(F7.1))
 6020 FORMAT(/,&
             1X,’     INPUT MATRIX B’,/)
 6030 FORMAT(/,&
             1X,’ **  OUTPUT  **’,/,/,&
             1X,’     IERR = ’,I4,/)
 6040 FORMAT(1X,’     OUTPUT MATRIX C’,/)
      END
```

(d) 出力結果

```
***  DAM1AD  ***

 **  INPUT  **

     LMA=11   LMB=11   LMC=11

     NM = 4   NN = 4
```

```
       INPUT MATRIX A

            1.0     2.0     0.0    -1.0
           -3.0    -5.0     1.0     2.0
            1.0     3.0     2.0    -2.0
            0.0     2.0     1.0    -1.0

       INPUT MATRIX B

           -3.0    -1.0     1.0    -1.0
           -3.0    -1.0     0.0     1.0
           -4.0    -1.0     1.0     0.0
          -10.0    -3.0     1.0     1.0

   **  OUTPUT  **

       IERR =    0

       OUTPUT MATRIX C

           -2.0     1.0     1.0    -2.0
           -6.0    -6.0     1.0     3.0
           -3.0     2.0     3.0    -2.0
          -10.0    -1.0     2.0     0.0
```

3.2.2 DAM1SB, RAM1SB
実行列 (2次元配列型) の差 $(C = A - B)$

(1) **機 能**

二つの実行列 A, B (2次元配列型) の差 $(C = A - B)$ を求める.

(2) **使用法**

倍精度サブルーチン:

CALL DAM1SB (A, LMA, NM, NN, B, LMB, C, LMC, IERR)

単精度サブルーチン:

CALL RAM1SB (A, LMA, NM, NN, B, LMB, C, LMC, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\left\{\begin{array}{l}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{array}\right\}$

項番	引数名	型	大きさ	入出力	内　容
1	A	$\left\{\begin{array}{c}\mathrm{D}\\\mathrm{R}\end{array}\right\}$	LMA, NN	入 力	実行列 A (2 次元配列型)
2	LMA	I	1	入 力	配列 A の整合寸法
3	NM	I	1	入 力	行列 A, B, C の行数
4	NN	I	1	入 力	行列 A, B, C の列数
5	B	$\left\{\begin{array}{c}\mathrm{D}\\\mathrm{R}\end{array}\right\}$	LMB, NN	入 力	実行列 B (2 次元配列型)
6	LMB	I	1	入 力	配列 B の整合寸法
7	C	$\left\{\begin{array}{c}\mathrm{D}\\\mathrm{R}\end{array}\right\}$	LMC, NN	出 力	行列 A, B の差 $(A - B)$ (2 次元配列型)
8	LMC	I	1	入 力	配列 C の整合寸法
9	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $\mathrm{NN} > 0$

(b) $0 < \mathrm{NM} \leq \mathrm{LMA}, \mathrm{LMB}, \mathrm{LMC}$

(5) **エラーインディケータ**

IERR の値	意　味	処 理 内 容
0	正常終了.	
1000	$\mathrm{NN} = 1$ であった.	処理を続ける.
3000	制限条件 (a) または (b) を満足しなかった.	処理を打ち切る.

(6) **注意事項**

なし

(7) **使用例**

(a) 問 題

行列 A, B が以下の時, $C = A - B$ を求める.

$$A = \begin{bmatrix} 1 & 2 & 0 & -1 \\ -3 & -5 & 1 & 2 \\ 1 & 3 & 2 & -2 \\ 0 & 2 & 1 & -1 \end{bmatrix}$$

$$B = \begin{bmatrix} -3 & -1 & 1 & -1 \\ -3 & -1 & 0 & 1 \\ -4 & -1 & 1 & 0 \\ -10 & -3 & 1 & 1 \end{bmatrix}$$

(b) 入力データ

行列 $A, B, \mathrm{LMA} = 11, \mathrm{LMB} = 11, \mathrm{LMC} = 11, \mathrm{NM} = 4, \mathrm{NN} = 4.$

(c) 主プログラム

```
      PROGRAM BAM1SB
! *** EXAMPLE OF DAM1SB ***
      IMPLICIT NONE
      INTEGER LMA,LMB,LMC,NM,NN
      PARAMETER( LMA=11, LMB=11, LMC=11 )
      PARAMETER( NM=4, NN=4 )
      INTEGER IERR,I,J
      REAL(8) A(LMA,NN),B(LMB,NN),C(LMC,NN)
!
      DO 100 I=1,NM
         READ(5,*) (A(I,J),J=1,NN)
  100 CONTINUE
      DO 110 I=1,NN
         READ(5,*) (B(I,J),J=1,NN)
  110 CONTINUE
!
      WRITE(6,6000) LMA,LMB,LMC,NM,NN
      DO 120 I=1,NM
         WRITE(6,6010) (A(I,J),J=1,NN)
  120 CONTINUE
      WRITE(6,6020)
      DO 130 I=1,NN
         WRITE(6,6010) (B(I,J),J=1,NN)
  130 CONTINUE
!
      CALL DAM1SB(A,LMA,NM,NN,B,LMB,C,LMC,IERR)
!
      WRITE(6,6030) IERR
      IF( IERR .GE. 3000 ) STOP
!
      WRITE(6,6040)
      DO 140 I=1,NM
         WRITE(6,6010) (C(I,J),J=1,NN)
  140 CONTINUE
      STOP
!
 6000 FORMAT(/,&
             1X,'***  DAM1SB  ***',/,/,&
             1X,' **  INPUT  **',/,/,&
             1X,'     LMA=',I2,'   LMB=',I2,'   LMC=',I2,/,/,&
             1X,'     NM =',I2,'   NN =',I2,/,/,&
             1X,'     INPUT MATRIX A',/)
 6010 FORMAT(1X,6X,11(F7.1))
 6020 FORMAT(/,&
             1X,'     INPUT MATRIX B',/)
 6030 FORMAT(/,&
             1X,' **  OUTPUT  **',/,/,&
             1X,'     IERR = ',I4,/)
 6040 FORMAT(1X,'     OUTPUT MATRIX C',/)
      END
```

(d) 出力結果

```
***  DAM1SB  ***

 **  INPUT  **

     LMA=11   LMB=11   LMC=11

     NM = 4   NN = 4

     INPUT MATRIX A

          1.0    2.0    0.0   -1.0
         -3.0   -5.0    1.0    2.0
          1.0    3.0    2.0   -2.0
          0.0    2.0    1.0   -1.0
```

```
    INPUT MATRIX B

        -3.0   -1.0    1.0   -1.0
        -3.0   -1.0    0.0    1.0
        -4.0   -1.0    1.0    0.0
       -10.0   -3.0    1.0    1.0

**  OUTPUT  **

    IERR =    0

    OUTPUT MATRIX C

         4.0    3.0   -1.0    0.0
         0.0   -4.0    1.0    1.0
         5.0    4.0    1.0   -2.0
        10.0    5.0    0.0   -2.0
```

3.2.3 DAM1MU, RAM1MU
実行列 (2 次元配列型) の積 $(C = AB)$

(1) **機 能**

二つの実行列 A, B (2 次元配列型) の積 $(C = AB)$ を求める.

(2) **使用法**

倍精度サブルーチン:

CALL DAM1MU (A, LMA, NM, NN, B, LNB, NL, C, LMC, IERR)

単精度サブルーチン:

CALL RAM1MU (A, LMA, NM, NN, B, LNB, NL, C, LMC, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\left\{\begin{array}{l}\text{32 ビット整数版では INTEGER(4)}\\ \text{64 ビット整数版では INTEGER(8)}\end{array}\right\}$

項番	引数名	型	大きさ	入出力	内　容
1	A	$\left\{\begin{matrix}\mathrm{D}\\ \mathrm{R}\end{matrix}\right\}$	LMA, NN	入 力	実行列 A (2 次元配列型)
2	LMA	I	1	入 力	配列 A の整合寸法
3	NM	I	1	入 力	行列 A の行数 (行列 C の行数)
4	NN	I	1	入 力	行列 A の列数 (行列 B の行数)
5	B	$\left\{\begin{matrix}\mathrm{D}\\ \mathrm{R}\end{matrix}\right\}$	LNB, NL	入 力	実行列 B (2 次元配列型)
6	LNB	I	1	入 力	配列 B の整合寸法
7	NL	I	1	入 力	行列 B の列数 (行列 C の列数)
8	C	$\left\{\begin{matrix}\mathrm{D}\\ \mathrm{R}\end{matrix}\right\}$	LMC, NL	出 力	行列 A, B の積 $(A \cdot B)$ (2 次元配列型)
9	LMC	I	1	入 力	配列 C の整合寸法
10	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{NM} \leq \mathrm{LMA}, \mathrm{LMC}$

(b) $0 < \mathrm{NN} \leq \mathrm{LNB}$

(c) $\mathrm{NL} > 0$

(5) **エラーインディケータ**

IERR の値	意　味	処 理 内 容
0	正常終了.	
1000	NN = 1 であった.	処理を続ける.
3000	制限条件 (a), (b) または (c) を満足しなかった.	処理を打ち切る.

(6) 注意事項

次数が大きいときには 3.2.4 $\left\{\begin{matrix}\text{DAM1MS}\\ \text{RAM1MS}\end{matrix}\right\}$ を使用されたい.

(7) 使用例

(a) 問 題

行列 A, B が以下の時, $C = AB$ を求める.

$$A = \begin{bmatrix} 1 & 2 & 0 & -1 \\ -3 & -5 & 1 & 2 \\ 1 & 3 & 2 & -2 \\ 0 & 2 & 1 & -1 \end{bmatrix}$$

$$B = \begin{bmatrix} -3 & -1 & 1 & -1 \\ -3 & -1 & 0 & 1 \\ -4 & -1 & 1 & 0 \\ -10 & -3 & 1 & 1 \end{bmatrix}$$

(b) 入力データ

行列 $A, B, \mathrm{LMA} = 11, \mathrm{LNB} = 11, \mathrm{LMC} = 11, \mathrm{NM} = 4, \mathrm{NN} = 4, \mathrm{NL} = 4.$

(c) 主プログラム

```
      PROGRAM BAM1MU
! *** EXAMPLE OF DAM1MU ***
      IMPLICIT NONE
      INTEGER LMA,LNB,LMC,NM,NN,NL
      PARAMETER( LMA=11, LNB=11, LMC=11 )
      PARAMETER( NM=4, NN=4, NL=4 )
      INTEGER IERR,I,J
      REAL(8) A(LMA,NN),B(LNB,NL),C(LMC,NL)
!
      DO 100 I=1,NM
         READ(5,*) (A(I,J),J=1,NN)
  100 CONTINUE
      DO 110 I=1,NN
         READ(5,*) (B(I,J),J=1,NL)
  110 CONTINUE
!
      WRITE(6,6000) LMA,LNB,LMC,NM,NN,NL
      DO 120 I=1,NM
         WRITE(6,6010) (A(I,J),J=1,NN)
  120 CONTINUE
      WRITE(6,6020)
      DO 130 I=1,NN
         WRITE(6,6010) (B(I,J),J=1,NL)
  130 CONTINUE
!
      CALL DAM1MU(A,LMA,NM,NN,B,LNB,NL,C,LMC,IERR)
!
      WRITE(6,6030) IERR
      IF( IERR .GE. 3000 ) STOP
!
      WRITE(6,6040)
      DO 140 I=1,NM
         WRITE(6,6010) (C(I,J),J=1,NL)
  140 CONTINUE
      STOP
!
 6000 FORMAT(/,&
             1X,’***  DAM1MU  ***’,/,/,&
             1X,’ **  INPUT  **’,/,/,&
             1X,’     LMA=’,I2,’   LNB=’,I2,’   LMC=’,I2,/,/,&
             1X,’     NM =’,I2,’   NN =’,I2,’   NL =’,I2,/,/,&
             1X,’     INPUT MATRIX A’,/)
 6010 FORMAT(1X,6X,11(F7.1))
 6020 FORMAT(/,&
             1X,’     INPUT MATRIX B’,/)
 6030 FORMAT(/,&
             1X,’ **  OUTPUT  **’,/,/,&
             1X,’     IERR = ’,I4,/)
 6040 FORMAT(1X,’     OUTPUT MATRIX C’,/)
      END
```

(d) 出力結果

```
***  DAM1MU  ***

 **  INPUT  **

     LMA=11   LNB=11   LMC=11
```

```
        NM = 4   NN = 4   NL = 4

        INPUT MATRIX A

            1.0    2.0    0.0   -1.0
           -3.0   -5.0    1.0    2.0
            1.0    3.0    2.0   -2.0
            0.0    2.0    1.0   -1.0

        INPUT MATRIX B

           -3.0   -1.0    1.0   -1.0
           -3.0   -1.0    0.0    1.0
           -4.0   -1.0    1.0    0.0
          -10.0   -3.0    1.0    1.0

    **  OUTPUT  **

        IERR =    0

        OUTPUT MATRIX C

            1.0    0.0    0.0    0.0
            0.0    1.0    0.0    0.0
            0.0    0.0    1.0    0.0
            0.0    0.0    0.0    1.0
```

3.2.4 DAM1MS, RAM1MS
実行列 (2 次元配列型) の積 ($C = AB$) の計算 (速度優先版)

(1) **機 能**

二つの実行列 A, B (2 次元配列型) の積 ($C = AB$) をストラッセンのアルゴリズムを用いて求める.

(2) **使用法**

倍精度サブルーチン:

CALL DAM1MS (A, LMA, M, N, B, LNB, K, C, LMC, W1, IERR)

単精度サブルーチン:

CALL RAM1MS (A, LMA, M, N, B, LNB, K, C, LMC, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\ \text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$

項番	引数名	型	大きさ	入出力	内　容
1	A	$\left\{\begin{matrix}\text{D}\\ \text{R}\end{matrix}\right\}$	LMA, N	入 力	実行列 A (2 次元配列型)
2	LMA	I	1	入 力	配列 A の整合寸法
3	M	I	1	入 力	行列 A の行数 (行列 C の行数)
4	N	I	1	入 力	行列 A の列数 (行列 B の行数)
5	B	$\left\{\begin{matrix}\text{D}\\ \text{R}\end{matrix}\right\}$	LNB, K	入 力	実行列 B (2 次元配列型)
6	LNB	I	1	入 力	配列 B の整合寸法
7	K	I	1	入 力	行列 B の列数 (行列 C の列数)
8	C	$\left\{\begin{matrix}\text{D}\\ \text{R}\end{matrix}\right\}$	LMC, K	出 力	実行列 A, B の積 ($A \cdot B$) (2 次元配列型)
9	LMC	I	1	入 力	配列 C の整合寸法
10	W1	$\left\{\begin{matrix}\text{D}\\ \text{R}\end{matrix}\right\}$	内容参照	ワーク	作業領域 **大きさ:** $((\mathrm{M} \times \mathrm{N} + \mathrm{N} \times \mathrm{K} + \mathrm{K} \times \mathrm{M})/3)$
11	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{M} \leq \mathrm{LMA}, \mathrm{LMC}$

(b) $0 < \mathrm{N} \leq \mathrm{LNB}$

(c) $\mathrm{K} > 0$

(5) **エラーインディケータ**

IERR の値	意　　　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	処理を続ける.
3000	制限条件 (a), (b) または (c) を満足しなかった.	処理を打ち切る.

(6) **注意事項**

(a) このサブルーチンは 3.2.3 $\left\{\begin{array}{l}\text{DAM1MU}\\\text{RAM1MU}\end{array}\right\}$ に較べて, 作業領域を使用する分, メモリを余計に使用するので, メモリを充分確保できない場合には 3.2.3 $\left\{\begin{array}{l}\text{DAM1MU}\\\text{RAM1MU}\end{array}\right\}$ の方を使用するとよい.

(b) このサブルーチンを用いて得られる結果は, 3.2.3 $\left\{\begin{array}{l}\text{DAM1MU}\\\text{RAM1MU}\end{array}\right\}$ を用いて得られる結果に較べて, 若干精度がおちる場合がある.

(7) **使用例**

(a) 問 題

行列 A, B が以下の時, $C = AB$ を求める.

$$A = \begin{bmatrix} 1 & 2 & 0 & -1 \\ -3 & -5 & 1 & 2 \\ 1 & 3 & 2 & -2 \\ 0 & 2 & 1 & -1 \end{bmatrix}$$

$$B = \begin{bmatrix} -3 & -1 & 1 & -1 \\ -3 & -1 & 0 & 1 \\ -4 & -1 & 1 & 0 \\ -10 & -3 & 1 & 1 \end{bmatrix}$$

(b) 入力データ

行列 $A, B, \mathrm{LMA} = 11, \mathrm{LNB} = 11, \mathrm{LMC} = 11, \mathrm{M} = 4, \mathrm{N} = 4, \mathrm{K} = 4.$

(c) 主プログラム

```
      PROGRAM BAM1MS
! *** EXAMPLE OF DAM1MS ***
      IMPLICIT NONE
      INTEGER LMA,LNB,LMC,M,N,K,IWK
      PARAMETER( LMA=11, LNB=11, LMC=11 )
      PARAMETER( M=4, N=4, K=4 )
      PARAMETER( IWK = (M*N+N*K+K*M)/3 )
      INTEGER IERR,I,J
      REAL(8) A(LMA,N),B(LNB,K),C(LMC,K),W1(IWK)
!
      DO 100 I=1,M
         READ(5,*) (A(I,J),J=1,N)
  100 CONTINUE
      DO 110 I=1,N
         READ(5,*) (B(I,J),J=1,K)
  110 CONTINUE
!
      WRITE(6,6000) LMA,LNB,LMC,M,N,K
      DO 120 I=1,M
         WRITE(6,6010) (A(I,J),J=1,N)
  120 CONTINUE
      WRITE(6,6020)
      DO 130 I=1,N
         WRITE(6,6010) (B(I,J),J=1,K)
  130 CONTINUE
!
      CALL DAM1MS(A,LMA,M,N,B,LNB,K,C,LMC,W1,IERR)
!
      WRITE(6,6030) IERR
```

```
      IF( IERR .GE. 3000 ) STOP
!
      WRITE(6,6040)
      DO 140 I=1,M
         WRITE(6,6010) (C(I,J),J=1,K)
  140 CONTINUE
      STOP
!
 6000 FORMAT(/,&
             1X,'***  DAM1MS  ***',/,/,&
             1X,' **  INPUT  **',/,/,&
             1X,'     LMA=',I2,'   LNB=',I2,'   LMC=',I2,/,/,&
             1X,'     M  =',I2,'   N  =',I2,'   K  =',I2,/,/,&
             1X,'     INPUT MATRIX A',/)
 6010 FORMAT(1X,6X,11(F7.1))
 6020 FORMAT(/,&
             1X,'     INPUT MATRIX B',/)
 6030 FORMAT(/,&
             1X,' **  OUTPUT  **',/,/,&
             1X,'     IERR = ',I4,/)
 6040 FORMAT(1X,'     OUTPUT MATRIX C',/)
      END
```

(d) 出力結果

```
***  DAM1MS  ***

 **  INPUT  **

     LMA=11   LNB=11   LMC=11

     M  = 4   N  = 4   K  = 4

     INPUT MATRIX A

         1.0    2.0    0.0   -1.0
        -3.0   -5.0    1.0    2.0
         1.0    3.0    2.0   -2.0
         0.0    2.0    1.0   -1.0

     INPUT MATRIX B

        -3.0   -1.0    1.0   -1.0
        -3.0   -1.0    0.0    1.0
        -4.0   -1.0    1.0    0.0
       -10.0   -3.0    1.0    1.0

 **  OUTPUT  **

     IERR =    0

     OUTPUT MATRIX C

         1.0    0.0    0.0    0.0
         0.0    1.0    0.0    0.0
         0.0    0.0    1.0    0.0
         0.0    0.0    0.0    1.0
```

3.2.5 DAMT1M, RAMT1M
実行列 (2 次元配列型) とその転置行列の積 ($B = AA^T$)

(1) **機 能**

実行列 A (2 次元配列型) とその転置行列との積 ($B = AA^T$) を求める.

(2) **使用法**

倍精度サブルーチン:

CALL DAMT1M (A, LMA, NM, NN, B, LMB, IERR)

単精度サブルーチン:

CALL RAMT1M (A, LMA, NM, NN, B, LMB, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型　I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$
R:単精度実数型　C:単精度複素数型

項番	引数名	型	大きさ	入出力	内　　容
1	A	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	LMA, NN	入 力	実行列 A (2 次元配列型)
2	LMA	I	1	入 力	配列 A の整合寸法
3	NM	I	1	入 力	行列 A の行数
4	NN	I	1	入 力	行列 A の列数
5	B	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	LMB, NM	出 力	行列 A とその転置行列の積 ($A \cdot A^T$)(2 次元配列型)
6	LMB	I	1	入 力	配列 B の整合寸法
7	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) NN > 0, NM > 0

(b) NM $\leq$ LMA, LMB

(5) **エラーインディケータ**

IERR の値	意　　味	処 理 内 容
0	正常終了.	
1000	NN $= 1$ であった.	処理を続ける.
3000	制限条件 (a) または (b) を満足しなかった.	処理を打ち切る.

(6) **注意事項**

なし

(7) **使用例**

(a) 問 題

行列 A が以下の時, $B = AA^T$ を求める.

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ -2 & 3 & 4 & 5 \\ -3 & -4 & 5 & 6 \\ -4 & -5 & -6 & 7 \end{bmatrix}$$

(b) 入力データ

行列 $A, \mathrm{LMA} = 11, \mathrm{LMB} = 11, \mathrm{NM} = 4, \mathrm{NN} = 4.$

(c) 主プログラム

```
      PROGRAM BAMT1M
! *** EXAMPLE OF DAMT1M ***
      IMPLICIT NONE
      INTEGER LMA,LMB,NM,NN
      PARAMETER( LMA=11, LMB=11 )
      PARAMETER( NM=4, NN=4 )
      INTEGER IERR,I,J
      REAL(8) A(LMA,NN),B(LMB,NM)
!
      DO 100 I=1,NM
         READ(5,*) (A(I,J),J=1,NN)
  100 CONTINUE
!
      WRITE(6,6000) LMA,LMB,NM,NN
      DO 110 I=1,NM
         WRITE(6,6010) (A(I,J),J=1,NN)
  110 CONTINUE
!
      CALL DAMT1M(A,LMA,NM,NN,B,LMB,IERR)
!
      WRITE(6,6020) IERR
      IF( IERR .GE. 3000 ) STOP
!
      WRITE(6,6030)
      DO 120 I=1,NM
         WRITE(6,6010) (B(I,J),J=1,NM)
  120 CONTINUE
      STOP
!
 6000 FORMAT(/,&
             1X,'***  DAMT1M  ***',/,/,&
             1X,' **  INPUT  **',/,/,&
             1X,'     LMA=',I2,'   LMB=',I2,'   NM=',I2,'   NN=',I2,/,/,&
             1X,'     INPUT MATRIX A',/)
 6010 FORMAT(1X,6X,11(F7.1))
 6020 FORMAT(/,&
             1X,' **  OUTPUT  **',/,/,&
             1X,'     IERR = ',I4,/)
 6030 FORMAT(1X,'     OUTPUT MATRIX B',/)
      END
```

(d) 出力結果

```
***  DAMT1M  ***

 **  INPUT  **

     LMA=11   LMB=11   NM= 4   NN= 4

     INPUT MATRIX A

         1.0    2.0    3.0    4.0
        -2.0    3.0    4.0    5.0
        -3.0   -4.0    5.0    6.0
        -4.0   -5.0   -6.0    7.0

 **  OUTPUT  **

     IERR =    0

     OUTPUT MATRIX B

        30.0   36.0   28.0   -4.0
        36.0   54.0   44.0    4.0
        28.0   44.0   86.0   44.0
        -4.0    4.0   44.0  126.0
```

3.2.6 DATM1M, RATM1M
実行列 (2 次元配列型) の転置行列と元の行列の積 $(B = A^T A)$

(1) 機 能

実行列 A (2 次元配列型) の転置行列と元の行列との積 $(B = A^T A)$ を求める.

(2) 使用法

倍精度サブルーチン:

CALL DATM1M (A, LMA, NM, NN, B, LNB, IERR)

単精度サブルーチン:

CALL RATM1M (A, LMA, NM, NN, B, LNB, IERR)

(3) 引 数

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$

項番	引数名	型	大きさ	入出力	内　　　容
1	A	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	LMA, NN	入 力	実行列 A (2 次元配列型)
2	LMA	I	1	入 力	配列 A の整合寸法
3	NM	I	1	入 力	行列 A の行数
4	NN	I	1	入 力	行列 A の列数
5	B	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	LNB, NN	出 力	行列 A の転置行列と元の行列との積 $(A^T \cdot A)$
6	LNB	I	1	入 力	配列 B の整合寸法
7	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a) $0 <$ NN $\leq$ LNB

(b) $0 <$ NM $\leq$ LMA

(5) エラーインディケータ

IERR の値	意　　　味	処 理 内 容
0	正常終了.	
1000	NN = 1 であった.	処理を続ける.
3000	制限条件 (a) または (b) を満足しなかった.	処理を打ち切る.

(6) 注意事項

なし

(7) **使用例**

(a) 問 題

行列 A が以下の時, $B = A^TA$ を求める.

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ -2 & 3 & 4 & 5 \\ -3 & -4 & 5 & 6 \\ -4 & -5 & -6 & 7 \end{bmatrix}$$

(b) 入力データ

行列 $A, \mathrm{LMA} = 11, \mathrm{LNB} = 11, \mathrm{NM} = 4, \mathrm{NN} = 4.$

(c) 主プログラム

```
      PROGRAM BATM1M
! *** EXAMPLE OF DATM1M ***
      IMPLICIT NONE
      INTEGER LMA,LNB,NM,NN
      PARAMETER( LMA=11, LNB=11 )
      PARAMETER( NM=4, NN=4 )
      INTEGER IERR,I,J
      REAL(8) A(LMA,NN),B(LNB,NN)
!
      DO 100 I=1,NM
         READ(5,*) (A(I,J),J=1,NN)
  100 CONTINUE
!
      WRITE(6,6000) LMA,LNB,NM,NN
      DO 110 I=1,NM
         WRITE(6,6010) (A(I,J),J=1,NN)
  110 CONTINUE
!
      CALL DATM1M(A,LMA,NM,NN,B,LNB,IERR)
!
      WRITE(6,6020) IERR
      IF( IERR .GE. 3000 ) STOP
!
      WRITE(6,6030)
      DO 120 I=1,NN
         WRITE(6,6010) (B(I,J),J=1,NN)
  120 CONTINUE
      STOP
!
 6000 FORMAT(/,&
             1X,'***  DATM1M  ***',/,/,&
             1X,' **  INPUT  **',/,/,&
             1X,'     LMA=',I2,'   LNB=',I2,'   NM=',I2,'   NN=',I2,/,/,&
             1X,'     INPUT MATRIX A',/)
 6010 FORMAT(1X,6X,11(F7.1))
 6020 FORMAT(/,&
             1X,' **  OUTPUT  **',/,/,&
             1X,'     IERR = ',I4,/)
 6030 FORMAT(1X,'     OUTPUT MATRIX B',/)
      END
```

(d) 出力結果

```
***  DATM1M  ***

 **  INPUT  **

     LMA=11   LNB=11   NM= 4   NN= 4

     INPUT MATRIX A

         1.0    2.0    3.0    4.0
        -2.0    3.0    4.0    5.0
        -3.0   -4.0    5.0    6.0
        -4.0   -5.0   -6.0    7.0

 **  OUTPUT  **

     IERR =    0

     OUTPUT MATRIX B

        30.0   28.0    4.0  -52.0
        28.0   54.0   28.0  -36.0
         4.0   28.0   86.0   20.0
       -52.0  -36.0   20.0  126.0
```

3.2.7 DAM1MM, RAM1MM
実行列 (2次元配列型) の積 $(C = C \pm AB)$

(1) 機 能

実行列 (2次元配列型) の行列積 $(C = [C\pm]AB)$ を求める.

(2) 使用法

倍精度サブルーチン:

CALL DAM1MM (A, LMA, NM, NN, B, LNB, NL, C, LMC, ISW, IERR)

単精度サブルーチン:

CALL RAM1MM (A, LMA, NM, NN, B, LNB, NL, C, LMC, ISW, IERR)

(3) 引 数

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: {32ビット整数版ではINTEGER(4) / 64ビット整数版ではINTEGER(8)}

項番	引数名	型	大きさ	入出力	内容
1	A	{D R}	LMA, NN	入 力	実行列 A (2次元配列型)
2	LMA	I	1	入 力	配列 A の整合寸法
3	NM	I	1	入 力	行列 A の行数 (行列 C の行数)
4	NN	I	1	入 力	行列 A の列数 (行列 B の行数)
5	B	{D R}	LNB, NL	入 力	実行列 B (2次元配列型)
6	LNB	I	1	入 力	配列 B の整合寸法
7	NL	I	1	入 力	行列 B の列数 (行列 C の列数)
8	C	{D R}	LMC, NL	入 力	初期実行列 C (ISW= ±1 の時)(2次元配列型)
				出 力	行列積 $(C = [C\pm]AB)$
9	LMC	I	1	入 力	配列 C の整合寸法
10	ISW	I	1	入 力	処理スイッチ ISW = 1 の時: $C = C + AB$ ISW = 0 の時: $C = AB$ ISW = −1 の時: $C = C - AB$
11	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a) $0 < \mathrm{NM} \leq \mathrm{LMA}, \mathrm{LMC}$

(b) $0 < \mathrm{NN} \leq \mathrm{LNB}$

(c) $\mathrm{NL} > 0$

(d) $\mathrm{ISW} \in \{0, 1, -1\}$

(5) **エラーインディケータ**

IERR の値	意　　　　味	処　理　内　容
0	正常終了.	
1000	NN = 1 であった.	処理を続ける.
3000	制限条件 (a), (b) または (c) を満足しなかった.	処理を打ち切る.
3010	制限条件 (d) を満足しなかった.	

(6) **注意事項**

なし.

(7) **使用例**

(a) 問 題

行列 A, B が以下の時, $C = AB$ を求める.

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 & 5 \end{bmatrix}$$

(b) 入力データ

行列 $A, B, \mathrm{LMA} = 11, \mathrm{LNB} = 11, \mathrm{LNC} = 11, \mathrm{NM} = 4, \mathrm{NN} = 5, \mathrm{NL} = 6, \mathrm{ISW} = 0.$

(c) 主プログラム

```
      PROGRAM BAM1MM
! *** EXAMPLE OF DAM1MM ***
      IMPLICIT NONE
      INTEGER LMA,LNB,LMC,ISW,NM,NN,NL
      PARAMETER( LMA=11, LNB=11, LMC=11, ISW=0 )
      PARAMETER( NM=4, NN=5, NL=6 )
      INTEGER IERR,I,J
      REAL(8) A(LMA,NN),B(LNB,NL),C(LMC,NL)
!
      DO 100 I=1,NM
         READ(5,*) (A(I,J),J=1,NN)
  100 CONTINUE
      DO 110 I=1,NN
         READ(5,*) (B(I,J),J=1,NL)
  110 CONTINUE
!
      WRITE(6,6000) LMA,LNB,LMC,NM,NN,NL,ISW
      WRITE(6,6010) 'MATRIX A'
      DO 120 I=1,NM
         WRITE(6,6020) (A(I,J),J=1,NN)
  120 CONTINUE
      WRITE(6,6010) 'MATRIX B'
      DO 130 I=1,NN
         WRITE(6,6020) (B(I,J),J=1,NL)
  130 CONTINUE
!
      CALL DAM1MM(A,LMA,NM,NN,B,LNB,NL,C,LMC,ISW,IERR)
!
      WRITE(6,6030) IERR
      IF( IERR .GE. 3000 ) STOP
!
      WRITE(6,6040) 'MATRIX C'
      DO 140 I=1,NM
         WRITE(6,6020) (C(I,J),J=1,NL)
  140 CONTINUE
      STOP
!
 6000 FORMAT(/,&
             1X,'***  DAM1MM  ***',/,/,&
```

```
             1X,' **  INPUT  **',/,/,&
             1X,'     LMA=',I2,'   LNB=',I2,'   LMC=',I2,/,/,&
             1X,'     NM =',I2,'   NN =',I2,'   NL =',I2,/,/,&
             1X,'     ISW=',I2,/,/,&
             1X,'     INPUT MATRIX')
 6010 FORMAT(/,&
             1X,5X,A)
 6020 FORMAT(1X,6X,11(F7.1))
 6030 FORMAT(/,&
             1X,' **  OUTPUT  **',/,/,&
             1X,'     IERR = ',I4,/)
 6040 FORMAT(1X,'     OUTPUT MATRIX',/,/,&
             1X,5X,A)
      END
```

(d) 出力結果

```
***  DAM1MM  ***

 **  INPUT  **

     LMA=11   LNB=11   LMC=11

     NM = 4   NN = 5   NL = 6

     ISW= 0

     INPUT MATRIX

     MATRIX A
          1.0    1.0    1.0    1.0    1.0
          2.0    2.0    2.0    2.0    2.0
          3.0    3.0    3.0    3.0    3.0
          4.0    4.0    4.0    4.0    4.0

     MATRIX B
          1.0    1.0    1.0    1.0    1.0    1.0
          2.0    2.0    2.0    2.0    2.0    2.0
          3.0    3.0    3.0    3.0    3.0    3.0
          4.0    4.0    4.0    4.0    4.0    4.0
          5.0    5.0    5.0    5.0    5.0    5.0

 **  OUTPUT  **

     IERR =    0

     OUTPUT MATRIX

     MATRIX C
         15.0   15.0   15.0   15.0   15.0   15.0
         30.0   30.0   30.0   30.0   30.0   30.0
         45.0   45.0   45.0   45.0   45.0   45.0
         60.0   60.0   60.0   60.0   60.0   60.0
```

3.2.8 DAM1MT, RAM1MT
実行列 (2 次元配列型) の積 ($C = C \pm AB^T$)

(1) **機 能**

実行列 (2 次元配列型) の行列積 ($C = [C\pm]AB^T$) を求める.

(2) **使用法**

倍精度サブルーチン:

CALL DAM1MT (A, LMA, NM, NN, B, LLB, NL, C, LMC, ISW, IERR)

単精度サブルーチン:

CALL RAM1MT (A, LMA, NM, NN, B, LLB, NL, C, LMC, ISW, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\ \text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$

項番	引数名	型	大きさ	入出力	内　容
1	A	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	LMA, NN	入 力	実行列 A (2 次元配列型)
2	LMA	I	1	入 力	配列 A の整合寸法
3	NM	I	1	入 力	行列 A の行数 (行列 C の行数)
4	NN	I	1	入 力	行列 A の列数 (行列 B の列数)
5	B	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	LLB, NN	入 力	実行列 B (2 次元配列型)
6	LLB	I	1	入 力	配列 B の整合寸法
7	NL	I	1	入 力	行列 B の行数 (行列 C の列数)
8	C	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	LMC, NL	入 力	初期実行列 C (ISW $= \pm 1$ の時)(2 次元配列型)
				出 力	行列積 ($C = [C\pm]AB^T$)
9	LMC	I	1	入 力	配列 C の整合寸法
10	ISW	I	1	入 力	処理スイッチ ISW $= 1$ の時: $C = C + AB^T$ ISW $= 0$ の時: $C = AB^T$ ISW $= -1$ の時: $C = C - AB^T$
11	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{NM} \leq \mathrm{LMA}, \mathrm{LMC}$

(b) $0 < \mathrm{NL} \leq \mathrm{LLB}$

(c) $\mathrm{NN} > 0$

(d) $\mathrm{ISW} \in \{0, 1, -1\}$

(5) **エラーインディケータ**

IERR の値	意　　　　　　味	処　理　内　容
0	正常終了.	
1000	NN = 1 であった.	処理を続ける.
3000	制限条件 (a), (b) または (c) を満足しなかった.	処理を打ち切る.
3010	制限条件 (d) を満足しなかった.	

(6) **注意事項**

なし.

(7) **使用例**

(a) 問 題

行列 A, B が以下の時, $C = AB^T$ を求める.

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix}$$

(b) 入力データ

行列 $A, B, \mathrm{LMA} = 11, \mathrm{LLB} = 11, \mathrm{LNC} = 11, \mathrm{NM} = 4, \mathrm{NN} = 5, \mathrm{NL} = 5, \mathrm{ISW} = 0.$

(c) 主プログラム

```
      PROGRAM BAM1MT
! *** EXAMPLE OF DAM1MT ***
      IMPLICIT NONE
      INTEGER LMA,LLB,LMC,ISW,NM,NN,NL
      PARAMETER( LMA=11, LLB=11, LMC=11, ISW=0 )
      PARAMETER( NM=4, NN=5, NL=5 )
      INTEGER IERR,I,J
      REAL(8) A(LMA,NN),B(LLB,NN),C(LMC,NL)
!
      DO 100 I=1,NM
         READ(5,*) (A(I,J),J=1,NN)
  100 CONTINUE
      DO 110 I=1,NL
         READ(5,*) (B(I,J),J=1,NN)
  110 CONTINUE
!
      WRITE(6,6000) LMA,LLB,LMC,NM,NN,NL,ISW
      WRITE(6,6010) ’MATRIX A’
      DO 120 I=1,NM
         WRITE(6,6020) (A(I,J),J=1,NN)
  120 CONTINUE
      WRITE(6,6010) ’MATRIX B’
      DO 130 I=1,NL
         WRITE(6,6020) (B(I,J),J=1,NN)
  130 CONTINUE
!
      CALL DAM1MT(A,LMA,NM,NN,B,LLB,NL,C,LMC,ISW,IERR)
!
      WRITE(6,6030) IERR
      IF( IERR .GE. 3000 ) STOP
!
      WRITE(6,6040) ’MATRIX C’
      DO 140 I=1,NM
         WRITE(6,6020) (C(I,J),J=1,NL)
  140 CONTINUE
      STOP
!
 6000 FORMAT(/,&
            1X,’***  DAM1MT  ***’,/,/,&
```

```
              1X,' **  INPUT  **',/,/,&
              1X,'     LMA=',I2,'   LLB=',I2,'   LMC=',I2,/,/,&
              1X,'     NM =',I2,'   NN =',I2,'   NL =',I2,/,/,&
              1X,'     ISW=',I2,/,/,&
              1X,'     INPUT MATRIX')
 6010 FORMAT(/,&
              1X,5X,A)
 6020 FORMAT(1X,6X,11(F7.1))
 6030 FORMAT(/,&
              1X,' **  OUTPUT  **',/,/,&
              1X,'     IERR = ',I4,/)
 6040 FORMAT(1X,'     OUTPUT MATRIX',/,/,&
              1X,5X,A)
      END
```

(d) 出力結果

```
 ***  DAM1MT  ***

  **  INPUT  **

      LMA=11   LLB=11   LMC=11

      NM = 4   NN = 5   NL = 5

      ISW= 0

      INPUT MATRIX

      MATRIX A
           1.0    1.0    1.0    1.0    1.0
           2.0    2.0    2.0    2.0    2.0
           3.0    3.0    3.0    3.0    3.0
           4.0    4.0    4.0    4.0    4.0

      MATRIX B
           1.0    1.0    1.0    1.0    1.0
           2.0    2.0    2.0    2.0    2.0
           3.0    3.0    3.0    3.0    3.0
           4.0    4.0    4.0    4.0    4.0
           5.0    5.0    5.0    5.0    5.0

  **  OUTPUT  **

      IERR =    0

      OUTPUT MATRIX

      MATRIX C
           5.0   10.0   15.0   20.0   25.0
          10.0   20.0   30.0   40.0   50.0
          15.0   30.0   45.0   60.0   75.0
          20.0   40.0   60.0   80.0  100.0
```

3.2.9 DAM1TM, RAM1TM
実行列 (2 次元配列型) の積 ($C = C \pm A^T B$)

(1) **機 能**

実行列の行列積 ($C = [C\pm]A^T B$) を求める.

(2) **使用法**

倍精度サブルーチン:

CALL DAM1TM (A, LNA, NM, NN, B, LNB, NL, C, LMC, ISW, IERR)

単精度サブルーチン:

CALL RAM1TM (A, LNA, NM, NN, B, LNB, NL, C, LMC, ISW, IERR)

(3) **引 数**

D:倍精度実数型 Z:倍精度複素数型 I: {32 ビット整数版では INTEGER(4)
R:単精度実数型 C:単精度複素数型 64 ビット整数版では INTEGER(8)}

項番	引数名	型	大きさ	入出力	内 容
1	A	D R	LNA, NM	入 力	実行列 A (2 次元配列型)
2	LNA	I	1	入 力	配列 A の整合寸法
3	NM	I	1	入 力	行列 A の列数 (行列 C の行数)
4	NN	I	1	入 力	行列 A の行数 (行列 B の行数)
5	B	D R	LNB, NL	入 力	実行列 B (2 次元配列型)
6	LNB	I	1	入 力	配列 B の整合寸法
7	NL	I	1	入 力	行列 B の列数 (行列 C の列数)
8	C	D R	LMC, NL	入 力	初期実行列 C (ISW $= \pm 1$ の時)(2 次元配列型)
				出 力	行列積 ($C = [C\pm]A^T B$)
9	LMC	I	1	入 力	配列 C の整合寸法
10	ISW	I	1	入 力	処理スイッチ ISW $= 1$ の時: $C = C + A^T B$ ISW $= 0$ の時: $C = A^T B$ ISW $= -1$ の時: $C = C - A^T B$
11	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{NM} \leq \mathrm{LMC}$

(b) $0 < \mathrm{NN} \leq \mathrm{LNA}, \mathrm{LNB}$

(c) $\mathrm{NL} > 0$

(d) $\mathrm{ISW} \in \{0, 1, -1\}$

(5) エラーインディケータ

IERR の値	意　　　　味	処　理　内　容
0	正常終了.	
1000	NN = 1 であった.	処理を続ける.
3000	制限条件 (a), (b) または (c) を満足しなかった.	処理を打ち切る.
3010	制限条件 (d) を満足しなかった.	

(6) 注意事項

なし.

(7) 使用例

(a) 問 題

行列 A, B が以下の時, $C = A^T B$ を求める.

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 \end{bmatrix}$$

(b) 入力データ

行列 $A, B, \mathrm{LNA} = 11, \mathrm{LNB} = 11, \mathrm{LNC} = 11, \mathrm{NM} = 5, \mathrm{NN} = 5, \mathrm{NL} = 4, \mathrm{ISW} = 0.$

(c) 主プログラム

```
      PROGRAM BAM1TM
! *** EXAMPLE OF DAM1TM ***
      IMPLICIT NONE
      INTEGER LNA,LNB,LMC,ISW,NM,NN,NL
      PARAMETER( LNA=11, LNB=11, LMC=11, ISW=0 )
      PARAMETER( NM=5, NN=5, NL=4 )
      INTEGER IERR,I,J
      REAL(8) A(LNA,NM),B(LNB,NL),C(LMC,NL)
!
      DO 100 I=1,NN
         READ(5,*) (A(I,J),J=1,NM)
  100 CONTINUE
      DO 110 I=1,NN
         READ(5,*) (B(I,J),J=1,NL)
  110 CONTINUE
!
      WRITE(6,6000) LNA,LNB,LMC,NM,NN,NL,ISW
      WRITE(6,6010) ’MATRIX A’
      DO 120 I=1,NN
         WRITE(6,6020) (A(I,J),J=1,NM)
  120 CONTINUE
      WRITE(6,6010) ’MATRIX B’
      DO 130 I=1,NN
         WRITE(6,6020) (B(I,J),J=1,NL)
  130 CONTINUE
!
      CALL DAM1TM(A,LNA,NM,NN,B,LNB,NL,C,LMC,ISW,IERR)
!
      WRITE(6,6030) IERR
      IF( IERR .GE. 3000 ) STOP
!
      WRITE(6,6040) ’MATRIX C’
      DO 140 I=1,NM
         WRITE(6,6020) (C(I,J),J=1,NL)
  140 CONTINUE
      STOP
!
```

```
6000 FORMAT(/,&
            1X,'***  DAM1TM  ***',/,/,&
            1X,' **  INPUT  **',/,/,&
            1X,'     LNA=',I2,'   LNB=',I2,'   LMC=',I2,/,/,&
            1X,'     NM =',I2,'   NN =',I2,'   NL =',I2,/,/,&
            1X,'     ISW=',I2,/,/,&
            1X,'     INPUT MATRIX')
6010 FORMAT(/,&
            1X,5X,A)
6020 FORMAT(1X,6X,11(F7.1))
6030 FORMAT(/,&
            1X,' **  OUTPUT  **',/,/,&
            1X,'     IERR = ',I4,/)
6040 FORMAT(1X,'     OUTPUT MATRIX',/,/,&
            1X,5X,A)
     END
```

(d) 出力結果

```
***  DAM1TM  ***

 **  INPUT  **

     LNA=11   LNB=11   LMC=11

     NM = 5   NN = 5   NL = 4

     ISW= 0

     INPUT MATRIX

     MATRIX A
          1.0    1.0    1.0    1.0    1.0
          2.0    2.0    2.0    2.0    2.0
          3.0    3.0    3.0    3.0    3.0
          4.0    4.0    4.0    4.0    4.0
          5.0    5.0    5.0    5.0    5.0

     MATRIX B
          1.0    1.0    1.0    1.0
          2.0    2.0    2.0    2.0
          3.0    3.0    3.0    3.0
          4.0    4.0    4.0    4.0
          5.0    5.0    5.0    5.0

 **  OUTPUT  **

     IERR =    0

     OUTPUT MATRIX

     MATRIX C
         55.0   55.0   55.0   55.0
         55.0   55.0   55.0   55.0
         55.0   55.0   55.0   55.0
         55.0   55.0   55.0   55.0
         55.0   55.0   55.0   55.0
```

3.2.10 DAM1TT, RAM1TT
実行列 (2 次元配列型) の積 ($C = C \pm A^T B^T$)

(1) **機 能**

実行列の行列積 ($C = [C\pm]A^T B^T$) を求める.

(2) **使用法**

倍精度サブルーチン:

CALL DAM1TT (A, LNA, NM, NN, B, LLB, NL, C, LMC, ISW, IERR)

単精度サブルーチン:

CALL RAM1TT (A, LNA, NM, NN, B, LLB, NL, C, LMC, ISW, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型　I: { 32 ビット整数版では INTEGER(4)
R:単精度実数型　C:単精度複素数型　　　64 ビット整数版では INTEGER(8) }

項番	引数名	型	大きさ	入出力	内　容
1	A	D R	LNA, NM	入 力	実行列 A (2 次元配列型)
2	LNA	I	1	入 力	配列 A の整合寸法
3	NM	I	1	入 力	行列 A の列数 (行列 C の行数)
4	NN	I	1	入 力	行列 A の行数 (行列 B の列数)
5	B	D R	LLB, NN	入 力	実行列 B (2 次元配列型)
6	LLB	I	1	入 力	配列 B の整合寸法
7	NL	I	1	入 力	行列 B の行数 (行列 C の列数)
8	C	D R	LMC, NL	入 力	初期実行列 C (ISW $= \pm 1$ の時)(2 次元配列型)
				出 力	行列積 ($C = [C\pm]A^T B^T$)
9	LMC	I	1	入 力	配列 C の整合寸法
10	ISW	I	1	入 力	処理スイッチ ISW $= 1$ の時: $C = C + A^T B^T$ ISW $= 0$ の時: $C = A^T B^T$ ISW $= -1$ の時: $C = C - A^T B^T$
11	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 <$ NM $\leq$ LMC

(b) $0 <$ NN $\leq$ LNA

(c) $0 <$ NL $\leq$ LNB

(d) ISW $\in \{0, 1, -1\}$

(5) **エラーインディケータ**

IERR の値	意　　　　味	処　理　内　容
0	正常終了.	
1000	NN = 1 であった.	処理を続ける.
3000	制限条件 (a), (b) または (c) を満足しなかった.	処理を打ち切る.
3010	制限条件 (d) を満足しなかった.	

(6) **注意事項**

なし.

(7) **使用例**

(a) 問 題

行列 A, B が以下の時, $C = A^T B^T$ を求める.

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \end{bmatrix}$$

(b) 入力データ

行列 $A, B, \mathrm{LNA} = 11, \mathrm{LLB} = 11, \mathrm{LNC} = 11, \mathrm{NM} = 5, \mathrm{NN} = 5, \mathrm{NL} = 4, \mathrm{ISW} = 0.$

(c) 主プログラム

```
      PROGRAM BAM1TT
! *** EXAMPLE OF DAM1TT ***
      IMPLICIT NONE
      INTEGER LNA,LLB,LMC,ISW,NM,NN,NL
      PARAMETER( LNA=11, LLB=11, LMC=11, ISW=0 )
      PARAMETER( NM=5, NN=5, NL=4 )
      INTEGER IERR,I,J
      REAL(8) A(LNA,NM),B(LLB,NN),C(LMC,NL)
!
      DO 100 I=1,NN
         READ(5,*) (A(I,J),J=1,NM)
  100 CONTINUE
      DO 110 I=1,NL
         READ(5,*) (B(I,J),J=1,NN)
  110 CONTINUE
!
      WRITE(6,6000) LNA,LLB,LMC,NM,NN,NL,ISW
      WRITE(6,6010) ’MATRIX A’
      DO 120 I=1,NN
         WRITE(6,6020) (A(I,J),J=1,NM)
  120 CONTINUE
      WRITE(6,6010) ’MATRIX B’
      DO 130 I=1,NL
         WRITE(6,6020) (B(I,J),J=1,NN)
  130 CONTINUE
!
      CALL DAM1TT(A,LNA,NM,NN,B,LLB,NL,C,LMC,ISW,IERR)
!
      WRITE(6,6030) IERR
      IF( IERR .GE. 3000 ) STOP
!
      WRITE(6,6040) ’MATRIX C’
      DO 140 I=1,NM
         WRITE(6,6020) (C(I,J),J=1,NL)
  140 CONTINUE
      STOP
!
 6000 FORMAT(/,&
             1X,’***  DAM1TT  ***’,/,/,&
```

```
              1X,' **  INPUT  **',/,/,&
              1X,'     LNA=',I2,'   LLB=',I2,'   LMC=',I2,/,/,&
              1X,'     NM =',I2,'   NN =',I2,'   NL =',I2,/,/,&
              1X,'     ISW=',I2,/,/,&
              1X,'     INPUT MATRIX')
 6010 FORMAT(/,&
              1X,5X,A)
 6020 FORMAT(1X,6X,11(F7.1))
 6030 FORMAT(/,&
              1X,' **  OUTPUT  **',/,/,&
              1X,'     IERR = ',I4,/)
 6040 FORMAT(1X,'     OUTPUT MATRIX',/,/,&
              1X,5X,A)
      END
```

(d) 出力結果

```
 ***  DAM1TT  ***

  **  INPUT  **

      LNA=11   LLB=11   LMC=11

      NM = 5   NN = 5   NL = 4

      ISW= 0

      INPUT MATRIX

      MATRIX A
           1.0    1.0    1.0    1.0    1.0
           2.0    2.0    2.0    2.0    2.0
           3.0    3.0    3.0    3.0    3.0
           4.0    4.0    4.0    4.0    4.0
           5.0    5.0    5.0    5.0    5.0

      MATRIX B
           1.0    1.0    1.0    1.0    1.0
           2.0    2.0    2.0    2.0    2.0
           3.0    3.0    3.0    3.0    3.0
           4.0    4.0    4.0    4.0    4.0

  **  OUTPUT  **

      IERR =    0

      OUTPUT MATRIX

      MATRIX C
          15.0   30.0   45.0   60.0
          15.0   30.0   45.0   60.0
          15.0   30.0   45.0   60.0
          15.0   30.0   45.0   60.0
          15.0   30.0   45.0   60.0
```

3.2.11 ZAM1MM, CAM1MM
複素行列 (2次元配列型) (実数引数型) の積 ($C = C \pm AB$)

(1) **機 能**

複素行列 (2次元配列型) の行列積 ($C = [C\pm]AB$) を求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZAM1MM (AR, AI, LMA, NM, NN, BR, BI, LNB, NL, CR, CI, LMC, ISW, IERR)

単精度サブルーチン:

CALL CAM1MM (AR, AI, LMA, NM, NN, BR, BI, LNB, NL, CR, CI, LMC, ISW, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: {32ビット整数版ではINTEGER(4)
64ビット整数版ではINTEGER(8)}

項番	引数名	型	大きさ	入出力	内　　容
1	AR	D R	LMA, NN	入 力	複素行列 A の実部 (2次元配列型)
2	AI	D R	LMA, NN	入 力	複素行列 A の虚部 (2次元配列型)
3	LMA	I	1	入 力	配列 AR と AI の整合寸法
4	NM	I	1	入 力	行列 A の行数 (行列 C の行数)
5	NN	I	1	入 力	行列 A の列数 (行列 B の行数)
6	BR	D R	LNB, NL	入 力	複素行列 B の実部 (2次元配列型)
7	BI	D R	LNB, NL	入 力	複素行列 B の虚部 (2次元配列型)
8	LNB	I	1	入 力	配列 BR と BI の整合寸法
9	NL	I	1	入 力	行列 B の列数 (行列 C の列数)
10	CR	D R	LMC, NL	入 力	初期複素行列 C の実部 (ISW $= \pm 1$ の時)(2次元配列型)
				出 力	行列積 ($C = [C\pm]AB$) の実部
11	CI	D R	LMC, NL	入 力	初期複素行列 C の虚部 (ISW $= \pm 1$ の時)(2次元配列型)
				出 力	行列積 ($C = [C\pm]AB$) の虚部
12	LMC	I	1	入 力	配列 CR と CI の整合寸法
13	ISW	I	1	入 力	処理スイッチ ISW $= 1$ の時: $C = C + AB$ ISW $= 0$ の時: $C = AB$ ISW $= -1$ の時: $C = C - AB$
14	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{NM} \leq \mathrm{LMA}, \mathrm{LMC}$

(b) $0 < \mathrm{NN} \leq \mathrm{LNB}$

(c) $\mathrm{NL} > 0$

(d) $\mathrm{ISW} \in \{0, 1, -1\}$

(5) **エラーインディケータ**

IERR の値	意　　　　味	処　理　内　容
0	正常終了.	
1000	NN = 1 であった.	処理を続ける.
3000	制限条件 (a), (b) または (c) を満足しなかった.	処理を打ち切る.
3010	制限条件 (d) を満足しなかった.	

(6) **注意事項**
なし.

(7) **使用例**

(a) 問 題
行列 A, B が以下の時, $C = AB$ を求める.

$$A = \begin{bmatrix} 1+i & 1+2i & 1+3i & 1+4i & 1+5i \\ 2+i & 2+2i & 2+3i & 2+4i & 2+5i \\ 3+i & 3+2i & 3+3i & 3+4i & 3+5i \\ 4+i & 4+2i & 4+3i & 4+4i & 4+5i \end{bmatrix}$$

$$B = \begin{bmatrix} 1+i & 1+2i & 1+3i & 1+4i \\ 2+i & 2+2i & 2+3i & 2+4i \\ 3+i & 3+2i & 3+3i & 3+4i \\ 4+i & 4+2i & 4+3i & 4+4i \\ 5+i & 5+2i & 5+3i & 5+4i \end{bmatrix}$$

(b) 入力データ
行列 $A, B, \mathrm{LMA} = 11, \mathrm{LNB} = 11, \mathrm{LNC} = 11, \mathrm{NM} = 4, \mathrm{NN} = 5, \mathrm{NL} = 4, \mathrm{ISW} = 0.$

(c) 主プログラム

```
      PROGRAM AAM1MM
! *** EXAMPLE OF ZAM1MM ***
      IMPLICIT NONE
      INTEGER LMA,LNB,LMC,ISW,NM,NN,NL
      PARAMETER( LMA=11, LNB=11, LMC=11, ISW=0 )
      PARAMETER( NM=4, NN=5, NL=4 )
      INTEGER IERR,I,J
      REAL(8) AR(LMA,NN),BR(LNB,NL),CR(LMC,NL)
      REAL(8) AI(LMA,NN),BI(LNB,NL),CI(LMC,NL)
!
      DO 100 I=1,NM
         READ(5,*) (AR(I,J),AI(I,J),J=1,NN)
  100 CONTINUE
      DO 110 I=1,NN
         READ(5,*) (BR(I,J),BI(I,J),J=1,NL)
  110 CONTINUE
!
      WRITE(6,6000) LMA,LNB,LMC,NM,NN,NL,ISW
      WRITE(6,6010) 'MATRIX A'
      DO 120 I=1,NM
         WRITE(6,6020) (AR(I,J),AI(I,J),J=1,NN)
  120 CONTINUE
      WRITE(6,6010) 'MATRIX B'
      DO 130 I=1,NN
```

```
         WRITE(6,6030) (BR(I,J),BI(I,J),J=1,NL)
  130 CONTINUE
!
      CALL ZAM1MM(AR,AI,LMA,NM,NN,BR,BI,LNB,NL,CR,CI,LMC,ISW,IERR)
!
      WRITE(6,6040) IERR
      IF( IERR .GE. 3000 ) STOP
!
      WRITE(6,6050) ’MATRIX C’
      DO 140 I=1,NM
         WRITE(6,6030) (CR(I,J),CI(I,J),J=1,NL)
  140 CONTINUE
      STOP
!
 6000 FORMAT(/,&
             1X,’***  ZAM1MM  ***’,/,/,&
             1X,’ **  INPUT  **’,/,/,&
             1X,’     LMA=’,I2,’   LNB=’,I2,’   LMC=’,I2,/,/,&
             1X,’     NM =’,I2,’   NN =’,I2,’   NL =’,I2,/,/,&
             1X,’     ISW=’,I2,/,/,&
             1X,’     INPUT MATRIX’)
 6010 FORMAT(/,&
             1X,5X,A)
 6030 FORMAT(1X,6X,4(’(’,F5.1,’,’,F5.1,’)’))
 6020 FORMAT(1X,6X,5(’(’,F5.1,’,’,F5.1,’)’))
 6040 FORMAT(/,&
             1X,’ **  OUTPUT  **’,/,/,&
             1X,’     IERR = ’,I4,/)
 6050 FORMAT(1X,’     OUTPUT MATRIX’,/,/,&
             1X,5X,A)
      END
```

(d) 出力結果

```
***  ZAM1MM  ***

 **  INPUT  **

     LMA=11   LNB=11   LMC=11

     NM = 4   NN = 5   NL = 4

     ISW= 0

     INPUT MATRIX

     MATRIX A
     (  1.0,  1.0)(  1.0,  2.0)(  1.0,  3.0)(  1.0,  4.0)(  1.0,  5.0)
     (  2.0,  1.0)(  2.0,  2.0)(  2.0,  3.0)(  2.0,  4.0)(  2.0,  5.0)
     (  3.0,  1.0)(  3.0,  2.0)(  3.0,  3.0)(  3.0,  4.0)(  3.0,  5.0)
     (  4.0,  1.0)(  4.0,  2.0)(  4.0,  3.0)(  4.0,  4.0)(  4.0,  5.0)

     MATRIX B
     (  1.0,  1.0)(  1.0,  2.0)(  1.0,  3.0)(  1.0,  4.0)
     (  2.0,  1.0)(  2.0,  2.0)(  2.0,  3.0)(  2.0,  4.0)
     (  3.0,  1.0)(  3.0,  2.0)(  3.0,  3.0)(  3.0,  4.0)
     (  4.0,  1.0)(  4.0,  2.0)(  4.0,  3.0)(  4.0,  4.0)
     (  5.0,  1.0)(  5.0,  2.0)(  5.0,  3.0)(  5.0,  4.0)

 **  OUTPUT  **

     IERR =    0

     OUTPUT MATRIX

     MATRIX C
     (  0.0, 60.0)(-15.0, 65.0)(-30.0, 70.0)(-45.0, 75.0)
     ( 15.0, 65.0)(  0.0, 75.0)(-15.0, 85.0)(-30.0, 95.0)
     ( 30.0, 70.0)( 15.0, 85.0)(  0.0,100.0)(-15.0,115.0)
     ( 45.0, 75.0)( 30.0, 95.0)( 15.0,115.0)(  0.0,135.0)
```

3.2.12 ZAM1MH, CAM1MH
複素行列 (2 次元配列型) (実数引数型) の積 ($C = C \pm AB^*$)

(1) **機 能**

複素行列 (2 次元配列型) の行列積 ($C = [C\pm]AB^*$) を求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZAM1MH (AR, AI, LMA, NM, NN, BR, BI, LLB, NL, CR, CI, LMC, ISW, IERR)

単精度サブルーチン:

CALL CAM1MH (AR, AI, LMA, NM, NN, BR, BI, LLB, NL, CR, CI, LMC, ISW, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$

項番	引数名	型	大きさ	入出力	内　容
1	AR	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	LMA, NN	入 力	複素行列 A の実部 (2 次元配列型)
2	AI	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	LMA, NN	入 力	複素行列 A の虚部 (2 次元配列型)
3	LMA	I	1	入 力	配列 AR と AI の整合寸法
4	NM	I	1	入 力	行列 A の行数 (行列 C の行数)
5	NN	I	1	入 力	行列 A の列数 (行列 B の列数)
6	BR	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	LLB, NN	入 力	複素行列 B の実部 (2 次元配列型)
7	BI	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	LLB, NN	入 力	複素行列 B の虚部 (2 次元配列型)
8	LLB	I	1	入 力	配列 BR と BI の整合寸法
9	NL	I	1	入 力	行列 B の行数 (行列 C の列数)
10	CR	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	LMC, NL	入 力	初期複素行列 C の実部 (ISW $= \pm1$ の時)(2 次元配列型)
				出 力	行列積 ($C = [C\pm]AB^*$) の実部
11	CI	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	LMC, NL	入 力	初期複素行列 C の虚部 (ISW $= \pm1$ の時)(2 次元配列型)
				出 力	行列積 ($C = [C\pm]AB^*$) の虚部
12	LMC	I	1	入 力	配列 CR と CI の整合寸法
13	ISW	I	1	入 力	処理スイッチ ISW $= 1$ の時: $C = C + AB^*$ ISW $= 0$ の時: $C = AB^*$ ISW $= -1$ の時: $C = C - AB^*$
14	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{NM} \leq \mathrm{LMA}, \mathrm{LMC}$

(b) $0 < \mathrm{NL} \leq \mathrm{LLB}$

(c) $\mathrm{NN} > 0$

(d) $\mathrm{ISW} \in \{0, 1, -1\}$

(5) **エラーインディケータ**

IERR の値	意　　　　　味	処　理　内　容
0	正常終了.	
1000	NN = 1 であった.	処理を続ける.
3000	制限条件 (a), (b) または (c) を満足しなかった.	処理を打ち切る.
3010	制限条件 (d) を満足しなかった.	

(6) **注意事項**
なし.

(7) **使用例**

(a) 問 題
行列 A, B が以下の時, $C = AB^*$ を求める.

$$A = \begin{bmatrix} 1+i & 1+2i & 1+3i & 1+4i & 1+5i \\ 2+i & 2+2i & 2+3i & 2+4i & 2+5i \\ 3+i & 3+2i & 3+3i & 3+4i & 3+5i \\ 4+i & 4+2i & 4+3i & 4+4i & 4+5i \end{bmatrix}$$

$$B = \begin{bmatrix} 1+i & 1+2i & 1+3i & 1+4i & 1+5i \\ 2+i & 2+2i & 2+3i & 2+4i & 2+5i \\ 3+i & 3+2i & 3+3i & 3+4i & 3+5i \\ 4+i & 4+2i & 4+3i & 4+4i & 4+5i \end{bmatrix}$$

(b) 入力データ
行列 A, 行列 B, $\mathrm{LMA} = 11, \mathrm{LLB} = 11, \mathrm{LNC} = 11, \mathrm{NM} = 4, \mathrm{NN} = 5, \mathrm{NL} = 4, \mathrm{ISW} = 0.$

(c) 主プログラム

```
      PROGRAM AAM1MH
! *** EXAMPLE OF ZAM1MH ***
      IMPLICIT NONE
      INTEGER LMA,LLB,LMC,ISW,NM,NN,NL
      PARAMETER( LMA=11, LLB=11, LMC=11, ISW=0 )
      PARAMETER( NM=4, NN=5, NL=4 )
      INTEGER IERR,I,J
      REAL(8) AR(LMA,NN),BR(LLB,NN),CR(LMC,NL)
      REAL(8) AI(LMA,NN),BI(LLB,NN),CI(LMC,NL)
!
      DO 100 I=1,NM
         READ(5,*) (AR(I,J),AI(I,J),J=1,NN)
  100 CONTINUE
      DO 110 I=1,NL
         READ(5,*) (BR(I,J),BI(I,J),J=1,NN)
  110 CONTINUE
!
      WRITE(6,6000) LMA,LLB,LMC,NM,NN,NL,ISW
      WRITE(6,6010) 'MATRIX A'
      DO 120 I=1,NM
         WRITE(6,6020) (AR(I,J),AI(I,J),J=1,NN)
  120 CONTINUE
      WRITE(6,6010) 'MATRIX B'
      DO 130 I=1,NL
         WRITE(6,6020) (BR(I,J),BI(I,J),J=1,NN)
  130 CONTINUE
```

```
!
      CALL ZAM1MH(AR,AI,LMA,NM,NN,BR,BI,LLB,NL,CR,CI,LMC,ISW,IERR)
!
      WRITE(6,6030) IERR
      IF( IERR .GE. 3000 ) STOP
!
      WRITE(6,6040) 'MATRIX C'
      DO 140 I=1,NM
         WRITE(6,6050) (CR(I,J),CI(I,J),J=1,NL)
  140 CONTINUE
      STOP
!
 6000 FORMAT(/,&
             1X,'***  ZAM1MH  ***',/,/,&
             1X,' **  INPUT  **',/,/,&
             1X,'     LMA=',I2,'   LLB=',I2,'   LMC=',I2,/,/,&
             1X,'     NM =',I2,'   NN =',I2,'   NL =',I2,/,/,&
             1X,'     ISW=',I2,/,/,&
             1X,'     INPUT MATRIX')
 6010 FORMAT(/,&
             1X,5X,A)
 6050 FORMAT(1X,6X,4('(',F5.1,',',F5.1,')'))
 6020 FORMAT(1X,6X,5('(',F5.1,',',F5.1,')'))
 6030 FORMAT(/,&
             1X,' **  OUTPUT  **',/,/,&
             1X,'     IERR = ',I4,/)
 6040 FORMAT(1X,'     OUTPUT MATRIX',/,/,&
             1X,5X,A)
      END
```

(d) 出力結果

```
***  ZAM1MH  ***

 **  INPUT  **

     LMA=11   LLB=11   LMC=11

     NM = 4   NN = 5   NL = 4

     ISW= 0

     INPUT MATRIX

     MATRIX A
      (  1.0,  1.0)(  1.0,  2.0)(  1.0,  3.0)(  1.0,  4.0)(  1.0,  5.0)
      (  2.0,  1.0)(  2.0,  2.0)(  2.0,  3.0)(  2.0,  4.0)(  2.0,  5.0)
      (  3.0,  1.0)(  3.0,  2.0)(  3.0,  3.0)(  3.0,  4.0)(  3.0,  5.0)
      (  4.0,  1.0)(  4.0,  2.0)(  4.0,  3.0)(  4.0,  4.0)(  4.0,  5.0)

     MATRIX B
      (  1.0,  1.0)(  1.0,  2.0)(  1.0,  3.0)(  1.0,  4.0)(  1.0,  5.0)
      (  2.0,  1.0)(  2.0,  2.0)(  2.0,  3.0)(  2.0,  4.0)(  2.0,  5.0)
      (  3.0,  1.0)(  3.0,  2.0)(  3.0,  3.0)(  3.0,  4.0)(  3.0,  5.0)
      (  4.0,  1.0)(  4.0,  2.0)(  4.0,  3.0)(  4.0,  4.0)(  4.0,  5.0)

 **  OUTPUT  **

     IERR =    0

     OUTPUT MATRIX

     MATRIX C
      ( 60.0,  0.0)( 65.0, 15.0)( 70.0, 30.0)( 75.0, 45.0)
      ( 65.0,-15.0)( 75.0,  0.0)( 85.0, 15.0)( 95.0, 30.0)
      ( 70.0,-30.0)( 85.0,-15.0)(100.0,  0.0)(115.0, 15.0)
      ( 75.0,-45.0)( 95.0,-30.0)(115.0,-15.0)(135.0,  0.0)
```

3.2.13 ZAM1HM, CAM1HM
複素行列 (2 次元配列型) (実数引数型) の積 ($C = C \pm A^*B$)

(1) **機 能**

複素行列の行列積 ($C = [C\pm]A^*B$) を求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZAM1HM (AR, AI, LNA, NM, NN, BR, BI, LNB, NL, CR, CI, LMC, ISW, IERR)

単精度サブルーチン:

CALL CAM1HM (AR, AI, LNA, NM, NN, BR, BI, LNB, NL, CR, CI, LMC, ISW, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$

項番	引数名	型	大きさ	入出力	内　容
1	AR	$\left\{\begin{matrix}D\\R\end{matrix}\right\}$	LNA, NM	入 力	複素行列 A の実部 (2 次元配列型)
2	AI	$\left\{\begin{matrix}D\\R\end{matrix}\right\}$	LNA, NM	入 力	複素行列 A の虚部 (2 次元配列型)
3	LNA	I	1	入 力	配列 AR と AI の整合寸法
4	NM	I	1	入 力	行列 A の列数 (行列 C の行数)
5	NN	I	1	入 力	行列 A の行数 (行列 B の行数)
6	BR	$\left\{\begin{matrix}D\\R\end{matrix}\right\}$	LNB, NL	入 力	複素行列 B の実部 (2 次元配列型)
7	BI	$\left\{\begin{matrix}D\\R\end{matrix}\right\}$	LNB, NL	入 力	複素行列 B の虚部 (2 次元配列型)
8	LNB	I	1	入 力	配列 BR と BI の整合寸法
9	NL	I	1	入 力	行列 B の列数 (行列 C の列数)
10	CR	$\left\{\begin{matrix}D\\R\end{matrix}\right\}$	LMC, NL	入 力	初期複素行列 C の実部 (ISW $= \pm 1$ の時)(2 次元配列型)
				出 力	行列積 ($C = [C\pm]A^*B$) の実部
11	CI	$\left\{\begin{matrix}D\\R\end{matrix}\right\}$	LMC, NL	入 力	初期複素行列 C の虚部 (ISW $= \pm 1$ の時)(2 次元配列型)
				出 力	行列積 ($C = [C\pm]A^*B$) の虚部
12	LMC	I	1	入 力	配列 CR と CI の整合寸法
13	ISW	I	1	入 力	処理スイッチ ISW $= 1$ の時: $C = C + A^*B$ ISW $= 0$ の時: $C = A^*B$ ISW $= -1$ の時: $C = C - A^*B$
14	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{NM} \le \mathrm{LMC}$

(b) $0 < \mathrm{NN} \le \mathrm{LNA}, \mathrm{LNB}$

(c) $\mathrm{NL} > 0$

(d) $\mathrm{ISW} \in \{0, 1, -1\}$

(5) **エラーインディケータ**

IERR の値	意　　　　味	処　理　内　容
0	正常終了.	
1000	NN = 1 であった.	処理を続ける.
3000	制限条件 (a), (b) または (c) を満足しなかった.	処理を打ち切る.
3010	制限条件 (d) を満足しなかった.	

(6) **注意事項**

なし.

(7) **使用例**

(a) 問 題

$$A = \begin{bmatrix} 1+i & 1+2i & 1+3i & 1+4i & 1+5i \\ 2+i & 2+2i & 2+3i & 2+4i & 2+5i \\ 3+i & 3+2i & 3+3i & 3+4i & 3+5i \\ 4+i & 4+2i & 4+3i & 4+4i & 4+5i \end{bmatrix}$$

$$B = \begin{bmatrix} 1+i & 1+2i & 1+3i & 1+4i \\ 2+i & 2+2i & 2+3i & 2+4i \\ 3+i & 3+2i & 3+3i & 3+4i \\ 4+i & 4+2i & 4+3i & 4+4i \end{bmatrix}$$

$C = A^{*}B$ を求める.

(b) 入力データ

行列 A, 行列 B, LNA = 11, LNB = 11, LNC = 11, NM = 5, NN = 4, NL = 4, ISW = 0.

(c) 主プログラム

```
      PROGRAM AAM1HM
! *** EXAMPLE OF ZAM1HM ***
      IMPLICIT NONE
      INTEGER LNA,LNB,LMC,ISW,NM,NN,NL
      PARAMETER( LNA=11, LNB=11, LMC=11, ISW=0 )
      PARAMETER( NM=5, NN=4, NL=4 )
      INTEGER IERR,I,J
      REAL(8) AR(LNA,NM),BR(LNB,NL),CR(LMC,NL)
      REAL(8) AI(LNA,NM),BI(LNB,NL),CI(LMC,NL)
!
      DO 100 I=1,NN
         READ(5,*) (AR(I,J),AI(I,J),J=1,NM)
  100 CONTINUE
      DO 110 I=1,NN
         READ(5,*) (BR(I,J),BI(I,J),J=1,NL)
  110 CONTINUE
!
      WRITE(6,6000) LNA,LNB,LMC,NM,NN,NL,ISW
      WRITE(6,6010) 'MATRIX A'
      DO 120 I=1,NN
         WRITE(6,6020) (AR(I,J),AI(I,J),J=1,NM)
  120 CONTINUE
      WRITE(6,6010) 'MATRIX B'
      DO 130 I=1,NN
         WRITE(6,6030) (BR(I,J),BI(I,J),J=1,NL)
  130 CONTINUE
```

```
!
      CALL ZAM1HM(AR,AI,LNA,NM,NN,BR,BI,LNB,NL,CR,CI,LMC,ISW,IERR)
!
      WRITE(6,6040) IERR
      IF( IERR .GE. 3000 ) STOP
!
      WRITE(6,6050) 'MATRIX C'
      DO 140 I=1,NM
         WRITE(6,6030) (CR(I,J),CI(I,J),J=1,NL)
  140 CONTINUE
      STOP
!
 6000 FORMAT(/,&
             1X,'***  ZAM1HM  ***',/,/,&
             1X,' **  INPUT  **',/,/,&
             1X,'     LNA=',I2,'   LNB=',I2,'   LMC=',I2,/,/,&
             1X,'     NM =',I2,'   NN =',I2,'   NL =',I2,/,/,&
             1X,'     ISW=',I2,/,/,&
             1X,'     INPUT MATRIX')
 6010 FORMAT(/,&
             1X,5X,A)
 6030 FORMAT(1X,4X,4('(',F5.1,',',F5.1,')'))
 6020 FORMAT(1X,4X,5('(',F5.1,',',F5.1,')'))
 6040 FORMAT(/,&
             1X,' **  OUTPUT  **',/,/,&
             1X,'     IERR = ',I4,/)
 6050 FORMAT(1X,'     OUTPUT MATRIX',/,/,&
             1X,5X,A)
      END
```

(d) 出力結果

```
***  ZAM1HM  ***

 **  INPUT  **

     LNA=11   LNB=11   LMC=11

     NM = 5   NN = 4   NL = 4

     ISW= 0

     INPUT MATRIX

     MATRIX A
    (  1.0,  1.0)(  1.0,  2.0)(  1.0,  3.0)(  1.0,  4.0)(  1.0,  5.0)
    (  2.0,  1.0)(  2.0,  2.0)(  2.0,  3.0)(  2.0,  4.0)(  2.0,  5.0)
    (  3.0,  1.0)(  3.0,  2.0)(  3.0,  3.0)(  3.0,  4.0)(  3.0,  5.0)
    (  4.0,  1.0)(  4.0,  2.0)(  4.0,  3.0)(  4.0,  4.0)(  4.0,  5.0)

     MATRIX B
    (  1.0,  1.0)(  1.0,  2.0)(  1.0,  3.0)(  1.0,  4.0)
    (  2.0,  1.0)(  2.0,  2.0)(  2.0,  3.0)(  2.0,  4.0)
    (  3.0,  1.0)(  3.0,  2.0)(  3.0,  3.0)(  3.0,  4.0)
    (  4.0,  1.0)(  4.0,  2.0)(  4.0,  3.0)(  4.0,  4.0)

 **  OUTPUT  **

     IERR =    0

     OUTPUT MATRIX

     MATRIX C
    ( 34.0,  0.0)( 38.0, 10.0)( 42.0, 20.0)( 46.0, 30.0)
    ( 38.0,-10.0)( 46.0,  0.0)( 54.0, 10.0)( 62.0, 20.0)
    ( 42.0,-20.0)( 54.0,-10.0)( 66.0,  0.0)( 78.0, 10.0)
    ( 46.0,-30.0)( 62.0,-20.0)( 78.0,-10.0)( 94.0,  0.0)
    ( 50.0,-40.0)( 70.0,-30.0)( 90.0,-20.0)(110.0,-10.0)
```

3.2.14 ZAM1HH, CAM1HH
複素行列 (2 次元配列型) (実数引数型) の積 ($C = C \pm A^*B^*$)

(1) **機 能**

複素行列の行列積 ($C = [C\pm]A^*B^*$) を求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZAM1HH (AR, AI, LNA, NM, NN, BR, BI, LLB, NL, CR, CI, LMC, ISW, IERR)

単精度サブルーチン:

CALL CAM1HH (AR, AI, LNA, NM, NN, BR, BI, LLB, NL, CR, CI, LMC, ISW, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$

項番	引数名	型	大きさ	入出力	内　　容
1	AR	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	LNA, NM	入 力	複素行列 A の実部 (2 次元配列型)
2	AI	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	LNA, NM	入 力	複素行列 A の虚部 (2 次元配列型)
3	LNA	I	1	入 力	配列 AR と AI の整合寸法
4	NM	I	1	入 力	行列 A の列数 (行列 C の行数)
5	NN	I	1	入 力	行列 A の行数 (行列 B の行数)
6	BR	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	LLB, NN	入 力	複素行列 B の実部 (2 次元配列型)
7	BI	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	LLB, NN	入 力	複素行列 B の虚部 (2 次元配列型)
8	LLB	I	1	入 力	配列 BR と BI の整合寸法
9	NL	I	1	入 力	行列 B の行数 (行列 C の列数)
10	CR	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	LMC, NL	入 力	初期複素行列 C の実部 (ISW $= \pm1$ の時)(2 次元配列型)
				出 力	行列積 ($C = [C\pm]A^*B^*$) の実部
11	CI	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	LMC, NL	入 力	初期複素行列 C の虚部 (ISW $= \pm1$ の時)(2 次元配列型)
				出 力	行列積 ($C = [C\pm]A^*B^*$) の虚部
12	LMC	I	1	入 力	配列 CR と CI の整合寸法
13	ISW	I	1	入 力	処理スイッチ ISW $= 1$ の時: $C = C + A^*B^*$ ISW $= 0$ の時: $C = A^*B^*$ ISW $= -1$ の時: $C = C - A^*B^*$
14	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{NM} \leq \mathrm{LMC}$

(b) $0 < \mathrm{NN} \leq \mathrm{LNA}$

(c) $0 < \mathrm{NL} \leq \mathrm{LNB}$

(d) $\mathrm{ISW} \in \{0, 1, -1\}$

(5) **エラーインディケータ**

IERR の値	意　　　味	処　理　内　容
0	正常終了.	
1000	NN = 1 であった.	処理を続ける.
3000	制限条件 (a), (b) または (c) を満足しなかった.	処理を打ち切る.
3010	制限条件 (d) を満足しなかった.	

(6) **注意事項**

なし.

(7) **使用例**

(a) 問 題

$$A = \begin{bmatrix} 1+i & 1+2i & 1+3i & 1+4i & 1+5i \\ 2+i & 2+2i & 2+3i & 2+4i & 2+5i \\ 3+i & 3+2i & 3+3i & 3+4i & 3+5i \\ 4+i & 4+2i & 4+3i & 4+4i & 4+5i \\ 5+i & 5+2i & 5+3i & 5+4i & 5+5i \end{bmatrix}$$

$$B = \begin{bmatrix} 1+i & 1+2i & 1+3i & 1+4i & 1+5i \\ 2+i & 2+2i & 2+3i & 2+4i & 2+5i \\ 3+i & 3+2i & 3+3i & 3+4i & 3+5i \\ 4+i & 4+2i & 4+3i & 4+4i & 4+5i \end{bmatrix}$$

$C = A^*B^*$ を求める.

(b) 入力データ

行列 A, 行列 B, $\mathrm{LNA} = 11$, $\mathrm{LLB} = 11$, $\mathrm{LNC} = 11$, $\mathrm{NM} = 5$, $\mathrm{NN} = 5$, $\mathrm{NL} = 4$, $\mathrm{ISW} = 0$.

(c) 主プログラム

```
      PROGRAM AAM1HH
! *** EXAMPLE OF ZAM1HH ***
      IMPLICIT NONE
      INTEGER LNA,LLB,LMC,ISW,NM,NN,NL
      PARAMETER( LNA=11, LLB=11, LMC=11, ISW=0 )
      PARAMETER( NM=5, NN=5, NL=4 )
      INTEGER IERR,I,J
      REAL(8) AR(LNA,NM),BR(LLB,NN),CR(LMC,NL)
      REAL(8) AI(LNA,NM),BI(LLB,NN),CI(LMC,NL)
!
      DO 100 I=1,NN
         READ(5,*) (AR(I,J),AI(I,J),J=1,NM)
  100 CONTINUE
      DO 110 I=1,NL
         READ(5,*) (BR(I,J),BI(I,J),J=1,NN)
  110 CONTINUE
!
      WRITE(6,6000) LNA,LLB,LMC,NM,NN,NL,ISW
      WRITE(6,6010) 'MATRIX A'
      DO 120 I=1,NN
         WRITE(6,6020) (AR(I,J),AI(I,J),J=1,NM)
  120 CONTINUE
      WRITE(6,6010) 'MATRIX B'
```

```
      DO 130 I=1,NL
         WRITE(6,6020) (BR(I,J),BI(I,J),J=1,NN)
  130 CONTINUE
!
      CALL ZAM1HH(AR,AI,LNA,NM,NN,BR,BI,LLB,NL,CR,CI,LMC,ISW,IERR)
!
      WRITE(6,6030) IERR
      IF( IERR .GE. 3000 ) STOP
!
      WRITE(6,6040) 'MATRIX C'
      DO 140 I=1,NM
         WRITE(6,6050) (CR(I,J),CI(I,J),J=1,NL)
  140 CONTINUE
      STOP
!
 6000 FORMAT(/,&
             1X,'***  ZAM1HH  ***',/,/,&
             1X,' **  INPUT  **',/,/,&
             1X,'     LNA=',I2,'   LLB=',I2,'   LMC=',I2,/,/,&
             1X,'     NM =',I2,'   NN =',I2,'   NL =',I2,/,/,&
             1X,'     ISW=',I2,/,/,&
             1X,'     INPUT MATRIX')
 6010 FORMAT(/,&
             1X,5X,A)
 6020 FORMAT(1X,4X,5('(',F5.1,',',F5.1,')'))
 6050 FORMAT(1X,4X,4('(',F6.1,',',F6.1,')'))
 6030 FORMAT(/,&
             1X,' **  OUTPUT  **',/,/,&
             1X,'     IERR = ',I4,/)
 6040 FORMAT(1X,'     OUTPUT MATRIX',/,/,&
             1X,5X,A)
      END
```

(d) 出力結果

```
***  ZAM1HH  ***

 **  INPUT  **

     LNA=11   LLB=11   LMC=11

     NM = 5   NN = 5   NL = 4

     ISW= 0

     INPUT MATRIX

     MATRIX A
    (  1.0,  1.0)(  1.0,  2.0)(  1.0,  3.0)(  1.0,  4.0)(  1.0,  5.0)
    (  2.0,  1.0)(  2.0,  2.0)(  2.0,  3.0)(  2.0,  4.0)(  2.0,  5.0)
    (  3.0,  1.0)(  3.0,  2.0)(  3.0,  3.0)(  3.0,  4.0)(  3.0,  5.0)
    (  4.0,  1.0)(  4.0,  2.0)(  4.0,  3.0)(  4.0,  4.0)(  4.0,  5.0)
    (  5.0,  1.0)(  5.0,  2.0)(  5.0,  3.0)(  5.0,  4.0)(  5.0,  5.0)

     MATRIX B
    (  1.0,  1.0)(  1.0,  2.0)(  1.0,  3.0)(  1.0,  4.0)(  1.0,  5.0)
    (  2.0,  1.0)(  2.0,  2.0)(  2.0,  3.0)(  2.0,  4.0)(  2.0,  5.0)
    (  3.0,  1.0)(  3.0,  2.0)(  3.0,  3.0)(  3.0,  4.0)(  3.0,  5.0)
    (  4.0,  1.0)(  4.0,  2.0)(  4.0,  3.0)(  4.0,  4.0)(  4.0,  5.0)

 **  OUTPUT  **

     IERR =    0

     OUTPUT MATRIX

     MATRIX C
    (   0.0, -60.0)(  15.0, -65.0)(  30.0, -70.0)(  45.0, -75.0)
    ( -15.0, -65.0)(   0.0, -75.0)(  15.0, -85.0)(  30.0, -95.0)
    ( -30.0, -70.0)( -15.0, -85.0)(   0.0,-100.0)(  15.0,-115.0)
    ( -45.0, -75.0)( -30.0, -95.0)( -15.0,-115.0)(   0.0,-135.0)
    ( -60.0, -80.0)( -45.0,-105.0)( -30.0,-130.0)( -15.0,-155.0)
```

3.2.15 ZAN1MM, CAN1MM
複素行列 (2 次元配列型) (複素引数型) の積 ($C = C \pm AB$)

(1) **機 能**

複素行列 (2 次元配列型) の行列積 ($C = [C\pm]AB$) を求める．

(2) **使用法**

倍精度サブルーチン:

CALL ZAN1MM (A, LMA, NM, NN, B, LNB, NL, C, LMC, ISW, IERR)

単精度サブルーチン:

CALL CAN1MM (A, LMA, NM, NN, B, LNB, NL, C, LMC, ISW, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: $\left\{ \begin{matrix} \text{32 ビット整数版では INTEGER(4)} \\ \text{64 ビット整数版では INTEGER(8)} \end{matrix} \right\}$

項番	引数名	型	大きさ	入出力	内　　容
1	A	$\left\{ \begin{matrix} \text{Z} \\ \text{C} \end{matrix} \right\}$	LMA, NN	入 力	複素行列 A (2 次元配列型)
2	LMA	I	1	入 力	配列 A の整合寸法
3	NM	I	1	入 力	行列 A の行数 (行列 C の行数)
4	NN	I	1	入 力	行列 A の列数 (行列 B の行数)
5	B	$\left\{ \begin{matrix} \text{Z} \\ \text{C} \end{matrix} \right\}$	LNB, NL	入 力	複素行列 B (2 次元配列型)
6	LNB	I	1	入 力	配列 B の整合寸法
7	NL	I	1	入 力	行列 B の列数 (行列 C の列数)
8	C	$\left\{ \begin{matrix} \text{Z} \\ \text{C} \end{matrix} \right\}$	LMC, NL	入 力	初期複素行列 C (ISW = ±1 の時)(2 次元配列型)
				出 力	行列積 ($C = [C\pm]AB$)
9	LMC	I	1	入 力	配列 C の整合寸法
10	ISW	I	1	入 力	処理スイッチ ISW = 1 の時: $C = C + AB$ ISW = 0 の時: $C = AB$ ISW = −1 の時: $C = C - AB$
11	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{NM} \le \mathrm{LMA}, \mathrm{LMC}$

(b) $0 < \mathrm{NN} \le \mathrm{LNB}$

(c) $\mathrm{NL} > 0$

(d) $\mathrm{ISW} \in \{0, 1, -1\}$

(5) **エラーインディケータ**

IERR の値	意　　味	処　理　内　容
0	正常終了.	
1000	NN = 1 であった.	処理を続ける.
3000	制限条件 (a), (b) または (c) を満足しなかった.	処理を打ち切る.
3010	制限条件 (d) を満足しなかった.	

(6) **注意事項**

なし.

(7) **使用例**

(a) 問 題

$$A = \begin{bmatrix} 1+i & 1+2i & 1+3i & 1+4i & 1+5i \\ 2+i & 2+2i & 2+3i & 2+4i & 2+5i \\ 3+i & 3+2i & 3+3i & 3+4i & 3+5i \\ 4+i & 4+2i & 4+3i & 4+4i & 4+5i \end{bmatrix}$$

$$B = \begin{bmatrix} 1+i & 1+2i & 1+3i & 1+4i \\ 2+i & 2+2i & 2+3i & 2+4i \\ 3+i & 3+2i & 3+3i & 3+4i \\ 4+i & 4+2i & 4+3i & 4+4i \\ 5+i & 5+2i & 5+3i & 5+4i \end{bmatrix}$$

$C = AB$ を求める.

(b) 入力データ

行列 A, 行列 B, LMA = 11, LNB = 11, LNC = 11, NM = 4, NN = 5, NL = 4, ISW = 0.

(c) 主プログラム

```
      PROGRAM AAN1MM
! *** EXAMPLE OF ZAN1MM ***
      IMPLICIT NONE
      INTEGER LMA,LNB,LMC,ISW,NM,NN,NL
      PARAMETER( LMA=11, LNB=11, LMC=11, ISW=0 )
      PARAMETER( NM=4, NN=5, NL=4 )
      INTEGER IERR,I,J
      COMPLEX(8) A(LMA,NN),B(LNB,NL),C(LMC,NL)
!
      DO 100 I=1,NM
         READ(5,*) (A(I,J),J=1,NN)
  100 CONTINUE
      DO 110 I=1,NN
         READ(5,*) (B(I,J),J=1,NL)
  110 CONTINUE
!
      WRITE(6,6000) LMA,LNB,LMC,NM,NN,NL,ISW
      WRITE(6,6010) 'MATRIX A'
      DO 120 I=1,NM
         WRITE(6,6020) (A(I,J),J=1,NN)
  120 CONTINUE
      WRITE(6,6010) 'MATRIX B'
      DO 130 I=1,NN
         WRITE(6,6030) (B(I,J),J=1,NL)
  130 CONTINUE
!
      CALL ZAN1MM(A,LMA,NM,NN,B,LNB,NL,C,LMC,ISW,IERR)
!
      WRITE(6,6040) IERR
      IF( IERR .GE. 3000 ) STOP
!
      WRITE(6,6050) 'MATRIX C'
      DO 140 I=1,NM
         WRITE(6,6030) (C(I,J),J=1,NL)
  140 CONTINUE
      STOP
!
 6000 FORMAT(/,&
             1X,'***  ZAN1MM  ***',/,/,&
```

```
            1X,' **  INPUT  **',/,/,&
            1X,'     LMA=',I2,'   LNB=',I2,'   LMC=',I2,/,/,&
            1X,'     NM =',I2,'   NN =',I2,'   NL =',I2,/,/,&
            1X,'     ISW=',I2,/,/,&
            1X,'     INPUT MATRIX')
6010 FORMAT(/,&
            1X,5X,A)
6030 FORMAT(1X,6X,4('(',F5.1,',',F5.1,')'))
6020 FORMAT(1X,6X,5('(',F5.1,',',F5.1,')'))
6040 FORMAT(/,&
            1X,' **  OUTPUT  **',/,/,&
            1X,'     IERR = ',I4,/)
6050 FORMAT(1X,'     OUTPUT MATRIX',/,/,&
            1X,5X,A)
     END
```

(d) 出力結果

```
***  ZAN1MM  ***

 **  INPUT  **

     LMA=11   LNB=11   LMC=11

     NM = 4   NN = 5   NL = 4

     ISW= 0

     INPUT MATRIX

     MATRIX A
      (  1.0,  1.0)(  1.0,  2.0)(  1.0,  3.0)(  1.0,  4.0)(  1.0,  5.0)
      (  2.0,  1.0)(  2.0,  2.0)(  2.0,  3.0)(  2.0,  4.0)(  2.0,  5.0)
      (  3.0,  1.0)(  3.0,  2.0)(  3.0,  3.0)(  3.0,  4.0)(  3.0,  5.0)
      (  4.0,  1.0)(  4.0,  2.0)(  4.0,  3.0)(  4.0,  4.0)(  4.0,  5.0)

     MATRIX B
      (  1.0,  1.0)(  1.0,  2.0)(  1.0,  3.0)(  1.0,  4.0)
      (  2.0,  1.0)(  2.0,  2.0)(  2.0,  3.0)(  2.0,  4.0)
      (  3.0,  1.0)(  3.0,  2.0)(  3.0,  3.0)(  3.0,  4.0)
      (  4.0,  1.0)(  4.0,  2.0)(  4.0,  3.0)(  4.0,  4.0)
      (  5.0,  1.0)(  5.0,  2.0)(  5.0,  3.0)(  5.0,  4.0)

 **  OUTPUT  **

     IERR =    0

     OUTPUT MATRIX

     MATRIX C
      (  0.0, 60.0)(-15.0, 65.0)(-30.0, 70.0)(-45.0, 75.0)
      ( 15.0, 65.0)(  0.0, 75.0)(-15.0, 85.0)(-30.0, 95.0)
      ( 30.0, 70.0)( 15.0, 85.0)(  0.0,100.0)(-15.0,115.0)
      ( 45.0, 75.0)( 30.0, 95.0)( 15.0,115.0)(  0.0,135.0)
```

3.2.16 ZAN1MH, CAN1MH
複素行列 (2 次元配列型) (複素引数型) の積 ($C = C \pm AB^*$)

(1) **機 能**

複素行列 (2 次元配列型) の行列積 ($C = [C\pm]AB^*$) を求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZAN1MH (A, LMA, NM, NN, B, LLB, NL, C, LMC, ISW, IERR)

単精度サブルーチン:

CALL CAN1MH (A, LMA, NM, NN, B, LLB, NL, C, LMC, ISW, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$

項番	引数名	型	大きさ	入出力	内　容
1	A	$\left\{\begin{matrix}\text{Z}\\\text{C}\end{matrix}\right\}$	LMA, NN	入 力	複素行列 A (2 次元配列型)
2	LMA	I	1	入 力	配列 A の整合寸法
3	NM	I	1	入 力	行列 A の行数 (行列 C の行数)
4	NN	I	1	入 力	行列 A の列数 (行列 B の列数)
5	B	$\left\{\begin{matrix}\text{Z}\\\text{C}\end{matrix}\right\}$	LLB, NN	入 力	複素行列 B (2 次元配列型)
6	LLB	I	1	入 力	配列 B の整合寸法
7	NL	I	1	入 力	行列 B の行数 (行列 C の列数)
8	C	$\left\{\begin{matrix}\text{Z}\\\text{C}\end{matrix}\right\}$	LMC, NL	入 力	初期複素行列 C (ISW = ±1 の時)(2 次元配列型)
				出 力	行列積 ($C = [C\pm]AB$)
9	LMC	I	1	入 力	配列 C の整合寸法
10	ISW	I	1	入 力	処理スイッチ ISW = 1 の時: $C = C + AB^*$ ISW = 0 の時: $C = AB^*$ ISW = −1 の時: $C = C - AB^*$
11	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{NM} \leq \mathrm{LMA}, \mathrm{LMC}$

(b) $0 < \mathrm{NL} \leq \mathrm{LLB}$

(c) $\mathrm{NN} > 0$

(d) $\mathrm{ISW} \in \{0, 1, -1\}$

(5) エラーインディケータ

IERR の値	意　　味	処　理　内　容
0	正常終了.	
1000	NN = 1 であった.	処理を続ける.
3000	制限条件 (a), (b) または (c) を満足しなかった.	処理を打ち切る.
3010	制限条件 (d) を満足しなかった.	

(6) 注意事項

なし.

(7) 使用例

(a) 問 題

$$A = \begin{bmatrix} 1+i & 1+2i & 1+3i & 1+4i & 1+5i \\ 2+i & 2+2i & 2+3i & 2+4i & 2+5i \\ 3+i & 3+2i & 3+3i & 3+4i & 3+5i \\ 4+i & 4+2i & 4+3i & 4+4i & 4+5i \end{bmatrix}$$

$$B = \begin{bmatrix} 1+i & 1+2i & 1+3i & 1+4i & 1+5i \\ 2+i & 2+2i & 2+3i & 2+4i & 2+5i \\ 3+i & 3+2i & 3+3i & 3+4i & 3+5i \\ 4+i & 4+2i & 4+3i & 4+4i & 4+5i \end{bmatrix}$$

$C = AB^*$ を求める.

(b) 入力データ

行列 A, 行列 B, LMA = 11, LLB = 11, LNC = 11, NM = 4, NN = 5, NL = 4, ISW = 0.

(c) 主プログラム

```
      PROGRAM AAN1MH
! *** EXAMPLE OF ZAN1MH ***
      IMPLICIT NONE
      INTEGER LMA,LLB,LMC,ISW,NM,NN,NL
      PARAMETER( LMA=11, LLB=11, LMC=11, ISW=0 )
      PARAMETER( NM=4, NN=5, NL=4 )
      INTEGER IERR,I,J
      COMPLEX(8) A(LMA,NN),B(LLB,NN),C(LMC,NL)
!
      DO 100 I=1,NM
         READ(5,*) (A(I,J),J=1,NN)
  100 CONTINUE
      DO 110 I=1,NL
         READ(5,*) (B(I,J),J=1,NN)
  110 CONTINUE
!
      WRITE(6,6000) LMA,LLB,LMC,NM,NN,NL,ISW
      WRITE(6,6010) 'MATRIX A'
      DO 120 I=1,NM
         WRITE(6,6020) (A(I,J),J=1,NN)
  120 CONTINUE
      WRITE(6,6010) 'MATRIX B'
      DO 130 I=1,NL
         WRITE(6,6020) (B(I,J),J=1,NN)
  130 CONTINUE
!
      CALL ZAN1MH(A,LMA,NM,NN,B,LLB,NL,C,LMC,ISW,IERR)
!
      WRITE(6,6030) IERR
      IF( IERR .GE. 3000 ) STOP
!
      WRITE(6,6040) 'MATRIX C'
      DO 140 I=1,NM
         WRITE(6,6050) (C(I,J),J=1,NL)
  140 CONTINUE
      STOP
!
 6000 FORMAT(/,&
             1X,'***  ZAN1MH  ***',/,/,&
             1X,' **  INPUT  **',/,/,&
             1X,'     LMA=',I2,'   LLB=',I2,'   LMC=',I2,/,/,&
```

```
            1X,'     NM =',I2,'   NN =',I2,'   NL =',I2,/,/,&
            1X,'     ISW=',I2,/,/,&
            1X,'     INPUT MATRIX')
 6010 FORMAT(/,&
            1X,5X,A)
 6050 FORMAT(1X,6X,4('(',F5.1,',',F5.1,')'))
 6020 FORMAT(1X,6X,5('(',F5.1,',',F5.1,')'))
 6030 FORMAT(/,&
            1X,' **  OUTPUT  **',/,/,&
            1X,'     IERR = ',I4,/)
 6040 FORMAT(1X,'     OUTPUT MATRIX',/,/,&
            1X,5X,A)
      END
```

(d) 出力結果

```
***  ZAN1MH  ***

 **  INPUT  **

     LMA=11   LLB=11   LMC=11

     NM = 4   NN = 5   NL = 4

     ISW= 0

     INPUT MATRIX

     MATRIX A
      (  1.0,  1.0)(  1.0,  2.0)(  1.0,  3.0)(  1.0,  4.0)(  1.0,  5.0)
      (  2.0,  1.0)(  2.0,  2.0)(  2.0,  3.0)(  2.0,  4.0)(  2.0,  5.0)
      (  3.0,  1.0)(  3.0,  2.0)(  3.0,  3.0)(  3.0,  4.0)(  3.0,  5.0)
      (  4.0,  1.0)(  4.0,  2.0)(  4.0,  3.0)(  4.0,  4.0)(  4.0,  5.0)

     MATRIX B
      (  1.0,  1.0)(  1.0,  2.0)(  1.0,  3.0)(  1.0,  4.0)(  1.0,  5.0)
      (  2.0,  1.0)(  2.0,  2.0)(  2.0,  3.0)(  2.0,  4.0)(  2.0,  5.0)
      (  3.0,  1.0)(  3.0,  2.0)(  3.0,  3.0)(  3.0,  4.0)(  3.0,  5.0)
      (  4.0,  1.0)(  4.0,  2.0)(  4.0,  3.0)(  4.0,  4.0)(  4.0,  5.0)

 **  OUTPUT  **

     IERR =    0

     OUTPUT MATRIX

     MATRIX C
      ( 60.0,  0.0)( 65.0, 15.0)( 70.0, 30.0)( 75.0, 45.0)
      ( 65.0,-15.0)( 75.0,  0.0)( 85.0, 15.0)( 95.0, 30.0)
      ( 70.0,-30.0)( 85.0,-15.0)(100.0,  0.0)(115.0, 15.0)
      ( 75.0,-45.0)( 95.0,-30.0)(115.0,-15.0)(135.0,  0.0)
```

3.2.17 ZAN1HM, CAN1HM
複素行列 (2 次元配列型) (複素引数型) の積 ($C = C \pm A^*B$)

(1) **機 能**

複素行列の行列積 ($C = [C\pm]A^*B$) を求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZAN1HM (A, LNA, NM, NN, B, LNB, NL, C, LMC, ISW, IERR)

単精度サブルーチン:

CALL CAN1HM (A, LNA, NM, NN, B, LNB, NL, C, LMC, ISW, IERR)

(3) **引 数**

D:倍精度実数型 Z:倍精度複素数型
R:単精度実数型 C:単精度複素数型
I: 32 ビット整数版では INTEGER(4)
64 ビット整数版では INTEGER(8)

項番	引数名	型	大きさ	入出力	内 容
1	A	Z C	LNA, NM	入 力	複素行列 A (2 次元配列型)
2	LNA	I	1	入 力	配列 A の整合寸法
3	NM	I	1	入 力	行列 A の列数 (行列 C の行数)
4	NN	I	1	入 力	行列 A の行数 (行列 B の行数)
5	B	Z C	LNB, NL	入 力	複素行列 B (2 次元配列型)
6	LNB	I	1	入 力	配列 B の整合寸法
7	NL	I	1	入 力	行列 B の列数 (行列 C の列数)
8	C	Z C	LMC, NL	入 力	初期複素行列 C (ISW = ±1 の時)(2 次元配列型)
				出 力	行列積 ($C = [C\pm]A^*B$)
9	LMC	I	1	入 力	配列 C の整合寸法
10	ISW	I	1	入 力	処理スイッチ ISW = 1 の時: $C = C + A^*B$ ISW = 0 の時: $C = A^*B$ ISW = −1 の時: $C = C - A^*B$
11	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{NM} \leq \mathrm{LMC}$

(b) $0 < \mathrm{NN} \leq \mathrm{LNA}, \mathrm{LNB}$

(c) $\mathrm{NL} > 0$

(d) $\mathrm{ISW} \in \{0, 1, -1\}$

(5) **エラーインディケータ**

IERR の値	意　　　　　味	処　理　内　容
0	正常終了.	
1000	NN = 1 であった.	処理を続ける.
3000	制限条件 (a), (b) または (c) を満足しなかった.	処理を打ち切る.
3010	制限条件 (d) を満足しなかった.	

(6) **注意事項**

なし.

(7) **使用例**

(a) 問 題

$$A = \begin{bmatrix} 1+i & 1+2i & 1+3i & 1+4i & 1+5i \\ 2+i & 2+2i & 2+3i & 2+4i & 2+5i \\ 3+i & 3+2i & 3+3i & 3+4i & 3+5i \\ 4+i & 4+2i & 4+3i & 4+4i & 4+5i \end{bmatrix}$$

$$B = \begin{bmatrix} 1+i & 1+2i & 1+3i & 1+4i \\ 2+i & 2+2i & 2+3i & 2+4i \\ 3+i & 3+2i & 3+3i & 3+4i \\ 4+i & 4+2i & 4+3i & 4+4i \end{bmatrix}$$

$C = A^*B$ を求める.

(b) 入力データ

行列 A, 行列 B, LNA = 11, LNB = 11, LNC = 11, NM = 5, NN = 4, NL = 4, ISW = 0.

(c) 主プログラム

```
      PROGRAM AAN1HM
! *** EXAMPLE OF ZAN1HM ***
      IMPLICIT NONE
      INTEGER LNA,LNB,LMC,ISW,NM,NN,NL
      PARAMETER( LNA=11, LNB=11, LMC=11, ISW=0 )
      PARAMETER( NM=5, NN=4, NL=4 )
      INTEGER IERR,I,J
      COMPLEX(8) A(LNA,NM),B(LNB,NL),C(LMC,NL)
!
      DO 100 I=1,NN
         READ(5,*) (A(I,J),J=1,NM)
  100 CONTINUE
      DO 110 I=1,NN
         READ(5,*) (B(I,J),J=1,NL)
  110 CONTINUE
!
      WRITE(6,6000) LNA,LNB,LMC,NM,NN,NL,ISW
      WRITE(6,6010) 'MATRIX A'
      DO 120 I=1,NN
         WRITE(6,6020) (A(I,J),J=1,NM)
  120 CONTINUE
      WRITE(6,6010) 'MATRIX B'
      DO 130 I=1,NN
         WRITE(6,6030) (B(I,J),J=1,NL)
  130 CONTINUE
!
      CALL ZAN1HM(A,LNA,NM,NN,B,LNB,NL,C,LMC,ISW,IERR)
!
      WRITE(6,6040) IERR
      IF( IERR .GE. 3000 ) STOP
!
      WRITE(6,6050) 'MATRIX C'
      DO 140 I=1,NM
         WRITE(6,6030) (C(I,J),J=1,NL)
  140 CONTINUE
      STOP
!
 6000 FORMAT(/,&
             1X,'***  ZAN1HM  ***',/,/,&
             1X,' **  INPUT   **',/,/,&
             1X,'     LNA=',I2,'   LNB=',I2,'   LMC=',I2,/,/,&
```

```
            1X,'     NM =',I2,'   NN =',I2,'   NL =',I2,/,/,&
            1X,'     ISW=',I2,/,/,&
            1X,'     INPUT MATRIX')
6010 FORMAT(/,&
            1X,5X,A)
6030 FORMAT(1X,4X,4('(',F5.1,',',F5.1,')'))
6020 FORMAT(1X,4X,5('(',F5.1,',',F5.1,')'))
6040 FORMAT(/,&
            1X,' **  OUTPUT  **',/,/,&
            1X,'     IERR = ',I4,/)
6050 FORMAT(1X,'     OUTPUT MATRIX',/,/,&
            1X,5X,A)
     END
```

(d) 出力結果

```
***  ZAN1HM  ***

 **  INPUT  **

     LNA=11   LNB=11   LMC=11

     NM = 5   NN = 4   NL = 4

     ISW= 0

     INPUT MATRIX

     MATRIX A
    (  1.0,  1.0)(  1.0,  2.0)(  1.0,  3.0)(  1.0,  4.0)(  1.0,  5.0)
    (  2.0,  1.0)(  2.0,  2.0)(  2.0,  3.0)(  2.0,  4.0)(  2.0,  5.0)
    (  3.0,  1.0)(  3.0,  2.0)(  3.0,  3.0)(  3.0,  4.0)(  3.0,  5.0)
    (  4.0,  1.0)(  4.0,  2.0)(  4.0,  3.0)(  4.0,  4.0)(  4.0,  5.0)

     MATRIX B
    (  1.0,  1.0)(  1.0,  2.0)(  1.0,  3.0)(  1.0,  4.0)
    (  2.0,  1.0)(  2.0,  2.0)(  2.0,  3.0)(  2.0,  4.0)
    (  3.0,  1.0)(  3.0,  2.0)(  3.0,  3.0)(  3.0,  4.0)
    (  4.0,  1.0)(  4.0,  2.0)(  4.0,  3.0)(  4.0,  4.0)

 **  OUTPUT  **

     IERR =    0

     OUTPUT MATRIX

     MATRIX C
    ( 34.0,  0.0)( 38.0, 10.0)( 42.0, 20.0)( 46.0, 30.0)
    ( 38.0,-10.0)( 46.0,  0.0)( 54.0, 10.0)( 62.0, 20.0)
    ( 42.0,-20.0)( 54.0,-10.0)( 66.0,  0.0)( 78.0, 10.0)
    ( 46.0,-30.0)( 62.0,-20.0)( 78.0,-10.0)( 94.0,  0.0)
    ( 50.0,-40.0)( 70.0,-30.0)( 90.0,-20.0)(110.0,-10.0)
```

3.2.18 ZAN1HH, CAN1HH
複素行列 (2 次元配列型) (複素引数型) の積 ($C = C \pm A^*B^*$)

(1) **機 能**

実行列の行列積 ($C = [C\pm]A^*B^*$) を求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZAN1HH (A, LNA, NM, NN, B, LLB, NL, C, LMC, ISW, IERR)

単精度サブルーチン:

CALL CAN1HH (A, LNA, NM, NN, B, LLB, NL, C, LMC, ISW, IERR)

(3) **引 数**

D:倍精度実数型 Z:倍精度複素数型
R:単精度実数型 C:単精度複素数型

I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\ \text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$

項番	引数名	型	大きさ	入出力	内 容
1	A	$\left\{\begin{matrix}Z\\C\end{matrix}\right\}$	LNA, NM	入 力	行列 A (2 次元配列型)
2	LNA	I	1	入 力	配列 A の整合寸法
3	NM	I	1	入 力	行列 A の列数 (行列 C の行数)
4	NN	I	1	入 力	行列 A の行数 (行列 B の列数)
5	B	$\left\{\begin{matrix}Z\\C\end{matrix}\right\}$	LLB, NN	入 力	行列 B (2 次元配列型)
6	LLB	I	1	入 力	配列 B の整合寸法
7	NL	I	1	入 力	行列 B の行数 (行列 C の列数)
8	C	$\left\{\begin{matrix}Z\\C\end{matrix}\right\}$	LMC, NL	入 力	初期複素行列 C (ISW $= \pm 1$ の時)(2 次元配列型)
				出 力	行列積 ($C = [C\pm]A^*B^*$)
9	LMC	I	1	入 力	配列 C の整合寸法
10	ISW	I	1	入 力	処理スイッチ ISW $= 1$ の時: $C = C + A^*B^*$ ISW $= 0$ の時: $C = A^*B^*$ ISW $= -1$ の時: $C = C - A^*B^*$
11	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 <$ NM $\leq$ LMC

(b) $0 <$ NN $\leq$ LNA

(c) $0 <$ NL $\leq$ LNB

(d) ISW $\in \{0, 1, -1\}$

(5) エラーインディケータ

IERR の値	意　　　味	処　理　内　容
0	正常終了.	
1000	NN = 1 であった.	処理を続ける.
3000	制限条件 (a), (b) または (c) を満足しなかった.	処理を打ち切る.
3010	制限条件 (d) を満足しなかった.	

(6) 注意事項

なし.

(7) 使用例

(a) 問 題

$$A = \begin{bmatrix} 1+i & 1+2i & 1+3i & 1+4i & 1+5i \\ 2+i & 2+2i & 2+3i & 2+4i & 2+5i \\ 3+i & 3+2i & 3+3i & 3+4i & 3+5i \\ 4+i & 4+2i & 4+3i & 4+4i & 4+5i \\ 5+i & 5+2i & 5+3i & 5+4i & 5+5i \end{bmatrix}$$

$$B = \begin{bmatrix} 1+i & 1+2i & 1+3i & 1+4i & 1+5i \\ 2+i & 2+2i & 2+3i & 2+4i & 2+5i \\ 3+i & 3+2i & 3+3i & 3+4i & 3+5i \\ 4+i & 4+2i & 4+3i & 4+4i & 4+5i \end{bmatrix}$$

$C = A^* B^*$ を求める.

(b) 入力データ

行列 A, 行列 B, LNA = 11, LLB = 11, LNC = 11, NM = 5, NN = 5, NL = 4, ISW = 0.

(c) 主プログラム

```
      PROGRAM AAN1HH
! *** EXAMPLE OF ZAN1HH ***
      IMPLICIT NONE
      INTEGER LNA,LLB,LMC,ISW,NM,NN,NL
      PARAMETER( LNA=11, LLB=11, LMC=11, ISW=0 )
      PARAMETER( NM=5, NN=5, NL=4 )
      INTEGER IERR,I,J
      COMPLEX(8) A(LNA,NM),B(LLB,NN),C(LMC,NL)
!
      DO 100 I=1,NN
         READ(5,*) (A(I,J),J=1,NM)
  100 CONTINUE
      DO 110 I=1,NL
         READ(5,*) (B(I,J),J=1,NN)
  110 CONTINUE
!
      WRITE(6,6000) LNA,LLB,LMC,NM,NN,NL,ISW
      WRITE(6,6010) 'MATRIX A'
      DO 120 I=1,NN
         WRITE(6,6020) (A(I,J),J=1,NM)
  120 CONTINUE
      WRITE(6,6010) 'MATRIX B'
      DO 130 I=1,NL
         WRITE(6,6020) (B(I,J),J=1,NN)
  130 CONTINUE
!
      CALL ZAN1HH(A,LNA,NM,NN,B,LLB,NL,C,LMC,ISW,IERR)
!
      WRITE(6,6030) IERR
      IF( IERR .GE. 3000 ) STOP
!
      WRITE(6,6040) 'MATRIX C'
      DO 140 I=1,NM
         WRITE(6,6050) (C(I,J),J=1,NL)
  140 CONTINUE
      STOP
!
 6000 FORMAT(/,&
             1X,'***  ZAN1HH  ***',/,/,&
```

```
            1X,' **  INPUT  **',/,/,&
            1X,'     LNA=',I2,'   LLB=',I2,'   LMC=',I2,/,/,&
            1X,'     NM =',I2,'   NN =',I2,'   NL =',I2,/,/,&
            1X,'     ISW=',I2,/,/,&
            1X,'     INPUT MATRIX')
6010 FORMAT(/,&
            1X,5X,A)
6020 FORMAT(1X,4X,5('(',F5.1,',',F5.1,')'))
6050 FORMAT(1X,4X,4('(',F6.1,',',F6.1,')'))
6030 FORMAT(/,&
            1X,' **  OUTPUT  **',/,/,&
            1X,'     IERR = ',I4,/)
6040 FORMAT(1X,'     OUTPUT MATRIX',/,/,&
            1X,5X,A)
     END
```

(d) 出力結果

```
***  ZAN1HH  ***

 **  INPUT  **

     LNA=11   LLB=11   LMC=11

     NM = 5   NN = 5   NL = 4

     ISW= 0

     INPUT MATRIX

     MATRIX A
    (  1.0,  1.0)(  1.0,  2.0)(  1.0,  3.0)(  1.0,  4.0)(  1.0,  5.0)
    (  2.0,  1.0)(  2.0,  2.0)(  2.0,  3.0)(  2.0,  4.0)(  2.0,  5.0)
    (  3.0,  1.0)(  3.0,  2.0)(  3.0,  3.0)(  3.0,  4.0)(  3.0,  5.0)
    (  4.0,  1.0)(  4.0,  2.0)(  4.0,  3.0)(  4.0,  4.0)(  4.0,  5.0)
    (  5.0,  1.0)(  5.0,  2.0)(  5.0,  3.0)(  5.0,  4.0)(  5.0,  5.0)

     MATRIX B
    (  1.0,  1.0)(  1.0,  2.0)(  1.0,  3.0)(  1.0,  4.0)(  1.0,  5.0)
    (  2.0,  1.0)(  2.0,  2.0)(  2.0,  3.0)(  2.0,  4.0)(  2.0,  5.0)
    (  3.0,  1.0)(  3.0,  2.0)(  3.0,  3.0)(  3.0,  4.0)(  3.0,  5.0)
    (  4.0,  1.0)(  4.0,  2.0)(  4.0,  3.0)(  4.0,  4.0)(  4.0,  5.0)

 **  OUTPUT  **

     IERR =    0

     OUTPUT MATRIX

     MATRIX C
    (   0.0, -60.0)(  15.0, -65.0)(  30.0, -70.0)(  45.0, -75.0)
    ( -15.0, -65.0)(   0.0, -75.0)(  15.0, -85.0)(  30.0, -95.0)
    ( -30.0, -70.0)( -15.0, -85.0)(   0.0,-100.0)(  15.0,-115.0)
    ( -45.0, -75.0)( -30.0, -95.0)( -15.0,-115.0)(   0.0,-135.0)
    ( -60.0, -80.0)( -45.0,-105.0)( -30.0,-130.0)( -15.0,-155.0)
```

3.2.19 DAM1VM, RAM1VM
実行列 (2 次元配列型) とベクトルの積 ($y = Ax$)

(1) **機 能**

実行列 A (2 次元配列型) とベクトル $\boldsymbol{x}$ の積 ($y = Ax$) を求める.

(2) **使用法**

倍精度サブルーチン:

CALL DAM1VM (A, LMA, M, N, X, Y, IERR)

単精度サブルーチン:

CALL RAM1VM (A, LMA, M, N, X, Y, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: { 32 ビット整数版では INTEGER(4)
64 ビット整数版では INTEGER(8) }

項番	引数名	型	大きさ	入出力	内　容
1	A	D R	LMA, N	入 力	実行列 A (2 次元配列型)
2	LMA	I	1	入 力	配列 A の整合寸法
3	M	I	1	入 力	行列 A の行数
4	N	I	1	入 力	行列 A の列数
5	X	D R	N	入 力	乗数ベクトル $\boldsymbol{x}$
6	Y	D R	M	出 力	行列 A とベクトル $\boldsymbol{x}$ の積 ($A\boldsymbol{x}$)
7	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) N > 0

(b) 0 < M ≤ LMA

(5) **エラーインディケータ**

IERR の値	意　味	処 理 内 容
0	正常終了.	
1000	N = 1 であった.	処理を続ける.
3000	制限条件 (a) または (b) を満足しなかった.	処理を打ち切る.

(6) **注意事項**

なし

(7) **使用例**

(a) 問 題

$$A = \begin{bmatrix} 9 & 8 & 7 \\ 8 & 8 & 7 \\ 7 & 7 & 7 \\ 7 & 6 & 6 \\ 6 & 6 & 6 \\ 5 & 6 & 7 \end{bmatrix}$$

$$\boldsymbol{x} = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$$

$\boldsymbol{y}= A\boldsymbol{x}$ を求める.

(b) 入力データ

行列 A, ベクトル $\boldsymbol{x}$, LMA = 11, M = 6, N = 3.

(c) 主プログラム

```
      PROGRAM BAM1VM
! *** EXAMPLE OF DAM1VM ***
      IMPLICIT NONE
      INTEGER LMA,M,N
      PARAMETER( LMA=11, M=6, N=3 )
      INTEGER IERR,I,J
      REAL(8) A(LMA,N),X(N),Y(M)
!
      DO 100 I=1,M
         READ(5,*) (A(I,J),J=1,N)
  100 CONTINUE
      DO 110 I=1,N
         READ(5,*) X(I)
  110 CONTINUE
!
      WRITE(6,6000) LMA,M,N
      DO 120 I=1,M
         WRITE(6,6010) (A(I,J),J=1,N)
  120 CONTINUE
      WRITE(6,6020)
      DO 130 I=1,N
         WRITE(6,6010) X(I)
  130 CONTINUE
!
      CALL DAM1VM(A,LMA,M,N,X,Y,IERR)
!
      WRITE(6,6030) IERR
      IF( IERR .GE. 3000 ) STOP
!
      WRITE(6,6040)
      DO 140 I=1,M
         WRITE(6,6010) Y(I)
  140 CONTINUE
      STOP
!
 6000 FORMAT(/,&
             1X,’***  DAM1VM  ***’,/,/,&
             1X,’ **  INPUT  **’,/,/,&
             1X,’     LMA=’,I2,’   M=’,I2,’   N=’,I2,/,/,&
             1X,’     INPUT MATRIX A’,/)
 6010 FORMAT(1X,6X,11(F7.1))
 6020 FORMAT(/,&
             1X,’     INPUT VECTOR X’,/)
 6030 FORMAT(/,&
             1X,’ **  OUTPUT  **’,/,/,&
             1X,’     IERR = ’,I4,/)
 6040 FORMAT(1X,’     OUTPUT VECTOR Y’,/)
      END
```

(d) 出力結果

```
***  DAM1VM  ***

 **  INPUT  **

     LMA=11   M= 6   N= 3

     INPUT MATRIX A

          9.0    8.0    7.0
          8.0    8.0    7.0
          7.0    7.0    7.0
          7.0    6.0    6.0
          6.0    6.0    6.0
```

```
             5.0    6.0    7.0

        INPUT VECTOR X

             1.0
            -1.0
             1.0

    **  OUTPUT  **

        IERR =    0

        OUTPUT VECTOR Y

             8.0
             7.0
             7.0
             7.0
             6.0
             6.0
```

3.2.20 DAM3VM, RAM3VM
実バンド行列 A (バンド型) とベクトルの積 $(y = Ax)$

(1) **機 能**

実バンド行列 A (バンド型) とベクトル $\boldsymbol{x}$ の積 $(y = Ax)$ を求める.

(2) **使用法**

倍精度サブルーチン:

CALL DAM3VM (A, LMA, N, MU, ML, X, Y, IERR)

単精度サブルーチン:

CALL RAM3VM (A, LMA, N, MU, ML, X, Y, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: $\left\{\begin{array}{l}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{array}\right\}$

項番	引数名	型	大きさ	入出力	内　　容
1	A	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	LMA, N	入 力	実バンド行列 A (バンド型)(付録 B 参照)
2	LMA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A の次数
4	MU	I	1	入 力	行列 A の上バンド幅
5	ML	I	1	入 力	行列 A の下バンド幅
6	X	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	N	入 力	乗数ベクトル$\boldsymbol{x}$
7	Y	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	N	出 力	行列 A とベクトル$\boldsymbol{x}$の積 $(A\boldsymbol{x})$
8	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $\mathrm{N} > 0$

(b) $0 \leq \mathrm{MU} < \mathrm{N}$

(c) $0 \leq \mathrm{ML} < \mathrm{N}$

(d) $\mathrm{MU} + \mathrm{ML} < \mathrm{LMA}$

(5) **エラーインディケータ**

IERR の値	意　　味	処 理 内 容
0	正常終了.	
1000	$\mathrm{N} = 1$ であった.	処理を続ける.
3000	制限条件 (a), (b), (c) または (d) を満足しなかった.	処理を打ち切る.

(6) **注意事項**

なし

(7) **使用例**

(a) 問 題

$$A = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}$$

$$\boldsymbol{x} = \begin{bmatrix} 4 \\ 3 \\ 2 \\ 1 \end{bmatrix}$$

$\boldsymbol{y}= A\boldsymbol{x}$ を求める.

(b) 入力データ

行列 A, ベクトル $\boldsymbol{x}$, LMA = 11, N = 4, MU = 1, ML = 1.

(c) 主プログラム

```
      PROGRAM BAM3VM
! *** EXAMPLE OF DAM3VM ***
      IMPLICIT NONE
      INTEGER LMA,N,MU,ML
      PARAMETER( LMA=11, N=4, MU=1, ML=1 )
      INTEGER LNA
      PARAMETER( LNA=11 )
      INTEGER IERR,I,J,JERR
      REAL(8) A(LMA,N),X(N),Y(N)
      REAL(8) AA(LNA,N)
!
      DO 100 I=1,N
         READ(5,*) (AA(I,J),J=1,N)
  100 CONTINUE
      DO 110 I=1,N
         READ(5,*) X(I)
  110 CONTINUE
!
      WRITE(6,6000) LMA,N,MU,ML
      DO 120 I=1,N
         WRITE(6,6010) (AA(I,J),J=1,N)
  120 CONTINUE
      WRITE(6,6020)
      DO 130 I=1,N
         WRITE(6,6010) X(I)
  130 CONTINUE
!
      CALL DABMCS(AA,LNA,N,MU,ML,A,LMA,JERR)
      IF( JERR .GE. 3000 ) THEN
         WRITE(6,6030) JERR
         STOP
      ENDIF
!
      CALL DAM3VM(A,LMA,N,MU,ML,X,Y,IERR)
!
      WRITE(6,6040) IERR
      IF( IERR .GE. 3000 ) STOP
!
      WRITE(6,6050)
      DO 140 I=1,N
         WRITE(6,6010) Y(I)
  140 CONTINUE
      STOP
!
 6000 FORMAT(/,&
             1X,'***  DAM3VM  ***',/,/,&
             1X,' **  INPUT  **',/,/,&
             1X,'     LMA=',I2,'   N=',I2,'   MU =',I2,'   ML =',I2,/,/,&
             1X,'     INPUT MATRIX A',/)
 6010 FORMAT(1X,6X,11(F7.1))
 6020 FORMAT(/,&
             1X,'     INPUT VECTOR X',/)
 6030 FORMAT(/,&
             1X,'     DABMCS IERR = ',I4,/)
 6040 FORMAT(/,&
             1X,' **  OUTPUT  **',/,/,&
             1X,'     IERR = ',I4,/)
 6050 FORMAT(1X,'     OUTPUT VECTOR Y',/)
      END
```

(d) 出力結果

```
*** DAM3VM ***

 ** INPUT **

    LMA=11   N= 4   MU = 1   ML = 1

    INPUT MATRIX A

         1.0   -1.0    0.0    0.0
        -1.0    2.0   -1.0    0.0
         0.0   -1.0    2.0   -1.0
         0.0    0.0   -1.0    2.0

    INPUT VECTOR X

         4.0
         3.0
         2.0
         1.0

 ** OUTPUT **

    IERR =    0

    OUTPUT VECTOR Y

         1.0
         0.0
         0.0
         0.0
```

3.2.21 DAM4VM, RAM4VM
実対称バンド行列 (対称バンド型) とベクトルの積 ($y = Ax$)

(1) **機 能**

実対称バンド行列 A (対称バンド型) とベクトル $\boldsymbol{x}$ の積 ($y = Ax$) を求める.

(2) **使用法**

倍精度サブルーチン:
　CALL DAM4VM (A, LMA, N, MB, X, Y, IERR)
単精度サブルーチン:
　CALL RAM4VM (A, LMA, N, MB, X, Y, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: $\begin{Bmatrix}\text{32 ビット整数版では INTEGER(4)}\\ \text{64 ビット整数版では INTEGER(8)}\end{Bmatrix}$

項番	引数名	型	大きさ	入出力	内　　容
1	A	$\begin{Bmatrix}\text{D}\\ \text{R}\end{Bmatrix}$	LMA, N	入 力	実対称バンド行列 A(対称バンド型) (付録 B 参照)
2	LMA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A の次数
4	MB	I	1	入 力	行列 A のバンド幅
5	X	$\begin{Bmatrix}\text{D}\\ \text{R}\end{Bmatrix}$	N	入 力	乗数ベクトル$\boldsymbol{x}$
6	Y	$\begin{Bmatrix}\text{D}\\ \text{R}\end{Bmatrix}$	N	出 力	行列 A とベクトル$\boldsymbol{x}$の積 ($A\boldsymbol{x}$)
7	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $\mathrm{N} > 0$

(b) $0 \leq \mathrm{MB} < \mathrm{N}$

(c) $\mathrm{MB} < \mathrm{LMA}$

(5) **エラーインディケータ**

IERR の値	意　　味	処 理 内 容
0	正常終了.	
1000	$\mathrm{N} = 1$ であった.	処理を続ける.
3000	制限条件 (a), (b) または (c) を満足しなかった.	処理を打ち切る.

(6) **注意事項**

なし

(7) **使用例**

(a) 問 題

$$A = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & -2 \end{bmatrix}$$

$$\boldsymbol{x} = \begin{bmatrix} 4 \\ 3 \\ 2 \\ 1 \end{bmatrix}$$

$\boldsymbol{y}= A\boldsymbol{x}$ を求める.

(b) 入力データ

行列 A, ベクトル $\boldsymbol{x}$, LMA = 11, N = 4, MB = 1.

(c) 主プログラム

```
      PROGRAM BAM4VM
! *** EXAMPLE OF DAM4VM ***
      IMPLICIT NONE
      INTEGER LMA,N,MB
      PARAMETER( LMA=11, N=4, MB=1 )
      INTEGER LNA
      PARAMETER( LNA=11 )
      INTEGER IERR,I,J,JERR
      REAL(8) A(LMA,N),X(N),Y(N)
      REAL(8) AA(LNA,N)
!
      DO 100 I=1,N
         READ(5,*) (AA(I,J),J=1,N)
  100 CONTINUE
      DO 110 I=1,N
         READ(5,*) X(I)
  110 CONTINUE
!
      WRITE(6,6000) LMA,N,MB
      DO 120 I=1,N
         WRITE(6,6010) (AA(I,J),J=1,N)
  120 CONTINUE
      WRITE(6,6020)
      DO 130 I=1,N
         WRITE(6,6010) X(I)
  130 CONTINUE
!
      CALL DASBCS(AA,LNA,N,MB,A,LMA,JERR)
      IF( JERR .GE. 3000 ) THEN
         WRITE(6,6030) JERR
         STOP
      ENDIF
!
      CALL DAM4VM(A,LMA,N,MB,X,Y,IERR)
!
      WRITE(6,6040) IERR
      IF( IERR .GE. 3000 ) STOP
!
      WRITE(6,6050)
      DO 140 I=1,N
         WRITE(6,6010) Y(I)
  140 CONTINUE
      STOP
!
 6000 FORMAT(/,&
             1X,'***  DAM4VM  ***',/,/,&
             1X,' **  INPUT  **',/,/,&
             1X,'     LMA=',I2,'   N=',I2,'   MB=',I2,/,/,&
             1X,'     INPUT MATRIX A',/)
 6010 FORMAT(1X,6X,11(F7.1))
 6020 FORMAT(/,&
             1X,'     INPUT VECTOR X',/)
 6030 FORMAT(/,&
             1X,'     DASBCS IERR = ',I4,/)
 6040 FORMAT(/,&
             1X,' **  OUTPUT  **',/,/,&
             1X,'     IERR = ',I4,/)
 6050 FORMAT(1X,'     OUTPUT VECTOR Y',/)
      END
```

(d) 出力結果

```
***  DAM4VM  ***

 **  INPUT  **

     LMA=11   N= 4   MB= 1

     INPUT MATRIX A

          1.0   -1.0    0.0    0.0
         -1.0    2.0   -1.0    0.0
          0.0   -1.0    2.0   -1.0
          0.0    0.0   -1.0    2.0

     INPUT VECTOR X

          4.0
          3.0
          2.0
          1.0

 **  OUTPUT  **

     IERR =    0

     OUTPUT VECTOR Y

          1.0
          0.0
          0.0
          0.0
```

3.2.22 DAM1TP, RAM1TP
実行列 (2 次元配列型) の転置 ($B = A^T$)

(1) **機 能**

実行列 A (2 次元配列型) の転置行列 ($B = A^T$) を求める.

(2) **使用法**

倍精度サブルーチン:

CALL DAM1TP (A, LMA, M, N, B, LNB, IERR)

単精度サブルーチン:

CALL RAM1TP (A, LMA, M, N, B, LNB, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: { 32 ビット整数版では INTEGER(4)
64 ビット整数版では INTEGER(8) }

項番	引数名	型	大きさ	入出力	内　　容
1	A	{D R}	LMA, N	入 力	実行列 A (2 次元配列型)
2	LMA	I	1	入 力	配列 A の整合寸法
3	M	I	1	入 力	行列 A の行数
4	N	I	1	入 力	行列 A の列数
5	B	{D R}	LNB, M	出 力	行列 A の転置行列 A^T (2 次元配列型)
6	LNB	I	1	入 力	配列 B の整合寸法
7	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{M} \le \mathrm{LMA}$

(b) $0 < \mathrm{N} \le \mathrm{LNB}$

(5) **エラーインディケータ**

IERR の値	意　　味	処 理 内 容
0	正常終了.	
1000	N = 1 であった.	処理を続ける.
3000	制限条件 (a) または (b) を満足しなかった.	処理を打ち切る.

(6) **注意事項**

なし

(7) **使用例**

(a) 問 題

$$A = \begin{bmatrix} 11 & 12 & 13 & 0 \\ 21 & 22 & 23 & 24 \\ 31 & 32 & 33 & 34 \\ 0 & 42 & 43 & 44 \end{bmatrix}$$

$B = A^T$ を求める.

(b) 入力データ

行列 A, LMA = 11, LNB = 11 M = 4, N = 4.

(c) 主プログラム

```
      PROGRAM BAM1TP
! *** EXAMPLE OF DAM1TP ***
      IMPLICIT NONE
      INTEGER LMA,LNB,M,N
      PARAMETER( LMA=11, LNB=11, M=4, N=4 )
      INTEGER IERR,I,J
      REAL(8) A(LMA,N),B(LNB,M)
!
      DO 100 I=1,M
         READ(5,*) (A(I,J),J=1,N)
  100 CONTINUE
!
      WRITE(6,6000) LMA,LNB,M,N
      DO 110 I=1,M
         WRITE(6,6010) (A(I,J),J=1,N)
  110 CONTINUE
!
      CALL DAM1TP(A,LMA,M,N,B,LNB,IERR)
!
      WRITE(6,6020) IERR
      IF( IERR .GE. 3000 ) STOP
!
      WRITE(6,6030)
      DO 120 I=1,N
         WRITE(6,6010) (B(I,J),J=1,M)
  120 CONTINUE
      STOP
!
 6000 FORMAT(/,&
             1X,’***  DAM1TP  ***’,/,/,&
             1X,’ **  INPUT  **’,/,/,&
             1X,’     LMA=’,I2,’   LNB=’,I2,’   M=’,I2,’   N=’,I2,/,/,&
             1X,’     INPUT MATRIX A’,/)
 6010 FORMAT(1X,6X,11(F7.1))
 6020 FORMAT(/,&
             1X,’ **  OUTPUT  **’,/,/,&
             1X,’     IERR = ’,I4,/)
 6030 FORMAT(1X,’     OUTPUT MATRIX B’,/)
      END
```

(d) 出力結果

```
***  DAM1TP  ***

 **  INPUT  **

     LMA=11   LNB=11   M= 4   N= 4

     INPUT MATRIX A

         11.0   12.0   13.0   14.0
         21.0   22.0   23.0   24.0
         31.0   32.0   33.0   34.0
         41.0   42.0   43.0   44.0

 **  OUTPUT  **

     IERR =    0

     OUTPUT MATRIX B

         11.0   21.0   31.0   41.0
         12.0   22.0   32.0   42.0
         13.0   23.0   33.0   43.0
         14.0   24.0   34.0   44.0
```

3.2.23 DAM3TP, RAM3TP
実バンド行列 (バンド型) の転置 ($B = A^T$)

(1) **機能**

実バンド行列 A (バンド型) の転置行列 ($B = A^T$) を求める.

(2) **使用法**

倍精度サブルーチン:

CALL DAM3TP (A, LMA, N, MU, ML, B, LMB, IERR)

単精度サブルーチン:

CALL RAM3TP (A, LMA, N, MU, ML, B, LMB, IERR)

(3) **引数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: 32 ビット整数版では INTEGER(4)
　 64 ビット整数版では INTEGER(8)

項番	引数名	型	大きさ	入出力	内容
1	A	D R	LMA, N	入力	実バンド行列 A(バンド型)(付録 B 参照)
2	LMA	I	1	入力	配列 A の整合寸法
3	N	I	1	入力	行列 A の次数
4	MU	I	1	入力	行列 A の上バンド幅
5	ML	I	1	入力	行列 A の下バンド幅
6	B	D R	LMB, N	出力	行列 A の転置行列 A^T (バンド型)(付録 B 参照)
7	LMB	I	1	入力	配列 B の整合寸法
8	IERR	I	1	出力	エラーインディケータ

(4) **制限条件**

(a) $\mathrm{N} > 0$

(b) $0 \leq \mathrm{MU} < \mathrm{N}$

(c) $0 \leq \mathrm{ML} < \mathrm{N}$

(d) $\mathrm{MU} + \mathrm{ML} < \mathrm{LMA}, \mathrm{LMB}$

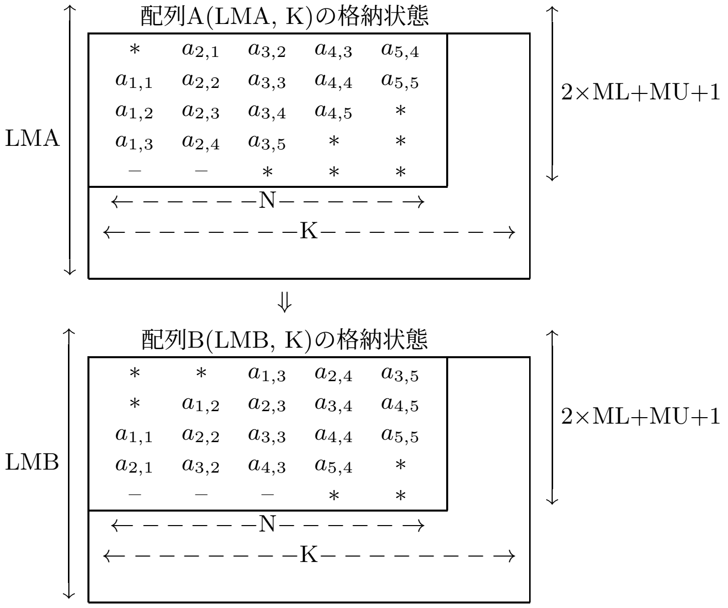
(5) **エラーインディケータ**

IERR の値	意味	処理内容
0	正常終了.	
1000	$\mathrm{N} = 1$ であった.	処理を続ける.
3000	制限条件 (a), (b), (c) または (d) を満足しなかった.	処理を打ち切る.

(6) **注意事項**

(a) 配列 B の要素のうち行列 A の要素が格納されない要素については, 引用時の値がそのまま保存される.

例



備 考

a. $*$ と–は, 入力時の値を保つ.
b. –は, 行列の LU 分解時に必要となる領域である.
c. MU は上バンド幅, ML は下バンド幅である.
d. LMB $>$ ML + MU, K $\geq$ N を満たさなければならない. (ただし, 変換後 LU 分解を行う場合には, LMB $\geq 2 \times$ ML + MU + 1, K $\geq$ N.)

(7) **使用例**

(a) 問 題

$$A = \begin{bmatrix} 11 & 12 & 13 & 0 \\ 21 & 22 & 23 & 24 \\ 0 & 32 & 33 & 34 \\ 0 & 0 & 43 & 44 \end{bmatrix}$$

$B = A^T$ を求める.

(b) 入力データ

行列 A, LMA = 11, LMB = 11, N = 4, MU = 2, ML = 1.

(c) 主プログラム

```
      PROGRAM BAM3TP
! *** EXAMPLE OF DAM3TP ***
      IMPLICIT NONE
      INTEGER LMA,LMB,N,MU,ML
      PARAMETER( LMA=11, LMB=11, N=4, MU=2, ML=1 )
      INTEGER LNA
      PARAMETER( LNA=11 )
      INTEGER IERR,I,J,JERR
      REAL(8) A(LMA,N),B(LMB,N)
      REAL(8) AA(LNA,N)
!
      DO 100 I=1,N
         READ(5,*) (AA(I,J),J=1,N)
  100 CONTINUE
!
      WRITE(6,6000) LMA,LMB,N,MU,ML
      DO 110 I=1,N
         WRITE(6,6010) (AA(I,J),J=1,N)
  110 CONTINUE
!
      CALL DABMCS(AA,LNA,N,MU,ML,A,LMA,JERR)
      IF( JERR .GE. 3000 ) THEN
```

```
            WRITE(6,6020) JERR
            STOP
         ENDIF
!
         CALL DAM3TP(A,LMA,N,MU,ML,B,LMB,IERR)
!
         WRITE(6,6030) IERR
         IF( IERR .GE. 3000 ) STOP
!
         CALL DABMEL(B,LMB,N,ML,MU,AA,LNA,JERR)
         IF( JERR .GE. 3000 ) THEN
            WRITE(6,6040) JERR
            STOP
         ENDIF
!
         WRITE(6,6050)
         DO 120 I=1,N
            WRITE(6,6010) (AA(I,J),J=1,N)
   120 CONTINUE
         STOP
!
  6000 FORMAT(/,&
                1X,'***  DAM3TP  ***',/,/,&
                1X,' **  INPUT  **',/,/,&
                1X,'     LMA=',I2,'   LMB=',I2,/,/,&
                1X,'     N  =',I2,'   MU =',I2,'   ML =',I2,/,/,&
                1X,'     INPUT MATRIX A',/)
  6010 FORMAT(1X,6X,11(F7.1))
  6020 FORMAT(/,&
                1X,'     DABMCS IERR = ',I4,/)
  6030 FORMAT(/,&
                1X,' **  OUTPUT  **',/,/,&
                1X,'     IERR = ',I4,/)
  6040 FORMAT(/,&
                1X,'     DABMEL IERR = ',I4,/)
  6050 FORMAT(1X,'     OUTPUT MATRIX B',/)
         END
```

(d) 出力結果

```
 ***  DAM3TP  ***

  **  INPUT  **

      LMA=11   LMB=11

      N  = 4   MU = 2   ML = 1

      INPUT MATRIX A

         11.0   12.0   13.0    0.0
         21.0   22.0   23.0   24.0
          0.0   32.0   33.0   34.0
          0.0    0.0   43.0   44.0

  **  OUTPUT  **

      IERR =    0

      OUTPUT MATRIX B

         11.0   21.0    0.0    0.0
         12.0   22.0   32.0    0.0
         13.0   23.0   33.0   43.0
          0.0   24.0   34.0   44.0
```

3.2.24 DAMVJ1, RAMVJ1
実不規則スパース行列 (JAD 型) の行列ベクトル積 ($\boldsymbol{y} = \beta\boldsymbol{y} + \alpha A\boldsymbol{x}$)

(1) 機 能

実不規則スパース行列 A(JAD 型) とベクトル $\boldsymbol{x}$ の積 ($\boldsymbol{y} = \beta\boldsymbol{y} + \alpha A\boldsymbol{x}$) を求める.

(2) 使用法

倍精度サブルーチン:

CALL DAMVJ1 (AJAD, LXA, IAJAD, JAJAD, JADORD, N, MJAD, ALPHA, BETA, X,
Y, W, IERR)

単精度サブルーチン:

CALL RAMVJ1 (AJAD, LXA, IAJAD, JAJAD, JADORD, N, MJAD, ALPHA, BETA, X,
Y, W, IERR)

(3) 引 数

D:倍精度実数型　Z:倍精度複素数型　I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$
R:単精度実数型　C:単精度複素数型

項番	引数名	型	大きさ	入出力	内容
1	AJAD	$\left\{\begin{matrix}\text{D}\\\text{R}\end{matrix}\right\}$	LXA	入 力	スパース行列 A(JAD 格納型)(付録 B 参照)
2	LXA	I	1	入 力	配列 AJAD および配列 JAJAD に割り当てる大きさ
3	IAJAD	I	MJAD+1	入 力	スパース行列 A(JAD 格納型) のインデックス配列 (付録 B 参照)
4	JAJAD	I	LXA	入 力	スパース行列 A(JAD 格納型) のインデックス配列 (付録 B 参照)
5	JADORD	I	N	入 力	スパース行列 A(JAD 格納型) のインデックス配列 (付録 B 参照)
6	N	I	1	入 力	ベクトル X 及び Y の次数
7	MJAD	I	1	入 力	行列 A の JAD 格納型における, jagged diagonal の本数
8	ALPHA	$\left\{\begin{matrix}\text{D}\\\text{R}\end{matrix}\right\}$	1	入 力	$\boldsymbol{y} = \beta\boldsymbol{y} + \alpha A\boldsymbol{x}$ の乗数 α
9	BETA	$\left\{\begin{matrix}\text{D}\\\text{R}\end{matrix}\right\}$	1	入 力	$\boldsymbol{y} = \beta\boldsymbol{y} + \alpha A\boldsymbol{x}$ の乗数 β
10	X	$\left\{\begin{matrix}\text{D}\\\text{R}\end{matrix}\right\}$	N	入 力	$\boldsymbol{y} = \beta\boldsymbol{y} + \alpha A\boldsymbol{x}$ のベクトル $\boldsymbol{x}$
11	Y	$\left\{\begin{matrix}\text{D}\\\text{R}\end{matrix}\right\}$	N	入出力	$\boldsymbol{y} = \beta\boldsymbol{y} + \alpha A\boldsymbol{x}$ のベクトル $\boldsymbol{y}$
12	W	$\left\{\begin{matrix}\text{D}\\\text{R}\end{matrix}\right\}$	N	ワーク	作業配列
13	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a) N > 0

(b) 0 < MJAD ≤ N

(c) IAJAD(MJAD + 1) − IAJAD(1) ≤ LXA

(5) エラーインディケータ

IERR の値	意　　　味	処　理　内　容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3100	制限条件 (b) を満足しなかった (N, A, IA, JA の入力値に矛盾がある).	
3200	制限条件 (c) を満足しなかった (配列 AJAD, JAJAD の大きさが不足).	

(6) 注意事項

(a) 行列 A が, 3×3 または 4×4 の行列をブロック行列要素に持つスパース行列の行列ベクトル積を求める場合は, 3.2.25 $\left\{\begin{matrix}\text{DAMVJ3}\\ \text{RAMVJ3}\end{matrix}\right\}$ または 3.2.26 $\left\{\begin{matrix}\text{DAMVJ4}\\ \text{RAMVJ4}\end{matrix}\right\}$ を用いて計算する方が性能が良い.

(b) 本サブルーチンを使用するための行列の格納形式の変換は, なるべく回数を削減した方がよい. 例えば, スパース行列の連立 1 次方程式や固有値方程式の反復解法等で, 行列 A を変更せずに行列ベクトル積を繰り返し計算する場合は, 反復ループの外で 2.2.5 $\left\{\begin{matrix}\text{DARSJD}\\ \text{RARSJD}\end{matrix}\right\}$ や 2.2.6 $\left\{\begin{matrix}\text{DARGJM}\\ \text{RARGJM}\end{matrix}\right\}$ を用いて一度だけ格納形式の変換を行い, 反復ループ内で本サブルーチンを繰り返し使えば効率良く計算できる.

(7) 使用例

(a) 問 題

実不規則スパース行列 A を (実 1 次元行方向ブロックリスト型) で配列 A に保持し, 内部的に (JAD 格納型) で配列 AJAD に格納してから, 行列ベクトル積 $\boldsymbol{y} = \beta\boldsymbol{y} + \alpha A\boldsymbol{x}$ を求める.

(b) 主プログラム

```
      PROGRAM BAMVJ1
! *** EXAMPLE OF DAMVJ1 ***
      IMPLICIT NONE
      INTEGER N,M,NZ,ISW,LXA,LXIA
      PARAMETER( N=4, M=1, NZ=8, ISW=0, LXA=NZ, LXIA=N+1 )
      INTEGER IA(N+1),JA(NZ),MJAD,IAJAD(LXIA),JAJAD(LXA),JADORD(N)
      INTEGER IW(N*2+1),IERR
      INTEGER I,J
      REAL(8) A(NZ),AJAD(LXA),X(N),Y(N),W(N)
      REAL(8) ALPHA,BETA
      PARAMETER (ALPHA=1.0D0, BETA=1.0D0)
      CHARACTER*80 FMT1,FMT2,FMT3,FMT4,FMT5
!
      READ(5,*) (IA(I),I=1,N+1)
      READ(5,*) (JA(I),I=1,NZ)
      DO 100 I=1,N
         DO 110 J=IA(I),IA(I+1)-1
            A(J) = J
  110    CONTINUE
  100 CONTINUE
      DO 120 I=1,N
         X(I) = I
         Y(I) = N - I + 1
  120 CONTINUE
!
      WRITE(6,6000)
      WRITE(6,6010) 'IA IN CSR'
      WRITE(FMT1,7000) N+1
      WRITE(6,FMT1) (IA(I),I=1,N+1)
```

```
      WRITE(6,6010) 'JA IN CSR:'
      DO 130 I=1,N
         WRITE(FMT2,7000) IA(I+1) - IA(I)
         WRITE(6,FMT2) (JA(J),J=IA(I),IA(I+1)-1)
  130 CONTINUE
      WRITE(6,6010) 'A IN CSR:'
      DO 140 I=1,N
         WRITE(FMT3,7010) IA(I+1) - IA(I)
         WRITE(6,FMT3)( A(J), J=IA(I),IA(I+1)-1 )
  140 CONTINUE
!
      WRITE(FMT4,7010) 1
      WRITE(6,6010) 'ALPHA:'
      WRITE(6,FMT4) ALPHA
      WRITE(6,6010) 'BETA:'
      WRITE(6,FMT4) BETA
      WRITE(6,6010) 'VECTOR X:'
      WRITE(FMT5,7010) N
      WRITE(6,FMT5) (X(I),I=1,N)
      WRITE(6,6010) 'VECTOR Y:'
      WRITE(6,FMT5) (Y(I),I=1,N)
!
!     CONVERT FROM CSR TO JAD
!
      CALL DARGJM&
      (N,M,A,IA,JA,ISW,LXA,LXIA,MJAD,AJAD,IAJAD,JAJAD,JADORD,IW,IERR)
!
!     MATRIX-VECTOR PRODUCT  Y=BETA*Y+ALPHA*A*X
!
      CALL DAMVJ1&
      (AJAD,LXA,IAJAD,JAJAD,JADORD,N,MJAD,ALPHA,BETA,X,Y,W,IERR)
!
      WRITE(6,6030) IERR
      IF( IERR .GE. 3000 ) STOP
!
      WRITE(6,6040)
      WRITE(6,FMT5) (Y(I),I=1,N)
      STOP
!
 6000 FORMAT(/,&
             1X,'***  DAMVJ1  ***',/,/,&
             1X,' **  INPUT  **',/,/,&
             1X,'   * MATRIX DATA FOR CSR FORMAT *')
 6010 FORMAT(/,&
             1X,'     ',A)
 6020 FORMAT(/,&
             1X,'   * ORIGINAL MATRIX *',/)
 6030 FORMAT(/,&
             1X,' **  OUTPUT  **',/,/,&
             1X,'     IERR = ',I4,/)
 6040 FORMAT(/,&
             1X,'   * RESULT Y=BETA*Y+ALPHA*A*X *')
 7000 FORMAT('(1X,5X,',I2,'(3X,I2))')
 7010 FORMAT('(1X,7X,',I2,'(1X,F4.0))')
      END
```

(c) 出力結果

```
***  DAMVJ1  ***

 **  INPUT  **

   * MATRIX DATA FOR CSR FORMAT *

     IA IN CSR
         1    3    6    7    9

     JA IN CSR:
         1    3
         1    2    3
         3
         3    4

     A IN CSR:
          1.   2.
          3.   4.   5.
          6.
          7.   8.

     ALPHA:
          1.

     BETA:
          1.

     VECTOR X:
          1.   2.   3.   4.

     VECTOR Y:
          4.   3.   2.   1.

 **  OUTPUT  **

     IERR =    0


   * RESULT Y=BETA*Y+ALPHA*A*X *
         11.  29.  20.  54.
```

3.2.25 DAMVJ3, RAMVJ3
実不規則スパース行列 (MJAD 型:3×3 ブロック行列) の行列ベクトル積 ($\boldsymbol{y} = \beta\boldsymbol{y}+\alpha A\boldsymbol{x}$)

(1) 機 能

実不規則スパース行列 A(MJAD 型:3 × 3 ブロック行列) とベクトル $\boldsymbol{x}$ の積 ($\boldsymbol{y} = \beta\boldsymbol{y} + \alpha A\boldsymbol{x}$) を求める.

(2) 使用法

倍精度サブルーチン:

CALL DAMVJ3 (AJAD, LXA, IAJAD, JAJAD, JADORD, NB, MJAD, ALPHA, BETA, X,
Y, W, IERR)

単精度サブルーチン:

CALL RAMVJ3 (AJAD, LXA, IAJAD, JAJAD, JADORD, NB, MJAD, ALPHA, BETA, X,
Y, W, IERR)

(3) 引 数

D:倍精度実数型 Z:倍精度複素数型
R:単精度実数型 C:単精度複素数型
I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$

項番	引数名	型	大きさ	入出力	内 容
1	AJAD	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	内容参照	入 力	スパース行列 A(MJAD 格納型) **大きさ**: LXA×3×3 (付録 B 参照)
2	LXA	I	1	入 力	配列 AJAD および配列 JAJAD に割り当てる大きさ
3	IAJAD	I	MJAD+1	入 力	スパース行列 A(MJAD 格納型) のインデックス配列 (付録 B 参照)
4	JAJAD	I	LXA	入 力	スパース行列 A(MJAD 格納型) のインデックス配列 (付録 B 参照)
5	JADORD	I	NB	入 力	スパース行列 A(MJAD 格納型) のインデックス配列 (付録 B 参照)
6	NB	I	1	入 力	行列 A を 3×3 のブロック行列で区分けした場合の行のブロック数 (または列のブロック数)
7	MJAD	I	1	入 力	行列 A の MJAD 格納型における, jagged diagonal の本数
8	ALPHA	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	1	入 力	$\boldsymbol{y} = \beta\boldsymbol{y} + \alpha A\boldsymbol{x}$ の乗数 α
9	BETA	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	1	入 力	$\boldsymbol{y} = \beta\boldsymbol{y} + \alpha A\boldsymbol{x}$ の乗数 β
10	X	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	NB×3	入 力	$\boldsymbol{y} = \beta\boldsymbol{y} + \alpha A\boldsymbol{x}$ のベクトル $\boldsymbol{x}$
11	Y	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	NB×3	入出力	$\boldsymbol{y} = \beta\boldsymbol{y} + \alpha A\boldsymbol{x}$ のベクトル $\boldsymbol{y}$
12	W	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	NB×3	ワーク	作業配列
13	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) NB > 0

(b) 0 < MJAD ≤ NB

(c) IAJAD(MJAD + 1) − IAJAD(1) ≤ LXA

(5) **エラーインディケータ**

IERR の値	意　　　　　味	処　理　内　容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3100	制限条件 (b) を満足しなかった (NB, A, IA, JA の入力値に矛盾がある).	
3200	制限条件 (c) を満足しなかった (出力配列 AJAD, JAJAD の大きさが不足).	

(6) **注意事項**

(a) 本サブルーチンは, 3×3 のブロック行列を要素に持つスパース行列の行列ベクトル積を求める. 1×1 または 4×4 のブロック行列の場合は, 3.2.24 $\left\{\begin{matrix}\text{DAMVJ1}\\\text{RAMVJ1}\end{matrix}\right\}$ または 3.2.26 $\left\{\begin{matrix}\text{DAMVJ4}\\\text{RAMVJ4}\end{matrix}\right\}$ を使用すればよい.

(b) 本サブルーチンを使用するための行列の格納形式の変換は, なるべく回数を削減した方がよい. 例えば, スパース行列の連立 1 次方程式や固有値方程式の反復解法等で, 行列 A を変更せずに行列ベクトル積を繰り返し計算する場合は, 反復ループの外で 2.2.5 $\left\{\begin{matrix}\text{DARSJD}\\\text{RARSJD}\end{matrix}\right\}$ や 2.2.6 $\left\{\begin{matrix}\text{DARGJM}\\\text{RARGJM}\end{matrix}\right\}$ を用いて一度だけ格納形式の変換を行い, 反復ループ内で本サブルーチンを繰り返し使えば効率良く計算できる.

(7) **使用例**

(a) 問 題

3×3 のブロック行列を要素に持つ実不規則スパース行列 A を (実 1 次元行方向ブロックリスト型) で配列 A に保持し, 内部的に (MJAD 格納型) で配列 AJAD に格納してから, 行列ベクトル積 $\boldsymbol{y} = \beta\boldsymbol{y} + \alpha A\boldsymbol{x}$ を求める.

(b) 主プログラム

```
      PROGRAM BAMVJ3
! *** EXAMPLE OF DAMVJ3 ***
      IMPLICIT NONE
      INTEGER NB,M,NZ,ISW,LXA,LXIA,MM
      PARAMETER( NB=4, M=3, NZ=8, ISW=0, LXA=NZ, LXIA=NB+1, MM=M*M )
      INTEGER IA(NB+1),JA(NZ),MJAD,IAJAD(LXIA),JAJAD(LXA),JADORD(NB)
      INTEGER IW(NB*2+1),IERR
      INTEGER I,J,K,L
      REAL(8) A(NZ*MM),AJAD(LXA,MM),X(NB*M),Y(NB*M),W(NB*M)
      REAL(8) ALPHA,BETA
      PARAMETER (ALPHA=1.0D0, BETA=1.0D0)
      CHARACTER*80 FMT1,FMT2,FMT3,FMT4,FMT5
!
      READ(5,*) (IA(I),I=1,NB+1)
      READ(5,*) (JA(I),I=1,NZ)
      DO 100 I=1,NB
         DO 110 J=IA(I),IA(I+1)-1
            DO 120 K=1,MM
               A( (J-1)*MM + K ) = K
 120        CONTINUE
 110     CONTINUE
 100  CONTINUE
      DO 130 I=1,NB
         DO 140 J=1,M
```

```
            X( (I-1)*M+J ) = I
            Y( (I-1)*M+J ) = NB - I + 1
  140    CONTINUE
  130 CONTINUE
!
      WRITE(6,6000)
      WRITE(6,6010) 'IA IN BLOCK CSR'
      WRITE(FMT1,7000) NB+1
      WRITE(6,FMT1) (IA(I),I=1,NB+1)
      WRITE(6,6010) 'JA IN BLOCK CSR:'
      DO 150 I=1,NB
         WRITE(FMT2,7000) IA(I+1) - IA(I)
         WRITE(6,FMT2) (JA(J),J=IA(I),IA(I+1)-1)
  150 CONTINUE
      WRITE(6,6010) 'A IN BLOCK CSR:'
      DO 160 I=1,NB
         WRITE(FMT3,7010) (IA(I+1) - IA(I))*M
         DO 170 K=1,M
            WRITE(6,FMT3)&
            ( (A((J-1)*MM+(K-1)*M+L), L=1,M) ,J=IA(I),IA(I+1)-1 )
  170    CONTINUE
  160 CONTINUE
!
      WRITE(FMT4,7010) 1
      WRITE(6,6010) 'ALPHA:'
      WRITE(6,FMT4) ALPHA
      WRITE(6,6010) 'BETA:'
      WRITE(6,FMT4) BETA
      WRITE(6,6010) 'BLOCK VECTOR X:'
      WRITE(FMT5,7010) NB*M
      WRITE(6,FMT5) (X(I),I=1,NB*M)
      WRITE(6,6010) 'BLOCK VECTOR Y:'
      WRITE(6,FMT5) (Y(I),I=1,NB*M)
!
!     CONVERT FROM BLOCK CSR TO JAD
!
      CALL DARGJM&
      (NB,M,A,IA,JA,ISW,LXA,LXIA,MJAD,AJAD,IAJAD,JAJAD,JADORD,IW,IERR)
!
!     MATRIX-VECTOR PRODUCT  Y=BETA*Y+ALPHA*A*X
!
      CALL DAMVJ3&
      (AJAD,LXA,IAJAD,JAJAD,JADORD,NB,MJAD,ALPHA,BETA,X,Y,W,IERR)
!
      WRITE(6,6030) IERR
      IF( IERR .GE. 3000 ) STOP
!
      WRITE(6,6040)
      WRITE(6,6010) 'BLOCK VECTOR Y:'
      WRITE(6,FMT5) (Y(I),I=1,NB*M)
      STOP
!
 6000 FORMAT(/,&
             1X,'***  DAMVJ3  ***',/,/,&
             1X,' **  INPUT  **',/,/,&
             1X,'   * MATRIX DATA FOR CSR FORMAT *')
 6010 FORMAT(/,&
             1X,'     ',A)
 6020 FORMAT(/,&
             1X,'   * ORIGINAL MATRIX *',/)
 6030 FORMAT(/,&
             1X,' **  OUTPUT  **',/,/,&
             1X,'     IERR = ',I4,/)
 6040 FORMAT(/,&
             1X,'   * RESULT Y=BETA*Y+ALPHA*A*X *')
 7000 FORMAT('(1X,5X,',I2,'(3X,I2))')
 7010 FORMAT('(1X,7X,',I2,'(1X,F4.0))')
      END
```

(c) 出力結果

```
***  DAMVJ3  ***

 **  INPUT  **

   * MATRIX DATA FOR CSR FORMAT *

     IA IN BLOCK CSR
         1    3    6    7    9

     JA IN BLOCK CSR:
         1    3
         1    2    3
         3
         3    4

     A IN BLOCK CSR:
          1.   2.   3.   1.   2.   3.
          4.   5.   6.   4.   5.   6.
          7.   8.   9.   7.   8.   9.
          1.   2.   3.   1.   2.   3.   1.   2.   3.
          4.   5.   6.   4.   5.   6.   4.   5.   6.
          7.   8.   9.   7.   8.   9.   7.   8.   9.
          1.   2.   3.
          4.   5.   6.
          7.   8.   9.
          1.   2.   3.   1.   2.   3.
          4.   5.   6.   4.   5.   6.
          7.   8.   9.   7.   8.   9.
```

```
        ALPHA:
             1.

        BETA:
             1.

        BLOCK VECTOR X:
             1.   1.   1.   2.   2.   2.   3.   3.   3.   4.   4.   4.

        BLOCK VECTOR Y:
             4.   4.   4.   3.   3.   3.   2.   2.   2.   1.   1.   1.

    **  OUTPUT  **

        IERR =    0


      * RESULT Y=BETA*Y+ALPHA*A*X *

        BLOCK VECTOR Y:
            28.  64. 100.  39.  93. 147.  20.  47.  74.  43. 106. 169.
```

3.2.26 DAMVJ4, RAMVJ4
実不規則スパース行列 (MJAD 型:4×4 ブロック行列) の行列ベクトル積 ($\boldsymbol{y} = \beta\boldsymbol{y}+\alpha A\boldsymbol{x}$)

(1) **機 能**

実不規則スパース行列 A(MJAD 型:4 × 4 ブロック行列) とベクトル $\boldsymbol{y}$ の積 ($\boldsymbol{y} = \beta\boldsymbol{y} + \alpha A\boldsymbol{x}$) を求める.

(2) **使用法**

倍精度サブルーチン:

CALL DAMVJ4 (AJAD, LXA, IAJAD, JAJAD, JADORD, NB, MJAD, ALPHA, BETA, X,
Y, W, IERR)

単精度サブルーチン:

CALL RAMVJ4 (AJAD, LXA, IAJAD, JAJAD, JADORD, NB, MJAD, ALPHA, BETA, X,
Y, W, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型　I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$
R:単精度実数型　C:単精度複素数型

項番	引数名	型	大きさ	入出力	内　容
1	AJAD	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	内容参照	入 力	スパース行列 A(MJAD 格納型) **大きさ**: LXA×4×4　(付録 B 参照)
2	LXA	I	1	入 力	配列 AJAD および配列 JAJAD に割り当てる大きさ
3	IAJAD	I	MJAD+1	入 力	スパース行列 A(MJAD 格納型) のインデックス配列 (付録 B 参照)
4	JAJAD	I	LXA	入 力	スパース行列 A(MJAD 格納型) のインデックス配列 (付録 B 参照)
5	JADORD	I	NB	入 力	スパース行列 A(MJAD 格納型) のインデックス配列 (付録 B 参照)
6	NB	I	1	入 力	行列 A を 4×4 のブロック行列で区分けした場合の行のブロック数 (または列のブロック数)
7	MJAD	I	1	入 力	行列 A の MJAD 格納型における, jagged diagonal の本数
8	ALPHA	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	1	入 力	$\boldsymbol{y} = \beta\boldsymbol{y} + \alpha A\boldsymbol{x}$ の乗数 α
9	BETA	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	1	入 力	$\boldsymbol{y} = \beta\boldsymbol{y} + \alpha A\boldsymbol{x}$ の乗数 β
10	X	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	NB×4	入 力	$\boldsymbol{y} = \beta\boldsymbol{y} + \alpha A\boldsymbol{x}$ のベクトル $\boldsymbol{x}$
11	Y	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	NB×4	入出力	$\boldsymbol{y} = \beta\boldsymbol{y} + \alpha A\boldsymbol{x}$ のベクトル $\boldsymbol{y}$
12	W	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	NB×4	ワーク	作業配列
13	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a) $\mathrm{NB} > 0$

(b) $0 < \mathrm{MJAD} \leq \mathrm{NB}$

(c) $\mathrm{IAJAD(MJAD+1)} - \mathrm{IAJAD(1)} \leq \mathrm{LXA}$

(5) エラーインディケータ

IERR の値	意　　　味	処　理　内　容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3100	制限条件 (b) を満足しなかった (NB, A, IA, JA の入力値に矛盾がある).	
3200	制限条件 (c) を満足しなかった (出力配列 AJAD, JAJAD の大きさが不足).	

(6) 注意事項

(a) 本サブルーチンは, 4×4 のブロック行列を要素に持つスパース行列の行列ベクトル積を求める. 1×1 または 3×3 のブロック行列の場合は, 3.2.24 $\left\{\begin{matrix}\mathrm{DAMVJ1}\\ \mathrm{RAMVJ1}\end{matrix}\right\}$ または 3.2.25 $\left\{\begin{matrix}\mathrm{DAMVJ3}\\ \mathrm{RAMVJ3}\end{matrix}\right\}$ を使用すればよい.

(b) 本サブルーチンを使用するための行列の格納形式の変換は, なるべく回数を削減した方がよい. 例えば, スパース行列の連立 1 次方程式や固有値方程式の反復解法等で, 行列 A を変更せずに行列ベクトル積を繰り返し計算する場合は, 反復ループの外で 2.2.5 $\left\{\begin{matrix}\mathrm{DARSJD}\\ \mathrm{RARSJD}\end{matrix}\right\}$ や 2.2.6 $\left\{\begin{matrix}\mathrm{DARGJM}\\ \mathrm{RARGJM}\end{matrix}\right\}$ を用いて一度だけ格納形式の変換を行い, 反復ループ内で本サブルーチンを繰り返し使えば効率良く計算できる.

(7) 使用例

(a) 問 題

4×4 のブロック行列を要素に持つ実不規則スパース行列 A を (実 1 次元行方向ブロックリスト型) で配列 A に保持し, 内部的に (MJAD 格納型) で配列 AJAD に格納してから, 行列ベクトル積 $\boldsymbol{y} = \beta\boldsymbol{y} + \alpha A\boldsymbol{x}$ を求める.

(b) 主プログラム

```
      PROGRAM BAMVJ4
! *** EXAMPLE OF DAMVJ4 ***
      IMPLICIT NONE
      INTEGER NB,M,NZ,ISW,LXA,LXIA,MM
      PARAMETER( NB=4, M=4, NZ=8, ISW=0, LXA=NZ, LXIA=NB+1, MM=M*M )
      INTEGER IA(NB+1),JA(NZ),MJAD,IAJAD(LXIA),JAJAD(LXA),JADORD(NB)
      INTEGER IW(NB*2+1),IERR
      INTEGER I,J,K,L
      REAL(8) A(NZ*MM),AJAD(LXA,MM),X(NB*M),Y(NB*M),W(NB*M)
      REAL(8) ALPHA,BETA
      PARAMETER (ALPHA=1.0D0, BETA=1.0D0)
      CHARACTER*80 FMT1,FMT2,FMT3,FMT4,FMT5
!
      READ(5,*) (IA(I),I=1,NB+1)
      READ(5,*) (JA(I),I=1,NZ)
      DO 100 I=1,NB
         DO 110 J=IA(I),IA(I+1)-1
            DO 120 K=1,MM
               A( (J-1)*MM + K ) = K
  120       CONTINUE
  110    CONTINUE
  100 CONTINUE
      DO 130 I=1,NB
         DO 140 J=1,M
```

```
            X( (I-1)*M+J ) = I
            Y( (I-1)*M+J ) = NB - I + 1
  140    CONTINUE
  130 CONTINUE
!
      WRITE(6,6000)
      WRITE(6,6010) 'IA IN BLOCK CSR'
      WRITE(FMT1,7000) NB+1
      WRITE(6,FMT1) (IA(I),I=1,NB+1)
      WRITE(6,6010) 'JA IN BLOCK CSR:'
      DO 150 I=1,NB
         WRITE(FMT2,7000) IA(I+1) - IA(I)
         WRITE(6,FMT2) (JA(J),J=IA(I),IA(I+1)-1)
  150 CONTINUE
      WRITE(6,6010) 'A IN BLOCK CSR:'
      DO 160 I=1,NB
         WRITE(FMT3,7010) (IA(I+1) - IA(I))*M
         DO 170 K=1,M
            WRITE(6,FMT3)&
            ( (A((J-1)*MM+(K-1)*M+L), L=1,M) ,J=IA(I),IA(I+1)-1 )
  170    CONTINUE
  160 CONTINUE
!
      WRITE(FMT4,7010) 1
      WRITE(6,6010) 'ALPHA:'
      WRITE(6,FMT4) ALPHA
      WRITE(6,6010) 'BETA:'
      WRITE(6,FMT4) BETA
      WRITE(6,6010) 'BLOCK VECTOR X:'
      WRITE(FMT5,7010) NB*M
      WRITE(6,FMT5) (X(I),I=1,NB*M)
      WRITE(6,6010) 'BLOCK VECTOR Y:'
      WRITE(6,FMT5) (Y(I),I=1,NB*M)
!
!     CONVERT FROM BLOCK CSR TO JAD
!
      CALL DARGJM&
      (NB,M,A,IA,JA,ISW,LXA,LXIA,MJAD,AJAD,IAJAD,JAJAD,JADORD,IW,IERR)
!
!     MATRIX-VECTOR PRODUCT  Y=BETA*Y+ALPHA*A*X
!
      CALL DAMVJ4&
      (AJAD,LXA,IAJAD,JAJAD,JADORD,NB,MJAD,ALPHA,BETA,X,Y,W,IERR)
!
      WRITE(6,6030) IERR
      IF( IERR .GE. 3000 ) STOP
!
      WRITE(6,6040)
      WRITE(6,6010) 'BLOCK VECTOR Y:'
      WRITE(6,FMT5) (Y(I),I=1,NB*M)
      STOP
!
 6000 FORMAT(/,&
             1X,'***  DAMVJ4  ***',/,/,&
             1X,' **  INPUT  **',/,/,&
             1X,'   * MATRIX DATA FOR CSR FORMAT *')
 6010 FORMAT(/,&
             1X,'     ',A)
 6020 FORMAT(/,&
             1X,'   * ORIGINAL MATRIX *',/)
 6030 FORMAT(/,&
             1X,' **  OUTPUT  **',/,/,&
             1X,'     IERR = ',I4,/)
 6040 FORMAT(/,&
             1X,'   * RESULT Y=BETA*Y+ALPHA*A*X *')
 7000 FORMAT('(1X,5X,',I2,'(3X,I2))')
 7010 FORMAT('(1X,7X,',I2,'(1X,F4.0))')
      END
```

(c) 出力結果

```
***  DAMVJ4  ***

 **  INPUT  **

   * MATRIX DATA FOR CSR FORMAT *

     IA IN BLOCK CSR
         1    3    6    7    9

     JA IN BLOCK CSR:
         1    3
         1    2    3
         3
         3    4

     A IN BLOCK CSR:
          1.   2.   3.   4.   1.   2.   3.   4.
          5.   6.   7.   8.   5.   6.   7.   8.
          9.  10.  11.  12.   9.  10.  11.  12.
         13.  14.  15.  16.  13.  14.  15.  16.
          1.   2.   3.   4.   1.   2.   3.   4.   1.   2.   3.   4.
          5.   6.   7.   8.   5.   6.   7.   8.   5.   6.   7.   8.
          9.  10.  11.  12.   9.  10.  11.  12.   9.  10.  11.  12.
         13.  14.  15.  16.  13.  14.  15.  16.  13.  14.  15.  16.
          1.   2.   3.   4.
          5.   6.   7.   8.
          9.  10.  11.  12.
         13.  14.  15.  16.
          1.   2.   3.   4.   1.   2.   3.   4.
```

```
          5.   6.   7.   8.   5.   6.   7.   8.
          9.  10.  11.  12.   9.  10.  11.  12.
         13.  14.  15.  16.  13.  14.  15.  16.

     ALPHA:
          1.

     BETA:
          1.

     BLOCK VECTOR X:
          1.   1.   1.   1.   2.   2.   2.   2.   3.   3.   3.   3.   4.   4.   4.   4.

     BLOCK VECTOR Y:
          4.   4.   4.   4.   3.   3.   3.   3.   2.   2.   2.   2.   1.   1.   1.   1.

 **  OUTPUT  **

     IERR =    0


   * RESULT Y=BETA*Y+ALPHA*A*X *

     BLOCK VECTOR Y:
         44. 108. 172. 236.  63. 159. 255. 351.  32.  80. 128. 176.  71. 183. 295. 407.
```

3.2.27 ZANVJ1, CANVJ1
複素不規則スパース行列 (JAD 型) の行列ベクトル積 ($\boldsymbol{y}=\beta\boldsymbol{y}+\alpha A\boldsymbol{x}$)

(1) **機 能**

複素不規則スパース行列 A(JAD 型) とベクトル $\boldsymbol{x}$ の積 ($\boldsymbol{y}=\beta\boldsymbol{y}+\alpha A\boldsymbol{x}$) を求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZANVJ1 (AJAD, LXA, IAJAD, JAJAD, JADORD, N, MJAD, ALPHA, BETA, X,
Y, W, IERR)

単精度サブルーチン:

CALL CANVJ1 (AJAD, LXA, IAJAD, JAJAD, JADORD, N, MJAD, ALPHA, BETA, X,
Y, W, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\left\{\begin{array}{l}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{array}\right\}$

項番	引数名	型	大きさ	入出力	内　容
1	AJAD	$\begin{Bmatrix}\mathrm{Z}\\\mathrm{C}\end{Bmatrix}$	LXA	入 力	スパース行列 A(JAD 格納型)(付録 B 参照)
2	LXA	I	1	入 力	配列 AJAD および配列 JAJAD に割り当てる大きさ
3	IAJAD	I	MJAD+1	入 力	スパース行列 A(JAD 格納型) のインデックス配列 (付録 B 参照)
4	JAJAD	I	LXA	入 力	スパース行列 A(JAD 格納型) のインデックス配列 (付録 B 参照)
5	JADORD	I	N	入 力	スパース行列 A(JAD 格納型) のインデックス配列 (付録 B 参照)
6	N	I	1	入 力	ベクトル X 及び Y の次数
7	MJAD	I	1	入 力	行列 A の JAD 格納型における, jagged diagonal の本数
8	ALPHA	$\begin{Bmatrix}\mathrm{Z}\\\mathrm{C}\end{Bmatrix}$	1	入 力	$\boldsymbol{y}=\beta\boldsymbol{y}+\alpha A\boldsymbol{x}$ の乗数 α
9	BETA	$\begin{Bmatrix}\mathrm{Z}\\\mathrm{C}\end{Bmatrix}$	1	入 力	$\boldsymbol{y}=\beta\boldsymbol{y}+\alpha A\boldsymbol{x}$ の乗数 β
10	X	$\begin{Bmatrix}\mathrm{Z}\\\mathrm{C}\end{Bmatrix}$	N	入 力	$\boldsymbol{y}=\beta\boldsymbol{y}+\alpha A\boldsymbol{x}$ のベクトル $\boldsymbol{x}$
11	Y	$\begin{Bmatrix}\mathrm{Z}\\\mathrm{C}\end{Bmatrix}$	N	入出力	$\boldsymbol{y}=\beta\boldsymbol{y}+\alpha A\boldsymbol{x}$ のベクトル $\boldsymbol{y}$
12	W	$\begin{Bmatrix}\mathrm{Z}\\\mathrm{C}\end{Bmatrix}$	N	ワーク	作業配列
13	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a) N > 0

(b) 0 < MJAD ≤ N

(c) IAJAD(MJAD + 1) − IAJAD(1) ≤ LXA

(5) エラーインディケータ

IERR の値	意　　　　　味	処　理　内　容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3100	制限条件 (b) を満足しなかった (N, A, IA, JA の入力値に矛盾がある).	
3200	制限条件 (c) を満足しなかった (配列 AJAD, JAJAD の大きさが不足).	

(6) 注意事項

(a) 本サブルーチンを使用するための行列の格納形式の変換は, なるべく回数を削減した方がよい. 例えば, スパース行列の連立 1 次方程式や固有値方程式の反復解法等で, 行列 A を変更せずに行列ベクトル積を繰り返し計算する場合は, 反復ループの外で 2.2.7 $\left\{\begin{array}{c}\text{ZARSJD}\\ \text{CARSJD}\end{array}\right\}$ や 2.2.8 $\left\{\begin{array}{c}\text{ZARGJM}\\ \text{CARGJM}\end{array}\right\}$ を用いて一度だけ格納形式の変換を行い, 反復ループ内で本サブルーチンを繰り返し使えば効率良く計算できる.

(7) 使用例

(a) 問 題

複素不規則スパース行列 A を (実 1 次元行方向ブロックリスト型) で配列 A に保持し, 内部的に (JAD 格納型) で配列 AJAD に格納してから, 行列ベクトル積 $\boldsymbol{y} = \beta\boldsymbol{y} + \alpha A\boldsymbol{x}$ を求める.

(b) 主プログラム

```
      PROGRAM AANVJ1
! *** EXAMPLE OF AANVJ1 ***
      IMPLICIT NONE
      INTEGER N,M,NZ,ISW,LXA,LXIA
      PARAMETER( N=4, M=1, NZ=8, ISW=0, LXA=NZ, LXIA=N+1 )
      INTEGER IA(N+1),JA(NZ),MJAD,IAJAD(LXIA),JAJAD(LXA),JADORD(N)
      INTEGER IW(N*2+1),IERR
      INTEGER I,J
      COMPLEX(8) A(NZ),AJAD(LXA),X(N),Y(N),W(N)
      COMPLEX(8) ALPHA,BETA
      PARAMETER (ALPHA=(1.0D0,0.D0), BETA=(1.0D0,0.D0))
      CHARACTER*80 FMT1,FMT2,FMT3,FMT4,FMT5,FMT6
!
      READ(5,*) (IA(I),I=1,N+1)
      READ(5,*) (JA(I),I=1,NZ)
      DO 100 I=1,N
         DO 110 J=IA(I),IA(I+1)-1
            A(J) = CMPLX(J,-J, KIND=8)
  110    CONTINUE
  100 CONTINUE
      DO 120 I=1,N
         X(I) = CMPLX(I,-I, KIND=8)
         Y(I) = CMPLX(N - I + 1, -N + I - 1, KIND=8)
  120 CONTINUE
!
      WRITE(6,6000)
      WRITE(6,6010) 'IA IN CSR'
      WRITE(FMT1,7000) N+1
      WRITE(6,FMT1) (IA(I),I=1,N+1)
      WRITE(6,6010) 'JA IN CSR:'
      DO 130 I=1,N
         WRITE(FMT2,7000) IA(I+1) - IA(I)
         WRITE(6,FMT2) (JA(J),J=IA(I),IA(I+1)-1)
  130 CONTINUE
      WRITE(6,6010) 'A IN CSR:'
      DO 140 I=1,N
```

```
         WRITE(FMT3,7010) IA(I+1) - IA(I)
         WRITE(6,FMT3)( A(J), J=IA(I),IA(I+1)-1 )
  140 CONTINUE
!
      WRITE(FMT4,7010) 1
      WRITE(6,6010) 'ALPHA:'
      WRITE(6,FMT4) ALPHA
      WRITE(6,6010) 'BETA:'
      WRITE(6,FMT4) BETA
      WRITE(6,6010) 'VECTOR X:'
      WRITE(FMT5,7010) N
      WRITE(6,FMT5) (X(I),I=1,N)
      WRITE(6,6010) 'VECTOR Y:'
      WRITE(6,FMT5) (Y(I),I=1,N)
!
!     CONVERT FROM CSR TO JAD
!
      CALL ZARGJM&
      (N,M,A,IA,JA,ISW,LXA,LXIA,MJAD,AJAD,IAJAD,JAJAD,JADORD,IW,IERR)
!
!     MATRIX-VECTOR PRODUCT  Y=BETA*Y+ALPHA*A*X
!
      CALL ZANVJ1&
      (AJAD,LXA,IAJAD,JAJAD,JADORD,N,MJAD,ALPHA,BETA,X,Y,W,IERR)
!
      WRITE(6,6030) IERR
      IF( IERR .GE. 3000 ) STOP
!
      WRITE(6,6040)
      WRITE(FMT6,7020) N
      WRITE(6,FMT6) (Y(I),I=1,N)
      STOP
!
 6000 FORMAT(/,&
             1X,'***  ZANVJ1  ***',/,/,&
             1X,' **  INPUT  **',/,/,&
             1X,'   * MATRIX DATA FOR CSR FORMAT *')
 6010 FORMAT(/,&
             1X,'     ',A)
 6020 FORMAT(/,&
             1X,'   * ORIGINAL MATRIX *',/)
 6030 FORMAT(/,&
             1X,' **  OUTPUT  **',/,/,&
             1X,'     IERR = ',I4,/)
 6040 FORMAT(/,&
             1X,'   * RESULT Y=BETA*Y+ALPHA*A*X *')
 7000 FORMAT('(1X,5X,',I2,'(3X,I2))')
 7010 FORMAT('(1X,7X,',I2,'(" (",F4.1,",",F4.1,")"))')
 7020 FORMAT('(1X,7X,',I2,'(" (",F4.1,",",F6.1,")"))')
      END
```

(c) 出力結果

```
 ***  ZANVJ1  ***

  **  INPUT  **

    * MATRIX DATA FOR CSR FORMAT *

      IA IN CSR
          1    3    6    7    9

      JA IN CSR:
          1    3
          1    2    3
          3
          3    4

      A IN CSR:
         ( 1.0,-1.0) ( 2.0,-2.0)
         ( 3.0,-3.0) ( 4.0,-4.0) ( 5.0,-5.0)
         ( 6.0,-6.0)
         ( 7.0,-7.0) ( 8.0,-8.0)

      ALPHA:
         ( 1.0, 0.0)

      BETA:
         ( 1.0, 0.0)

      VECTOR X:
         ( 1.0,-1.0) ( 2.0,-2.0) ( 3.0,-3.0) ( 4.0,-4.0)

      VECTOR Y:
         ( 4.0,-4.0) ( 3.0,-3.0) ( 2.0,-2.0) ( 1.0,-1.0)

  **  OUTPUT  **

      IERR =    0


    * RESULT Y=BETA*Y+ALPHA*A*X *
         ( 4.0, -18.0) ( 3.0, -55.0) ( 2.0, -38.0) ( 1.0,-107.0)
```

第 4 章　固有値・固有ベクトル

4.1　概 要

本章では, 行列の固有値および固有ベクトルを求めるサブルーチンについて説明する.

標準固有値問題では与えられた行列 A について

$$A\boldsymbol{x} = \lambda\boldsymbol{x}$$

を満たす値 λ およびベクトル $\boldsymbol{x}$ を求める. この λ は行列 A の固有値, $\boldsymbol{x}$ はこれに対応する固有ベクトルと呼ばれる.

また, 一般化固有値問題では与えられた行列 A ならびに B について,

$A\boldsymbol{x} = \lambda B\boldsymbol{x}$,
$AB\boldsymbol{x} = \lambda\boldsymbol{x}$ (A と B は両方ともエルミート行列, B は正定値),
$BA\boldsymbol{x} = \lambda\boldsymbol{x}$ (A と B は両方ともエルミート行列, B は正定値)

のいずれかを満たす値 λ およびベクトル $\boldsymbol{x}$ を求める. これらの λ, $\boldsymbol{x}$ も同様に固有値, 固有ベクトルと呼ばれる. 行列 A, B がエルミート行列で, かつ B が正定値の場合, 固有値は全て実数であり, 異なる固有値に対する固有ベクトルは互いに直交する.

固有値問題を解くために行列の種類に応じて, 種々の手法が提案されている. 本ライブラリでは, 通常

(1) 入力行列を実対称 3 重対角行列またはヘッセンベルグ行列に変換する.

(2) 実対称 3 重対角行列またはヘッセンベルグ行列の固有値および固有ベクトルを求める.

(3) 求めた固有ベクトルをもとの入力行列の固有ベクトルに変換する.

という 3 段階を経て固有値問題を解く.

本章に属するサブルーチンでは, つぎの 6 つのカテゴリーに対応する機能を用意している.

全固有値・全固有ベクトル
すべての固有値と対応する固有ベクトルを求める.

全固有値
固有値のみをすべて求める.

固有値・固有ベクトル
固有値を大きい方から数個, または小さい方から数個求め, 対応する固有ベクトルを求める.

固有値
固有値を大きい方から数個, または小さい方から数個求める.

固有値・固有ベクトル (区間指定)
区間指定された固有値を大きい方から数個, または小さい方から数個求め, 対応する固有ベクトルを求める.

固有値 (区間指定)
区間指定された固有値を大きい方から数個, または小さい方から数個求める.

ただし, 非対称行列に対応する機能の場合は「全固有値・全固有ベクトル」と「全固有値」に対応する機能のみ提供している.

4.1.1 使用上の注意

(1) 利用者は行列の種類により適切なサブルーチン群 (たとえば, 4.2 実行列, 4.9 実対称バンド行列など) を選び, 次に, 使用目的, 処理時間, 必要メモリ量などから最適のサブルーチンを選べばよい.

(2) 一般に, 「全固有値・全固有ベクトル」または「固有値・固有ベクトル」に対応する機能はそれぞれ「全固有値」または「固有値」に対応する機能より多くの処理時間, メモリ量を必要とする.

(3) 一般に, 「固有値・固有ベクトル」および「固有値」に対応する機能を使用して効果があるのは, 求める固有値の個数が, たかだか全体の 2 割程度までの場合で, それ以上求めたい場合は, 「全固有値・全固有ベクトル」または「全固有値」に対応する機能を使用したほうが計算時間は少なくてすむ.

(4) 本ライブラリにおける一般化固有値問題のサブルーチンでは, 行列 B が正定値であるという条件が課せられている. しかし, 以下のような場合は, 行列 B が正定値でなくても固有値・固有ベクトルを求めることができる.

    (a) 行列 B が正定値でないが, 行列 A は正定値の場合

    $$Bv = \lambda^{-1} Av$$

    によって固有値 $\lambda(\neq 0)$ と固有ベクトルが得られる.

    (b) A, B が共に正定値でないが, $A + B$ は正定値の場合

    $$Av = \frac{\lambda}{1+\lambda}(A+B)v$$

    によって固有値 $\lambda(\neq -1)$ と固有ベクトルが得られる.

(5) 入力行列が実対称行列またはエルミート行列である場合, 専用のサブルーチンを使用したほうが計算時間は少なくてすむ.

(6) 行列構造について
2 次元有限差分法, 3 次元有限差分法の陰解法等で生ずる規則スパース行列の固有値方程式を解く場合, 規則スパース行列用サブルーチンを用いる. それ以外の差分近似及び有限要素近似で生ずる不規則スパース行列の固有値方程式を解く場合, 不規則スパース行列用サブルーチンを用いる (行列のデータ格納方法は, 付録 B 参照).

4.1.2　使用しているアルゴリズム

4.1.2.1　実行列のヘッセンベルグ (Hessenberg) 行列への変換

基本相似変換によって, $n \times n$ 実行列 A をヘッセンベルグ行列 $H = (h_{ij}) : h_{ij} = 0 (i > j+1)$ に変換する．すなわち,

$$A_1 = A$$

として,

$$A_{k+1} = P^{-1}_{k+1} I_{k+1,(k+1)'} A_k I_{k+1,(k+1)'} P_{k+1}$$

の変換を全体として $n-2$ 回反復することによって得られる．ここで, $I_{k+1,(k+1)'}, P_{k+1}$ はそれぞれ次の (1), (2) で決定される置換行列, 相似変換行列である．なお, A_k ははじめの $k-1$ 列目までがヘッセンベルグ形になる．

$$A_k = (a^{(k)}_{ij})$$

とおくと,

(1) 第 k 列より,

$$|a^{(k)}_{(k+1)',k}| = \max_{i=k+1,\cdots,n} |a^{(k)}_{ik}|$$

を見つけて, $(k+1)'$ 行と $(k+1)$ 行, $(k+1)'$ 列と $(k+1)$ 列を入れ替える．もしこの値が 0 ならば, 行列は二つの小行列に分解され, この二つの小行列の固有値問題を解けばよい．

```
(2) for $i = k+2, n$
        $p^{(k+1)}_{i,k+1} \leftarrow \frac{a^{(k)}_{ik}}{a^{(k)}_{k+1,k}}$
        $(i$ 行$) \leftarrow (i$ 行$) - ((k+1)$ 行$) \times p^{(k+1)}_{i,k+1}$
        $((k+1)$ 列$) \leftarrow ((k+1)$ 列$) + (i$ 列$) \times p^{(k+1)}_{i,k+1}$
```

$$\begin{cases} (P_{k+1})_{i,k+1} = p^{(k+1)}_{i,k+1} & (i = k+2, \cdots, n) \\ (P_{k+1})_{ij} = \delta_{ij} & (\text{その他の } i, j) \\ & (\delta_{ij}\text{はクロネッカーのデルタ}) \end{cases}$$

注　一般には, 変換行列の累計を残しておけば, 固有ベクトルを求めることができる．つまり $m \times m$ 行列 A に正則行列による変換を繰り返し適用して,

$$Q^{-1}_m \cdots Q^{-1}_2 Q^{-1}_1 A Q_1 Q_2 \cdots Q_m = \Lambda \quad (\text{ただし}, \Lambda\text{は対角行列})$$

となったならば, Λ の対角成分に固有値が並び, 変換行列の積 $(Q_1 Q_2 \cdots Q_m)$ の各列ベクトルが固有ベクトルとなる．

4.1.2.2　複素行列のヘッセンベルグ行列への変換

ハウスホルダー (Householder) 法によって, $n \times n$ 複素行列 A をヘッセンベルグ行列 $H = (h_{ij})$ に変換する．すなわち, $A_1 = A$ として, $k = 1, 2, \cdots, n-2$ に対して, あるベクトル $\boldsymbol{u}_k$ を

$$H_k = \frac{1}{2} \boldsymbol{u}^*_k \boldsymbol{u}_k$$

$$P_k = I - \frac{\boldsymbol{u}_k \boldsymbol{u}^*_k}{H_k}$$

かつ,

$$A_{k+1} = P_k A_k P_k$$

の第 k 列の副対角成分より下がすべて 0 になるようにとれる. A_{n-1} が求めるヘッセンベルグ行列となる. なお, 変換行列 P_k はユニタリかつエルミート行列である.

4.1.2.3　実行列, 複素行列の平衡化

実行列と複素行列に対しては, ヘッセンベルグ行列に変換する前に平衡化を行う. その際, まず行と列を適当入れ替える置換 P をもとの行列 A に左と右から掛けて,

$$PAP = \begin{bmatrix} U & X & Y \\ O & B & Z \\ O & O & V \end{bmatrix}$$

とする. U, V は自明な孤立した固有値を対角成分にもつ上三角行列であり, B はこれ以上孤立した固有値を含まない正方行列である.
次に, $B_1 = B$ として正則な対角行列 D_k により, 相似変換

$$B_{k+1} = D_k^{-1} B_k D_k$$

を繰り返すことによって, B の互いに対応する行と列の絶対和がほぼ等しくなるようにする.
最終的には

$$\begin{bmatrix} U & XD & Y \\ O & D^{-1}BD & D^{-1}Z \\ O & O & V \end{bmatrix}$$

の形に変換される.

4.1.2.4　QR 法, ダブル QR 法

正則な複素ヘッセンベルグ行列 H に対して, ユニタリ行列 Q と上三角行列 R(対角成分がすべて正になる) があって

$$H = QR$$

と一意に分解される.

$$H_1 = H$$

とおき, H_k を

$$H_k = Q_k R_k$$

と分解して, それを逆順にかけて

$$H_{k+1} = R_k Q_k = Q_k^* H_k Q_k \quad (Q_k^* \text{は} \ Q_k \text{の随伴行列})$$

として, $H_1, H_2, \cdots, H_{k-1}, H_k$ を作ると, これらはすべてヘッセンベルグ行列であり, $k \to \infty$ とすると H_k は上三角行列に収束し, その対角成分には H の固有値が並ぶ.
実際の QR 法では, 収束を加速するために, H_k の代わりに原点移動を行った $H_k - \mu_k I$ を作って,

$$H_k - \mu_k I = Q_k R_k$$

と分解し, (μ_k は固有値の近似値としている．)

$$H_{k+1} = R_k Q_k + \mu_k I$$

を作ると,

$$H_{k+1} = Q_k^* H_k Q_k$$

となる．この操作を繰り返して収束させた後に, 原点移動量の累計で補正したものが固有値となる．

ダブル QR 法

実行列 (非対称) に対して上記の原点移動を行うと, 途中で複素行列が現れることがあるので, それを避けるために,

$$H_{k+2} = (Q_k Q_{k+1})^T H_k (Q_k Q_{k+1})$$

$$(H_k - \mu_k I)(H_k - \mu_{k+1} I) = (Q_k Q_{k+1})(R_{k+1} R_k)$$

とする．もし μ_k と μ_{k+1} が共に実または共役複素なら上の第 2 式の左辺は実行列となる．
実際にはハウスホルダー変換行列 P_i を用いて

$$Q_k Q_{k+1} = P_1 P_2 \cdots\cdots P_{n-1}$$

とし,

$$H_{k+2} = P_{n-1}^T \cdots\cdots P_2^T P_1^T H_k P_1 P_2 \cdots\cdots P_{n-1}$$

となる．
詳細は, 参考文献 (1), (2), (5) 等を参照されたい．

4.1.2.5　実対称行列の実対称 3 重対角行列への変換

ハウスホルダー法によって, $n \times n$ 実対称行列 A を実対称 3 重対角行列 T に変換する．
すなわち, $A_1 = A$ として, $k = 1, 2, \cdots, n-2$ に対して, あるベクトル $\boldsymbol{u}_k$ を

$$H_k = \frac{1}{2} \boldsymbol{u}_k^T \boldsymbol{u}_k$$

$$P_k = I - \frac{\boldsymbol{u}_k \boldsymbol{u}_k^T}{H_k}$$

かつ,

$$A_{k+1} = P_k A_k P_k$$

の第 k 列の副対角成分より下がすべて 0 になるように取れる．A_{n-1} が求める実対称 3 重対角行列となる．なお, 変換行列 P_k は直交かつ対称な行列である．

4.1.2.6　エルミート (Hermitian) 行列の実対称 3 重対角行列への変換

まず, ハウスホルダー法によって, $n \times n$ エルミート行列 A をエルミート 3 重対角行列 S に変換する．

$$S = P_{n-2} \cdots P_2 P_1 A P_1 P_2 \cdots P_{n-2}$$

さらに, 正則な複素対角行列 D によって (相似変換), 実対称 3 重対角行列 T に変換する．

$$T = D^* S D$$

4.1.2.7　ブロックアルゴリズムによるハウスホルダー変換

ハウスホルダー変換には, ブロックアルゴリズムを用いる．この手法は, もとの実対称行列を実対称 3 重対角行列に変換する際, 行列の積和によるランク-1 の行列更新を単純化する．

対称行列 A について k 回相似変換した行列を A_{k+1} とすると,

$$A_{k+1} = P_k A_k P_k = A_k - \boldsymbol{u}_k \boldsymbol{v}_k^T - \boldsymbol{v}_k \boldsymbol{u}_k^T$$

となる．ここで, P_k は, 直交行列である．ただし,

$$\begin{aligned}
&A_1 = A \\
&P_k = I - \frac{\boldsymbol{u}_k \boldsymbol{u}_k^T}{H_k} \\
&H_k = \frac{1}{2}\boldsymbol{u}_k^T \boldsymbol{u}_k \\
&\boldsymbol{y}_k = A_k \boldsymbol{u}_k \\
&\boldsymbol{v}_k = \frac{(\boldsymbol{y}_k - \frac{(\boldsymbol{u}_k^T \boldsymbol{y}_k)\boldsymbol{u}_k}{2H_k})}{H_k}
\end{aligned}$$

ハウスホルダー変換では, 1 回の相似変換に対し, 2 段の行列更新を行う．この行列 A_{k+1} は, A_k を直接使わずに表すことができる．

$$A_{k+1} = A_{k-1} - \boldsymbol{u}_{k-1}\boldsymbol{v}_{k-1}{}^T - \boldsymbol{v}_{k-1}\boldsymbol{u}_{k-1}{}^T - \boldsymbol{u}_k\boldsymbol{v}_k{}^T - \boldsymbol{v}_k\boldsymbol{u}_k{}^T$$

同様の操作によって, p 回の鏡像変換を行って得られる行列 A_{p+1} は,

$$A_{p+1} = A_1 - \sum_{i=1}^{p}(\boldsymbol{u}_i \boldsymbol{v}_i^T + \boldsymbol{v}_i \boldsymbol{u}_i^T)$$

となるので, A_1 に $2p$ 段の行列更新をまとめて行うことにより A_{p+1} が高速に求まる．なお, 行列 A がエルミート行列の場合, 転置記号 ^{T} はエルミート共役記号 * に読み替える．詳細は, 参考文献 (6)(7) を参照されたい．

4.1.2.8　実対称バンド行列の実対称 3 重対角行列への変換

ギブンス (Givens) 法によって $n \times n$ 実対称バンド行列 A (バンド幅 m) を実対称 3 重対角行列 T に変換する．
すなわち, 実対称バンド行列を A として, 変換行列 P を

$$P = \begin{array}{ccccccccccc|l}
 & & & i\,\text{列目} & & & & j\,\text{列目} & & & & \\
\hline
1 & & & \vdots & & & & \vdots & & & & \\
 & \ddots & & \vdots & & & & \vdots & & 0 & & \\
 & & 1 & \vdots & & & & \vdots & & & & \\
\cdots & \cdots & \cdots & \cos\theta & \cdots & \cdots & \cdots & \sin\theta & \cdots & \cdots & \cdots & i\,\text{行目} \\
 & & & \vdots & 1 & & & \vdots & & & & \\
 & & & \vdots & & \ddots & & \vdots & & & & \\
 & & & \vdots & & & 1 & \vdots & & & & \\
\cdots & \cdots & \cdots & -\sin\theta & \cdots & \cdots & \cdots & \cos\theta & \cdots & \cdots & \cdots & j\,\text{行目} \\
 & & & \vdots & & & & \vdots & 1 & & & \\
0 & & & \vdots & & & & \vdots & & \ddots & & \\
 & & & \vdots & & & & \vdots & & & 1 &
\end{array}$$

$$\cos\theta = \frac{a_{j,j-1}}{\sqrt{a_{j,j+1}^2 + a_{i,j-1}^2}}$$

$$\sin\theta = \frac{a_{i,j-1}}{\sqrt{a_{j,j-1}^2 + a_{i,j-1}^2}}$$

とすると,

$$A' = P^T AP$$

の変換によって $a_{i,j-1}$ を 0 にすることができる. ($a_{i,j}$ は行列 A の (i,j) 成分)
この変換を

$$j = 2, \cdots\cdots\cdots\cdots, n-1$$

$$i = j+1, \cdots\cdots, \mathrm{MIN}(m+j-1, n)$$

に対して用いる実対称バンド行列 A を実対称 3 重対角行列 T にすることができる. このとき, 変換の回数は $(m-1)(n-\frac{m}{2}-1)$ 回である. なお, 変換行列 P は直交行列である.

4.1.2.9 QR 法

3 重対角行列 T はユニタリ行列 Q と上三角行列 R(対角成分がすべて正になる) に

$$T = QR$$

と一意に分解される.

$$T_k = T$$

とおくと, T_k は

$$T_k \Rightarrow Q_k R_k$$

と分解される. それを逆順に掛けて,

$$T_{k+1} \Leftarrow R_k Q_k = Q_k^* T_k Q_k \quad (Q_k^*\text{は } Q_k\text{の随伴行列}) \quad (k = 1, \cdots)$$

とすると, $T_1, T_2, \cdots, T_k, T_{k+1}$ はすべて 3 重対角行列であり, $k \to \infty$ とすると T_k は対角行列に収束し, その対角成分には T の固有値が並ぶ. 反復計算は非対角成分が十分小さくなったとき, 収束したと判断し, 反復を打ち切る.
実際の QR 法では, 収束を加速するために, μ_k を固有値の近似値として, T_k の代りに原点移動を行った $T_k - \mu_k I$ を作り,

$$T_k - \mu_k I = Q_k R_k$$

と分解する.
固有値の近似値の算定方法としては, 隣接固有値 (または絶対値の近い固有値) がある場合を考えて, 右下すみ小行列の固有値を無平方根 QR 法で求めて μ_k とする.
これから,

$$T_{k+1} = R_k Q_k + \mu_k I$$

を作ると,

$$T_{k+1} = Q_k^* T_k Q_k$$

となる. この操作を繰り返して収束させた後に, 原点移動量で補正したものが固有値となる.
3 重対角行列の変換する前の, もとの行列の固有ベクトルは, ハウスホルダー変換により 3 重対角行列 T を求めた際の変換行列を順に掛けてゆき, さらに QR 法によって得られた変換行列 $Q_1, Q_2, \cdots, Q_k$ を掛けあわせれば得られる.
その行列の第 i 列は, 第 i 番目の固有値に対応する固有ベクトルに収束する.

4.1.2.10　無平方根 QR 法

実対称 3 重行列の固有値のみを求める場合は, QR 法の平方根計算を省いた無平方 QR 法が高速である.
対角要素を $\alpha_1, \cdots, \alpha_n$, 副対角要素を $\beta_1, \cdots, \beta_{n-1}$ とする. 計算中の変換行列の一成分を $P^{(i)}$ として $P^{(i)}$ の中の $\sin\theta$ と $\cos\theta$ を S_i, C_i とする.
QR 法では,

$$P_i = \alpha_i C_{i-1} - \beta_{i-1} S_{i-1} C_{i-2}$$

$$S_i = \frac{\beta_i}{\sqrt{P_i^2 + \beta_i^2}}$$

$$C_i = \frac{P_i}{\sqrt{P_i^2 + \beta_i^2}}$$

$$\text{新}\alpha_{i-1} = \alpha_i + P_{i-1} C_{i-2} - P_i C_{i-1}$$

$$\text{新}\beta_{i-2} = S_{i-2}\sqrt{P_{i-1}^2 + \beta_{i-1}^2}$$

と平方根の計算をしなければならないが, これを $P_i \beta_i S_i C_i$ 全て 2 乗の形で計算すると,

$$C_0 = 1, S_0 = 0, \gamma_1 = \alpha_1, P_1^2 = \alpha_1^2, \alpha_{n+1} = \beta_{n+1} = 0$$

とし

$$t_i^2 = P_i^2 + \beta_{i+1}^2$$

$$\text{新}\beta_i^2 = S_{i-1}^2 t_i^2$$

$$S_i^2 = \frac{\beta_{i+1}^2}{t_i^2}, C_i^2 = \frac{P_i^2}{t_i^2}$$

$$P_{i+1}^2 = \alpha_{i+1}^2 C_i^2 - 2\alpha_i + S_i^2 \gamma_i + \beta_{i+1}^2 S_i^2 C_{i-1}^2$$

$$\gamma_{i+1} = \alpha_{i+1} C_i^2 = S_i^2 \gamma_i$$

$$\text{新}\alpha_i = \alpha_{i+1} + \gamma_i - \gamma_{i+1}$$

となり, 平方根なしで計算できる.

4.1.2.11　バイセクション法

まず, 実対称 3 重対角行列 T の固有値をバイセクション法により, 大きい方から数個, または小さい方から数個求める場合を述べる.
T の対角成分を d_1, d_2, $\cdots$, d_n, 副対角成分を s_1, s_2, $\cdots$, s_{n-1} とし, λ を変数として,

$$f_0(\lambda) = 1$$

$$f_1(\lambda) = d_1 - \lambda$$

$$f_i(\lambda) = (d_1 - \lambda) f_{i-1}(\lambda) - s_{i-1}^2 f_{i-2}(\lambda) (i = 2, \cdots, n)$$

という関数列を作ると, $f_0(\lambda)$, $f_1(\lambda)$, $\cdots$, $f_m(\lambda)$ はスツルム (Sturm) 列をなす. つまり, ある λ に対する引き続く関数列の符号の不一致の個数を $L(\lambda)$ とすると, この $L(\lambda)$ は λ より小さな固有値の個数に等しい.
オーバフロー, アンダフローを防ぐため, 実際は $g_i(\lambda)$ を

$$g_i(\lambda) = \frac{f_i(\lambda)}{f_{i-1}(\lambda)} \ (i = 1, 2, \cdots, n)$$

で定義すれば, 負になる $g_i(\lambda)$ の個数が $L(\lambda)$ となる. なお $g_i(\lambda)$ は,

$$g_1(\lambda) = d_1 - \lambda$$

$$g_i(\lambda) = (d_i - \lambda) - \frac{s_{i-1}^2}{g_{i-1}(\lambda)} (i = 2, \cdots, n)$$

を満足する．もし, $g_{i-1}(\lambda) = 0$ となったときは

$$g_i(\lambda) = (d_i - \lambda) - \frac{|s_{i-1}|}{\varepsilon} (\varepsilon：\text{誤差判定のための単位})$$

とする．

T の固有値を $\lambda_1 \leq \lambda_2 \leq \cdots\cdots \leq \lambda_m$ とする．**ゲルシュゴリン (Gerschgorin) の定理**より, 固有値全体の下限 (x_{min}) および上限 (x_{max}) は,

$$x_{max} = \max_{1 \leq i \leq n} (d_i + (| s_{i-1} | + | s_i |))$$

$$x_{min} = \min_{1 \leq i \leq n} (d_i - (| s_{i-1} | + | s_i |))$$

で与えられる．ただし, $s_0 = s_n = 0$ とする．

この x_{min}, x_{max} をもとにして, 上記のように固有値の個数を数えながら区間分割を繰り返して, 固有値の存在範囲を小さくしていく．そうすれば微小区間の両端をある固有値に収束させせることができる．

この方法により, 多くの固有値を求める場合, 高速に処理可能となる．

スツルム関数列, ゲルシュゴリンの定理については, 参考文献 (2), (8) を参照されたい．

4.1.2.12 ブロックアルゴリズムによる相似 (ユニタリ) 変換の累積

QR 法および逆反復法を用いて実対称行列の固有ベクトルを求める場合, ハウスホルダー変換の際に用いた相似 (ユニタリ) 変換行列の累積が必要になる．この累積計算を高速に行うには, ブロックアルゴリズムの適用が非常に有効である．

ハウスホルダー変換により実対称行列から 3 重対角行列を求めた際の変換行列 P_k を

$$P_k = \boldsymbol{I} - \frac{\boldsymbol{u}_k \boldsymbol{u}_k^T}{H_k}$$

$$H_k = \frac{1}{2} \boldsymbol{u}_k^T \boldsymbol{u}_k$$

とする．この変換行列 P_k の累積は, 次式のようになる．

$$P_1 P_2 \cdots P_{n-2} = \boldsymbol{I} - \Sigma_{i=1}^{n-2} \boldsymbol{u}_i \boldsymbol{w}_i^T$$

ただし, $\boldsymbol{w}_i^T$ は, 以下の漸化式によって表される．

$$\boldsymbol{w}_{n-2}^T = \frac{\boldsymbol{u}_{n-2}^T}{H_{n-2}}$$

$$\boldsymbol{w}_i^T = \frac{\boldsymbol{u}_i^T - \displaystyle\sum_{j=i-1}^{n-2} (\boldsymbol{u}_i^T \boldsymbol{u}_j) \boldsymbol{w}_j^T}{H_i}$$

QR 法または逆反復法を用いて得られた実対称 3 重対角行列の固有ベクトルを $\boldsymbol{V}$ とする．もとの行列の固有ベクトル $\boldsymbol{X}$ は次式により得られる．

$$\boldsymbol{X} = P_1 \cdots P_{n-2} \boldsymbol{V}$$

$$= \boldsymbol{V} - \sum_{i=1}^{n-2} \boldsymbol{u}_i \boldsymbol{w}_i^T \boldsymbol{V}$$

相似 (ユニタリ) 変換行列 P_k と固有ベクトル $\boldsymbol{V}$ の積は, ランク-1 の行列更新である．したがって, 変換行列の累積は, 行列更新をまとめて行うことによって高速に計算できる．なお, もとの行列がエルミート行列の場合, 転置記号 ^{T} はエルミート共役記号 * に読み替える．

4.1.2.13　逆反復法

無平方根 QR 法またはバイセクション法で求まった固有値に対応する固有ベクトルを逆反復法で求める．
実対称 3 重対角行列 T のある固有値 λ_k の近似値 μ_k が求まったとする．このとき初期ベクトル $\boldsymbol{v}_0$ を適当にとって連立 1 次方程式,

$$(T - \mu_k I)\boldsymbol{v}_i = \boldsymbol{v}_{i-1} \quad (i = 1, 2, \cdots)$$

を繰り返し反復的に解いて, $\boldsymbol{v}_i$ が収束条件を満足したならそれを固有ベクトルとする．
連立 1 次方程式を解くのには, 部分軸選択を伴うガウス法による LU 分解を行い, 前進代入, 後退代入を用いて解く．

4.1.2.14　一般化固有値問題

エルミート行列の一般化固有値問題

$$A\boldsymbol{x} = \lambda B\boldsymbol{x} \quad (A : \text{エルミート}, B : \text{正値エルミート})$$

で, B をコレスキー分解して,

$$B = LL^*$$

とし,

$$(L^{-1}A(L^*)^{-1})(L^*\boldsymbol{x}) = \lambda(L^*\boldsymbol{x})$$

とする．

$$P = L^{-1}A(L^*)^{-1}$$

$$L^*\boldsymbol{x} = \boldsymbol{y}$$

とおけば, P はエルミート行列で

$$P\boldsymbol{y} = \lambda\boldsymbol{y}$$

という標準の固有値問題に変換される．行列 A の固有ベクトルは,

$$\boldsymbol{x} = (L^*)^{-1}\boldsymbol{y}$$

である．
さらに, エルミート行列に対する一般化固有値問題 $A\boldsymbol{x} = \lambda B\boldsymbol{x}(B$: 正定値) 以外の一般化固有値問題を, 正定値エルミート行列 B の位置で分けて

$$AB\boldsymbol{x} = \lambda\boldsymbol{x}$$

と

$$BA\boldsymbol{x} = \lambda\boldsymbol{x}$$

に分類する．これらを以下の手順で標準固有値問題に帰着し, その固有値 λ と固有ベクトル $\boldsymbol{x}$ を求める．エルミート行列の標準固有値問題に帰着させるために以下のような手順をとる．

① 正定値行列 B をコレスキー分解し, $B = L^*L(L$ は下三角行列) とする．

② $AB\boldsymbol{x} = \lambda\boldsymbol{x}$ は $C = LAL^*$ の固有値問題に帰着し, 固有ベクトルは L の逆行列を掛けることで得られる．

③ $BA\boldsymbol{x} = \lambda\boldsymbol{x}$ は $C = LAL^*$ の固有値問題に帰着し, 固有ベクトルは L^* を掛けることで得られる．

4.1.2.15 QZ 法, コンビネーションシフト QZ 法

$$Ax = \lambda Bx \left(\begin{array}{l} A: \text{ヘッセンベルグ行列} \\ B: \text{上三角行列, 正則} \end{array} \right)$$

に対し,
$C = AB^{-1}$ と置くと, C もまたヘッセンベルグ行列である.
従って, QR 法より,

$$C_1 = C$$

$$C_k = Q_k R_k \quad (Q_k: \text{ユニタリ行列}, R_k: \text{上三角行列})$$

$$C_{k+1} = R_k Q_k = Q_k^* C_k Q_k$$

として $C_1, C_2, \cdots, C_k, C_{k+1}, \cdots$ を作ると, これらはヘッセンベルグ行列であり, $k \to \infty$ とすると上三角行列に収束する.
ところで, ある Z(ユニタリ行列) を選び,

$$\begin{array}{l} A_{k+1} = Q_k A_k Z_k \\ B_{k+1} = Q_k B_k Z_k \end{array} \left(\begin{array}{l} A_{k+1}: \text{ヘッセンベルグ行列} \\ B_{k+1}: \text{上三角行列} \end{array} \right)$$

となるようにできる.
この時

$$A_{k+1}(B_{k+1})^{-1} = Q_k A_k Z_k Z_k^T B_k^{-1} Q_k^T = Q_k A_k B_k^{-1} Q_k^T = C_{k+1}$$

であるので, 上記の収束は A_k が上三角行列に収束していくことであり, 固有値は A_∞ の対角要素を α_i, B の対角要素を β_i とすると, α_i/β_i で表される.

コンビネーションシフト QZ 法

QR 法, ダブル QR 法における原点移動と同様に, QZ 法においても加速法として原点移動を行うことができる. コンビネーションシフト QZ 法は, QR 法タイプの原点移動とダブル QR 法タイプの原点移動を組み合わせて行うものである. 詳細は参考文献 (12), (13) 等を参照されたい.

4.1.2.16 サブスペース法

$$Ax = \lambda Bx \quad (A: \text{実対称行列}, B: \text{正値対称行列})$$

に対し, 出発ベクトル群

$$X_0 = (x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, x_4^{(0)} \cdots, \cdots, x_m^{(0)}) \; (x_i^{(0)}, x_j^{(0)} \text{は独立})$$

を定める.

$$Y_k = BX_k$$

$$AX_{k+1} = Y_k \quad (k \to \infty)$$

とすると, $x_i^{(k)}$ の張る空間 E_k は絶対値最小から m 個の固有値 $\lambda_i (i = 1, 2, \cdots, m)$ に対応する固有ベクトル $\phi_i (i = 1, 2, \cdots, m)$ の張る空間 E_∞ に収束する (ただし, $x_i^{(0)}$ は E_∞ と直交していないとする).
ここで, ベクトルの収束を速めるため, $AZ_k = Y_k$ と置き, 行列 A と B の $Z_i^{(k)}$ の張る空間上への射影を用いる.

$$A_k = Z_k^T A Z_k \quad (Z_k = (Z_1^{(k)}, Z_2^{(k)}, \cdots, Z_m^{(k)}))$$

$$B_k = Z_k^T B Z_k$$

A_k, B_k の固有値, 固有ベクトルを求め, Z_k を改善し, それを X_{k+1} とすると, X_{k+1} はより速く求める固有ベクトル $\boldsymbol{\phi} = (\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \cdots, \boldsymbol{\phi}_m)$ に収束する.

$$A_k Q_k = B_k Q_k \Lambda_k \left(\begin{array}{l} \Lambda_k : A_k, B_k\text{に対応する固有値を対角要素とする対角行列} \\ Q_k : A_k, B_k\text{の固有ベクトルを列ベクトルに持つ行列} \end{array} \right)$$

$$X_{k+1} = Z_k Q_k$$

$$X_{k+1} \rightarrow \Phi \quad (k \rightarrow \infty) \quad \Phi = (\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \cdots, \boldsymbol{\phi}_m)$$

$$\Lambda_k \rightarrow \Lambda \quad (k \rightarrow \infty) \quad \Lambda = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_m \end{bmatrix}$$

また, 絶対値最大の固有値から求める場合は,

$$Y_k = A X_k$$

$$B Z_k = Y_k$$

$$X_{k+1} = Z_k Q_k$$

とする.
実際の計算においては, 反復途中でベクトルが互いに平行に近くなったり, ノルムが増大するのを防ぐために, 1 回の反復ごとにベクトルを正規直交化する.
また, 収束速度は $|\lambda_i/\lambda_m|$ に比例するため, 実際に求めたい固有値数よりも多くのベクトルを, 反復処理に使用している. 詳細は, 文献 (14), (15) を参照されたい.

4.1.2.17 スツルム列チェック

$A\boldsymbol{x} = \lambda B\boldsymbol{x}$ という一般化固有値問題において, $(A - \lambda_m B)$ を $\mathrm{LDL^T}$ 分解した時の対角要素に現れる負の要素の個数を n 個とすると, n は λ_m より小さい固有値の数に相当する (ただし, 固有値はすべて正であるものとする.) 詳細は文献 (14) を参照されたい.

4.1.2.18 Jacobi-Davidson 法

規模の大きいスパース実対称 (あるいはエルミート) 行列の固有値問題を数値的に解くのに, Davidson (文献 (25)) によって提案された手法の変種が適用されることがしばしばある. これらの解法では, 問題の行列 A の近似逆行列を用いて逐次構成される部分空間の列が利用される. 行列 A は $A = A^H$ または $A^* = A^H$ を満たすものとする. ここで, A^* は行列 A と複素共役な行列を表し, $A^H = (A^T)^*$ (転置後に複素共役化) である. 以下に基本的な考え方を述べる: $\mathbf{V}^k$ を正規直交基底 $w_1^k, \cdots, w_m^k$ をもつ n-次元全体空間の部分空間, W は w_j^k を列成分とする行列, $S := W^H A W$, $\bar{\lambda}_j^k$ を S の固有値, T を S の固有ベクトルを列成分とする行列とする. $W T$ の列成分 x_j^k は A の固有ベクトルの近似であり, A の固有値を近似する Ritz 値は $\bar{\lambda}_j^k = (x_j^k)^H A x_j^k$ である. 次の条件を仮定する. $\bar{\lambda}_{j_s}^k, \cdots, \bar{\lambda}_{j_{s+l-1}}^k \in [\lambda_{\mathrm{lower}}, \lambda_{\mathrm{upper}}]$. $j \in j_s, \cdots, j_{s+l-1}$ に関して次の式を定義する.

$$q_j^k = (A - \bar{\lambda}_j^k I)\, x_j^k, \qquad r_j^k = (\bar{A} - \bar{\lambda}_j^k I)^{-1} q_j^k, \tag{4.1}$$

さらに $\mathbf{V}^{k+1} = \mathrm{span}(\mathbf{V}^k \cup r_{j_s}^k \cup \cdots \cup r_{j_{s+l-1}}^k)$ とする. ここに $\bar{A}$ は逆行列を容易に導けるような A の近似とする (文献 (25) では $\bar{A} = \mathrm{diag}(A)$ としている). そうすると, $\mathbf{V}^{k+1}$ は n-次元全体空間の $(m+l)$-次元部分空間であり,

一般的には前述の操作を繰り返すことにより与えられる固有値および固有ベクトルの近似は改良されていく. さらに再スタートを行えば効率が増すかもしれない. よい収束を得るためには, 固有値が λ_{lower} を下回るような A の全ての固有ベクトルを, $\mathbf{V}^k$ が粗い近似として含んでいなくてはならない (文献 (25)). 近似逆行列はあまり正確すぎてはならない. さもないと本手法は破綻してしまう. こうなる理由は文献 (26) で調べられている. このことにより, r_j^k の定義を改良した Jacobi-Davidson (JD) 法が導出される:

$$[(I - x_j^k\,(x_j^k)^H)\;(\bar{A} - \bar{\lambda}_j^k I)\;(I - x_j^k\,(x_j^k)^H)]\;r_j^k \;=\; q_j^k\,. \tag{4.2}$$

(4.2) の射影 $(I - x_j^k\,(x_j^k)^H)$ は具体的な行列の形に直して扱おうとすると簡単ではなくなるが, またそうする必要もなく, (4.2) を解くのは (4.1) を解くのよりもほんの少し多く計算コストがかかるだけである. この手法は $\bar{A} = A$ に関して 2 次的に収束する.

4.1.2.19 Jacobi-Davidson 法のための前処理

JD 法の特性は A に対する近似 $\bar{A}$ によって決定される. 前処理連立方程式 (4.2) の近似解を得るのに, 反復的なアプローチを試みることができる (文献 (20), (21), (26), (27)). ここでは, QMR アルゴリズムの実対称版あるいは複素エルミート版を用いる (文献 (22), (23), (24)). これらのアルゴリズムは, $\bar{A} = A$ に関する射影連立方程式 (4.2) に直接適用できる. QMR 反復の制御は以下の通り行う. 反復の終了は現在の残差ノルムが前回の内部 JD 反復での QMR 残差ノルムを下回った時点で行う. QMR 残差ノルムを制御することにより, 前処理連立方程式 (4.2) を最初に低い精度で解き, JD 反復を進行するにつれて解の精度が増していくようにすることができる. ブロック版 JD では, それぞれの前処理連立方程式 (4.2) の残差ノルムを, 近似すべきそれぞれの固有ベクトルについて別々に制御する. なぜなら, いくつかの固有ベクトルを近似するのは, 他の固有ベクトルを近似するのよりも難しいからである. このことは行列のスペクトルの性質を制御するのに適用できる.

複素エルミート QMR

$$p^0 = q^0 = d^0 = s^0 = 0,\ \nu^1 = 1,\quad \kappa^0 = -1,\quad w^1 = v^1 = r^0 = b - Bx^0$$

$$\gamma^1 = \|v^1\|,\quad \xi^1 = \gamma^1,\quad \rho^1 = (w^1)^T v^1,\ \epsilon^1 = (B^* w^1)^T v^1,\quad \mu^1 = 0, \tau^1 = \tfrac{\epsilon^1}{\rho^1}$$

$i = 1, 2, \cdots$

$$\begin{aligned}
p^i &= \frac{1}{\gamma^i} v^i - \mu^i p^{i-1} \\
q^i &= \frac{1}{\xi^i} B^* w^i - \frac{\gamma^i \mu^i}{\xi^i} q^{i-1} \\
v^{i+1} &= \boxed{Bp^i} - \frac{\tau^i}{\gamma^i} v^i \\
w^{i+1} &= q^i - \frac{\tau^i}{\xi^i} w^i
\end{aligned}$$

- if $(\ \|r^{i-1}\| \quad < \quad$ tolerance $)$ then STOP
- $\gamma^{i+1} = \|v^{i+1}\|$
- $\xi^{i+1} = \|w^{i+1}\|$
- $\rho^{i+1} = (w^{i+1})^T v^{i+1}$
- $\epsilon^{i+1} = (\ \boxed{B^* w^{i+1}}\ )^T v^{i+1}$

$$\begin{aligned}
\mu^{i+1} &= \frac{\gamma^i \xi^i \rho^{i+1}}{\gamma^{i+1} \tau^i \rho^i} \\
\tau^{i+1} &= \frac{\epsilon^{i+1}}{\rho^{i+1}} - \gamma^{i+1} \mu^{i+1} \\
\theta^i &= \frac{|\tau^i|^2 (1 - \nu^i)}{\nu^i |\tau^i|^2 + |\gamma^{i+1}|^2} \\
\kappa^i &= \frac{-\gamma^i (\tau^i)^* \kappa^{i-1}}{\nu^i |\tau^i|^2 + |\gamma^{i+1}|^2} \\
\nu^{i+1} &= \frac{\nu^i |\tau^i|^2}{\nu^i |\tau^i|^2 + |\gamma^{i+1}|^2} \\
d^i &= \theta^i d^{i-1} + \kappa^i p^i \\
s^i &= \theta^i s^{i-1} + \kappa^i B p^i \\
x^i &= x^{i-1} + d^i \\
r^i &= r^{i-1} - s^i
\end{aligned}$$

上のアルゴリズムは複素エルミート行列に対する JD に前処理を施すために使われている QMR 反復を示している．この手法は文献 (23) に述べられている QMR の変形から導かれる．JD の中では, 上のアルゴリズムの中の行列 B は前処理連立方程式 (4.2) の行列 $[(I - x_j^k\,(x_j^k)^H)\ (A - \bar{\lambda}_j^k I)\ (I - x_j^k\,(x_j^k)^H)]$ に相当している．QMR 反復ご

とに, B と B^* (上のアルゴリズム内の枠で印をつけた部分) を用いた 2 つの行列ベクトル演算が実行される．これは QMR が B^H ではなく B および $B^T = B^*$ についての演算を必要とする非エルミート Lanczos アルゴリズムに基づいているためである (文献 (24))．実対称問題の場合, QMR 反復ごとに必要とされる行列ベクトル演算はたった 1 回だけ行われる．なぜなら, このとき $q^i = Bp^i$ であり, したがって $v^{i+1} = q^i - (\tau^i/\gamma^i)v^i$ が成り立つからである．よって, 反復ごとに計算される行列ベクトル積は Bw^{i+1} の 1 回だけとなる．自然に考えれば, B のそれぞれの要素を $[(I - x_j^k\,(x_j^k)^H)\;(A - \bar{\lambda}_j^k I)\;(I - x_j^k\,(x_j^k)^H)]$ から計算することはない．たとえば, Bp^i という演算は複数のベクトル-ベクトル積演算と, A を用いた一個の行列ベクトル積演算に分割して行われる．

4.1.3 参考文献

(1) Wilkinson, J. H. and Reinsch, C. , “Handbook for Automatic Computation, Vol. II, Linear Algebra”, Springer-Verlag, (1971).

(2) Wilkinson, J. H. , “The Algebraic Eigenvalue Problem”, Clarendon Press, Oxford, (1965).

(3) 別府良孝, “スーパーコンピュータに適した固有値ルーチン”, bit 臨時増刊 (名取, 野寺編), 共立出版, (1987).

(4) 別府良孝, “固有値問題の高速算法”, 中央情報教育研究所, TN871-12, (1987).

(5) 別府良孝, 井坂秀高, 竹内聖彦, “3 重対角行列の固有値を求めるための諸算法について”, 情報処理学会第 42 回全国大会論文誌, Vol. 1, PP. 63-64(1991).

(6) Dongarra J. J. , Sorensen D. C. , and Hammarling A. J. , “Block reduction of matrices to condensed forms for eigenvalue computations”, Journal of Computational and Applied Mathematics, Vol. 27, PP. 215-227(1989).

(7) Dongarra J. J. and van de Geijn R. A. , “Reduction to Condensed Form for the Eigenvalue Problem on Distributed Memory Architectures”, LAPACK Working Note 30, PP. 1-12(1991).

(8) Francis, J. G. F. , “The QR transformation, I, II”, Comput. J. 4, pp. 265-271, pp. 332-345(1961, 1962).

(9) Cuppen, J. J. M., “A Divide and Conquer Method for the Symmetric Tridiagonal Eigenproblem”, Numer. Math. 36, pp. 177-195(1981).

(10) Gu, M. and Eisenstat, S. C., “A Stable and Efficient Algorithm for the rank-1 modification of the symmetric eigenproblem”, SIAM J. Matrix Anal. Appl. 15, pp. 1266-1276(1994).

(11) Gu, M. and Eisenstat, S. C., “A Divide-and-Conquer Algorithm for the Symmetric Tridiagonal Eigenproblem”, SIAM J. Matrix Anal. Appl. 16, pp. 172-191(1995).

(12) 戸川隼人, “マトリクスの数値計算”, オーム社, (1971).

(13) Moler, C. B and Stewart, G. W. , “An Algorithm for Generalized Matrix Eigenvalue Problems”, SIAM Numerical Analysis, Vol. 10, No. 2, pp. 241-256(1973).

(14) Ward, R. C. , “The Combination Shift QZ Algorithm”, SIAM Numerical Analysis, Vol. 12, No. 6, pp. 835-853(1973).

(15) Bathe and Wilson, “有限要素法の数値計算”, 菊池文雄訳, 科学技術出版社 (1979).

(16) 鷲津久一郎他, “有限要素法ハンドブック”, 培風館, (1981).

(17) Y. Beppu and I. Ninomiya, “HQRII—A Fast Diagonalization Subroutine”, Computers and Chemistry Vol. 6(1982).

(18) 井坂秀高, 別府良孝, 竹内聖彦, “QR 法の原点移動方法の比較”, 第 20 回数値解析シンポジウム講演予稿集. (1991)

(19) Basermann, A. , “Parallel preconditioned solvers for large sparse Hermitian eigenvalue problems” In *VECPAR’98 - Third International Conference for Vector and Parallel Processing. Lecture Notes in Computer Science*, Dongarra J and Hernandez V (eds). Springer: Berlin, 1999; **1573**:72–85.

(20) Basermann, A. , Steffen, B. , “New Preconditioned Solvers for Large Sparse Eigenvalue Problems on Massively Parallel Computers” In: Proceedings of the Eighth SIAM Conference on Parallel Processing for Scientific Computing (CD-ROM). SIAM, Philadelphia (1997)

(21) Basermann, A. , Steffen, B. , “Preconditioned solvers for large eigenvalue problems on massively parallel computers and workstation clusters” In *Parallel Computing: Fundamentals, Applications and New Directions*, D’Hollander EH, Joubert GR, Peters FJ, Trottenberg U (eds). Elsevier Science B. V. , 1998; 565–572.

(22) Basermann, A. , “QMR and TFQMR methods for sparse nonsymmetric problems on massively parallel systems” In *The Mathematics of Numerical Analysis. Series: Lectures in Applied Mathematics*, Renegar J, Shub M, Smale S (eds). AMS, 1996; **32**:59–76.

(23) Bücker, H. M. , Sauren, M. , “A Parallel Version of the Quasi-Minimal Residual Method Based on Coupled Two-Term Recurrences” In: Lecture Notes in Computer Science, Vol. 1184. Springer (1996) 157–165

(24) Freund, R. W. , Nachtigal, N. M. , “QMR: A Quasi-Minimal Residual Method for Non-Hermitian Linear Systems” Numer. Math. **60** (1991) 315–339

(25) Kosugi, N. , “Modifications of the Liu-Davidson Method for Obtaining One or Simultaneously Several Eigensolutions of a Large Real Symmetric Matrix” Comput. Phys. **55** (1984) 426–436

(26) Sleijpen GLG, van der Vorst HA. , “A Jacobi-Davidson iteration method for linear eigenvalue problems” *SIAM J. Matrix Anal. Appl.* 1996; **17**:401–425.

(27) Sleijpen GLG, van der Vorst HA, Meijerink E. , “Efficient expansion of subspaces in the Jacobi-Davidson method for standard and generalized eigenproblems” *ETNA* 1998; **7**:75–89.

(28) Lanczos, C. , “An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators”, J. Res. Nat. Bur. Standards, B45 (1950) 255–282.

(29) Paige, C. C. , “Computational Variants of the Lanczos Method for the Eigenproblem”, J. Inst. Math. Appl. , 10 (1972) 373–381.

(30) Simon, H. D. , “The Lanczos Algorithm with Partial Reorthogonalization”, Math. Comp. , 42 (1984) 115–142.

4.2 実行列 (2 次元配列型) (実数引数型)

4.2.1 DCGEAA, RCGEAA
実行列の全固有値・全固有ベクトル

(1) **機能**

実行列 A(2 次元配列型) の全固有値とそれに対応する全固有ベクトルを基本相似変換, ダブル QR 法により求める.

(2) **使用法**

倍精度サブルーチン:
  CALL DCGEAA (A, LNA, N, ER, EI, VE, LNV, IW1, W1, IERR)
単精度サブルーチン:
  CALL RCGEAA (A, LNA, N, ER, EI, VE, LNV, IW1, W1, IERR)

(3) **引数**

D:倍精度実数型 Z:倍精度複素数型
R:単精度実数型 C:単精度複素数型
I: {32 ビット整数版では INTEGER(4) / 64 ビット整数版では INTEGER(8)}

項番	引数名	型	大きさ	入出力	内容
1	A	D R	LNA, N	入力	実行列 A(2 次元配列型)
				出力	入力時の内容は保存されない
2	LNA	I	1	入力	配列 A の整合寸法
3	N	I	1	入力	行列 A の次数
4	ER	D R	N	出力	固有値の実部 (注意事項 (a), (b) 参照)
5	EI	D R	N	出力	固有値の虚部 (注意事項 (a), (b) 参照)
6	VE	D R	LNV, N	出力	固有ベクトル (注意事項 (c), (d) 参照)
7	LNV	I	1	入力	配列 VE の整合寸法
8	IW1	I	N	ワーク	作業領域
9	W1	D R	N	ワーク	作業領域
10	IERR	I	1	出力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNA}, \mathrm{LNV}$

(5) エラーインディケータ

IERR の値	意　　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	ER(1) ← A(1, 1), EI(1) ← 0.0, VE(1, 1) ← 1.0 とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
$5000 + i$	固有値を求める段階で収束しなかった. $(1 \leq i \leq \mathrm{N})$	ER, EI の 第 $(i+1), \cdots,$ 第 N 要素にそれまでに正しく求まった固有値が入る. このとき固有ベクトルは求まらない.

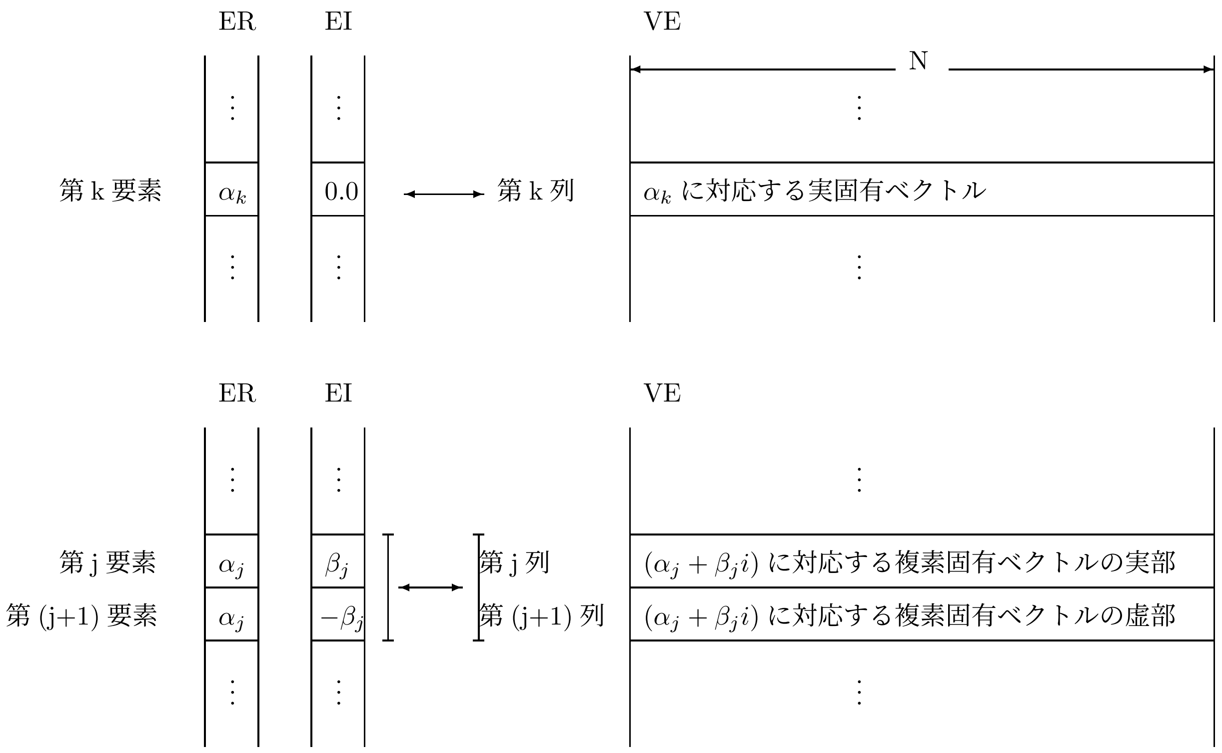
(6) 注意事項

(a) 固有値の実部は ER に, 虚部は EI に格納される. このとき, 第 j 要素の固有値が複素数であれば, それと共役な複素固有値が第 (j+1) 要素に格納される. ただし, 正の虚部が先に格納される.

(b) 固有値は添字の大きい方から求まり, j 番目に求まった固有値は ER, EI の第 (N−j+1) 要素に格納される. ただし, 求まる順番と固有値の大小は無関係である.

(c) 固有ベクトルは固有値 (ER, EI) の各固有値に対して図 4−1 のように格納される. すなわち, 第 k 要素の固有値が実数であれば, それに対応する実固有ベクトルが配列 VE の第 k 列に格納される. また, 第 j 要素および第 (j+1) 要素の固有値が 1 組の共役な複素固有値であれば, 第 j 要素の固有値に対応する複素固有ベクトルの実部, 虚部が, 配列 VE の第 j 列, 第 (j+1) 列にそれぞれ格納される. この複素固有ベクトルの共役なベクトルが, 第 (j+1) 要素の固有値に対応する固有ベクトルとなる.

図 4−1　固有値と固有ベクトルの格納の仕方



(d) 固有ベクトルはそのユークリッドノルムが, $\|x\|_2 = 1.0$ となるように正規化される.

(e) 固有ベクトルを必要としないときは, 4.2.2 $\left\{\begin{array}{l}\text{DCGEAN}\\ \text{RCGEAN}\end{array}\right\}$ を使用する.

(7) 使用例

(a) 問 題

$$A = \begin{bmatrix} 4 & -5 & 0 & 3 \\ 0 & 4 & -3 & -5 \\ 5 & -3 & 4 & 0 \\ 3 & 0 & 5 & 4 \end{bmatrix}$$

の全固有値とそれに対応する固有ベクトルを求める.

(b) 入力データ

行列 A, LNA=11, N=4, LNV=11

(c) 主プログラム

```
      PROGRAM BCGEAA
! *** EXAMPLE OF DCGEAA ***
      IMPLICIT REAL(8) (A-H,O-Z)
      PARAMETER ( ZERO = 0.0D0 )
      PARAMETER ( LNA = 11, LNV = 11 )
      DIMENSION A(LNA,LNA), ER(LNA), EI(LNA), VE(LNV,LNV),&
                IW1(LNA), W1(LNA)
!
      READ(5,*) N
      DO 10 I=1, N
         READ(5,*) (A(I,J), J=1, N)
   10 CONTINUE
!
      WRITE(6,1000) N
      DO 20 I=1, N
         WRITE(6,1100) (A(I,J), J=1, N)
   20 CONTINUE
!
```

```
      CALL DCGEAA(A,LNA,N,ER,EI,VE,LNV,IW1,W1,IERR)
!
      WRITE(6,1200)  IERR
!
      DO 70 J=1, N-1, 2
         WRITE(6,1300) (’EIGENVALUE ’, I=1, 2)
         WRITE(6,1400) ER(J), EI(J), ER(J+1), EI(J+1)
         WRITE(6,1300) (’EIGENVECTOR’, I=1, 2)
         IF(EI(J).EQ.ZERO) THEN
            IF(EI(J+1).EQ.ZERO) THEN
               DO 30 I=1, N
                  WRITE(6,1500) VE(I,J), ZERO, VE(I,J+1), ZERO
   30          CONTINUE
            ELSE
               DO 40 I=1, N
                  WRITE(6,1500) VE(I,J), ZERO, VE(I,J+1), VE(I,J+2)
   40          CONTINUE
            ENDIF
         ELSE
            IF(EI(J+1).EQ.ZERO) THEN
               DO 50 I=1, N
                  WRITE(6,1500) VE(I,J-1), -VE(I,J), VE(I,J+1), ZERO
   50          CONTINUE
            ELSE
               DO 60 I=1, N
                  WRITE(6,1500) VE(I,J), VE(I,J+1), VE(I,J), -VE(I,J+1)
   60          CONTINUE
            ENDIF
         ENDIF
   70 CONTINUE
      IF(MOD(N,2).NE.0) THEN
         WRITE(6,1300) ’EIGENVALUE ’
         WRITE(6,1400) ER(N), EI(N)
         WRITE(6,1300) ’EIGENVECTOR’
         IF(EI(N).EQ.ZERO) THEN
            DO 80 I=1, N
               WRITE(6,1500) VE(I,N), ZERO
   80       CONTINUE
         ELSE
            DO 90 I=1, N
               WRITE(6,1500) VE(I,N-1), -VE(I,N)
   90       CONTINUE
         ENDIF
      ENDIF
      STOP
!
 1000 FORMAT(’ ’,/,/,&
             ’ ***  DCGEAA  ***’,/,/,&
             ’  **  INPUT  **’,/,/,&
             ’      N = ’, I2,/,/,&
             ’      INPUT MATRIX A’,/)
 1100 FORMAT(7X, 11(F7.1))
 1200 FORMAT(’ ’,/,/,&
             ’  **  OUTPUT  **’,/,/,&
             ’      IERR = ’, I4)
 1300 FORMAT(’ ’,/, 2(14X, A11, 9X))
 1400 FORMAT(’ ’, 2(4X, 1PD13.6, ’ , ’, 1PD13.6, 1X))
 1500 FORMAT(’ ’, 2(4X, F13.10, ’ , ’, F13.10, 1X))
      END
```

(d) 出力結果

```
***  DCGEAA  ***

 **  INPUT  **

     N =  4

     INPUT MATRIX A

          4.0   -5.0    0.0    3.0
          0.0    4.0   -3.0   -5.0
          5.0   -3.0    4.0    0.0
          3.0    0.0    5.0    4.0


 **  OUTPUT  **

     IERR =    0

             EIGENVALUE                        EIGENVALUE
    1.200000D+01 ,  0.000000D+00      1.000000D+00 ,  5.000000D+00

             EIGENVECTOR                       EIGENVECTOR
    0.5000000000 ,  0.0000000000      0.1308649199 ,  0.4825705883
   -0.5000000000 ,  0.0000000000      0.4825705883 , -0.1308649199
    0.5000000000 ,  0.0000000000      0.4825705883 , -0.1308649199
    0.5000000000 ,  0.0000000000     -0.1308649199 , -0.4825705883

             EIGENVALUE                        EIGENVALUE
    1.000000D+00 , -5.000000D+00      2.000000D+00 ,  0.000000D+00

             EIGENVECTOR                       EIGENVECTOR
    0.1308649199 , -0.4825705883      0.5000000000 ,  0.0000000000
    0.4825705883 ,  0.1308649199      0.5000000000 ,  0.0000000000
    0.4825705883 ,  0.1308649199     -0.5000000000 ,  0.0000000000
   -0.1308649199 ,  0.4825705883      0.5000000000 ,  0.0000000000
```

4.2.2 DCGEAN, RCGEAN
実行列の全固有値

(1) 機能

実行列 A(2 次元配列型) の全固有値を基本相似変換, ダブル QR 法により求める.

(2) 使用法

倍精度サブルーチン:

CALL DCGEAN (A, LNA, N, ER, EI, IW1, W1, IERR)

単精度サブルーチン:

CALL RCGEAN (A, LNA, N, ER, EI, IW1, W1, IERR)

(3) 引数

D:倍精度実数型　Z:倍精度複素数型　I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\ \text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$
R:単精度実数型　C:単精度複素数型

項番	引数名	型	大きさ	入出力	内容
1	A	$\begin{Bmatrix}\mathrm{D}\\ \mathrm{R}\end{Bmatrix}$	LNA, N	入 力	実行列 A(2 次元配列型)
				出 力	入力時の内容は保存されない
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A の次数
4	ER	$\begin{Bmatrix}\mathrm{D}\\ \mathrm{R}\end{Bmatrix}$	N	出 力	固有値の実部 (注意事項 (a), (b) 参照)
5	EI	$\begin{Bmatrix}\mathrm{D}\\ \mathrm{R}\end{Bmatrix}$	N	出 力	固有値の虚部 (注意事項 (a), (b) 参照)
6	IW1	I	N	ワーク	作業領域
7	W1	$\begin{Bmatrix}\mathrm{D}\\ \mathrm{R}\end{Bmatrix}$	N	ワーク	作業領域
8	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a) $0 < \mathrm{N} \leq \mathrm{LNA}$

(5) エラーインディケータ

IERR の値	意　味	処 理 内 容
0	正常終了.	
1000	$\mathrm{N} = 1$ であった.	$\mathrm{ER}(1) \leftarrow \mathrm{A}(1,1)$, $\mathrm{EI}(1) \leftarrow 0.0$ とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
$5000 + i$	固有値を求める段階で収束しなかった. $(1 \leq i \leq \mathrm{N})$	ER, EI の 第 $(i+1), \cdots,$ 第 N 要素にそれまでに正しく求まった固有値が入る.

(6) **注意事項**

(a) 固有値の実部は ER に, 虚部は EI に格納される. このとき, 第 j 要素の固有値が複素数であれば, それと共役な複素固有値が第 (j+1) 要素に格納される. ただし, 正の虚部が先に格納される.

(b) 固有値は添字の大きい方から求まり, j 番目に求まった固有値は ER, EI の第 (N−j+1) 要素に格納される. ただし, 求まる順番と固有値の大小は無関係である.

4.3　実行列 (2 次元配列型) (複素引数型)

4.3.1　DCGNAA, RCGNAA
実行列の全固有値・全固有ベクトル

(1) **機 能**

実行列 A(2 次元配列型) の全固有値とそれに対応する全固有ベクトルを基本相似変換, ダブル QR 法により求める.

(2) **使用法**

倍精度サブルーチン:

CALL DCGNAA (A, LNA, N, E, VE, LNV, IW1, W1, IERR)

単精度サブルーチン:

CALL RCGNAA (A, LNA, N, E, VE, LNV, IW1, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: {32 ビット整数版では INTEGER(4)
64 ビット整数版では INTEGER(8)}

項番	引数名	型	大きさ	入出力	内　容
1	A	{D R}	LNA, N	入 力	実行列 A(2 次元配列型)
				出 力	入力時の内容は保存されない
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A の次数
4	E	{Z C}	N	出 力	固有値 (注意事項 (a), (b) 参照)
5	VE	{Z C}	LNV, N	出 力	固有ベクトル (注意事項 (c), (d) 参照)
6	LNV	I	1	入 力	配列 VE の整合寸法
7	IW1	I	N	ワーク	作業領域
8	W1	{D R}	N	ワーク	作業領域
9	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNA}, \mathrm{LNV}$

(5) **エラーインディケータ**

IERR の値	意　　　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	E(1) ← A(1, 1), VE(1, 1) ← 1.0 とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
$5000+i$	固有値を求める段階で収束しなかった. $(1 \leq i \leq \mathrm{N})$	E の 第 $(i+1), \cdots,$ 第 N 要素にそれまでに正しく求まった固有値が入る. このとき固有ベクトルは求まらない.

(6) **注意事項**

(a) 第 j 要素の固有値が複素数であれば, それと共役な複素固有値が第 (j+1) 要素に格納される. ただし, 正の虚部が先に格納される.

(b) 固有値は添字の大きい方から求まり, j 番目に求まった固有値は E の第 (N−j+1) 要素に格納される. ただし, 求まる順番と固有値の大小は無関係である.

(c) 固有値の第 k 要素に対応する固有ベクトルは, 配列 VE の第 k 列に格納される.

(d) 固有ベクトルはそのユークリッドノルムが, $\|x\|_2 = 1.0$ となるように正規化される.

(e) 固有ベクトルを必要としないときは, 4.3.2 $\left\{\begin{matrix}\text{DCGNAN}\\ \text{RCGNAN}\end{matrix}\right\}$を使用する.

(7) **使用例**

(a) 問 題

$$A = \begin{bmatrix} 4 & -5 & 0 & 3 \\ 0 & 4 & -3 & -5 \\ 5 & -3 & 4 & 0 \\ 3 & 0 & 5 & 4 \end{bmatrix}$$

の全固有値とそれに対応する固有ベクトルを求める.

(b) 入力データ

行列 A, LNA=11, N=4, LNV=11

(c) 主プログラム

```
      PROGRAM BCGNAA
! *** EXAMPLE OF DCGNAA ***
      IMPLICIT REAL(8) (A-H,O-Z)
      PARAMETER ( LNA = 11, LNV = 11 )
      COMPLEX(8) E,VE
      DIMENSION A(LNA,LNA),IW1(LNA),W1(LNA),E(LNA),VE(LNV,LNV)
!
      READ(5,*) N
      DO 10 I=1, N
         READ(5,*) (A(I,J), J=1, N)
   10 CONTINUE
!
      WRITE(6,1000) N
      DO 20 I=1, N
         WRITE(6,1100) (A(I,J), J=1, N)
   20 CONTINUE
!
      CALL DCGNAA(A,LNA,N,E,VE,LNV,IW1,W1,IERR)
!
      WRITE(6,1200)  IERR
!
      DO 70 J=1, N-1, 2
         WRITE(6,1300) ('EIGENVALUE ', I=1, 2)
```

```
         WRITE(6,1400) E(J),  E(J+1)
         WRITE(6,1300) ('EIGENVECTOR', I=1, 2)
               DO 30 I=1, N
                  WRITE(6,1500) VE(I,J),  VE(I,J+1)
   30          CONTINUE
   70 CONTINUE
      IF(MOD(N,2).NE.0) THEN
         WRITE(6,1300) 'EIGENVALUE '
         WRITE(6,1400) E(N)
         WRITE(6,1300) 'EIGENVECTOR'
            DO 80 I=1, N
               WRITE(6,1500) VE(I,N)
   80       CONTINUE
      ENDIF
      STOP
!
 1000 FORMAT(' ',/,/,&
             ' ***  DCGNAA  ***',/,/,&
             '  **  INPUT  **',/,/,&
             '      N = ', I2,/,/,&
             '      INPUT MATRIX A',/)
 1100 FORMAT(7X, 11(F7.1))
 1200 FORMAT(' ',/,/,&
             '  **  OUTPUT  **',/,/,&
             '      IERR = ', I4)
 1300 FORMAT(' ',/, 2(14X, A11, 9X))
 1400 FORMAT(' ', 2(4X, 1PD13.6, ' , ', 1PD13.6, 1X))
 1500 FORMAT(' ', 2(4X, F13.10, ' , ', F13.10, 1X))
      END
```

(d) 出力結果

```
***  DCGNAA  ***

 **  INPUT  **

     N =  4

     INPUT MATRIX A

          4.0   -5.0    0.0    3.0
          0.0    4.0   -3.0   -5.0
          5.0   -3.0    4.0    0.0
          3.0    0.0    5.0    4.0


 **  OUTPUT  **

     IERR =    0

             EIGENVALUE                       EIGENVALUE
     1.200000D+01 ,  0.000000D+00      1.000000D+00 ,  5.000000D+00

             EIGENVECTOR                      EIGENVECTOR
     0.5000000000 ,  0.0000000000      0.1308649199 ,  0.4825705883
    -0.5000000000 ,  0.0000000000      0.4825705883 , -0.1308649199
     0.5000000000 ,  0.0000000000      0.4825705883 , -0.1308649199
     0.5000000000 ,  0.0000000000     -0.1308649199 , -0.4825705883

             EIGENVALUE                       EIGENVALUE
     1.000000D+00 , -5.000000D+00      2.000000D+00 ,  0.000000D+00

             EIGENVECTOR                      EIGENVECTOR
     0.1308649199 , -0.4825705883      0.5000000000 ,  0.0000000000
     0.4825705883 ,  0.1308649199      0.5000000000 ,  0.0000000000
     0.4825705883 ,  0.1308649199     -0.5000000000 ,  0.0000000000
    -0.1308649199 ,  0.4825705883      0.5000000000 ,  0.0000000000
```

4.3.2 DCGNAN, RCGNAN
実行列の全固有値

(1) **機 能**

実行列 A(2 次元配列型) の全固有値を基本相似変換, ダブル QR 法により求める.

(2) **使用法**

倍精度サブルーチン:

CALL DCGNAN (A, LNA, N, E, IW1, W1, IERR)

単精度サブルーチン:

CALL RCGNAN (A, LNA, N, E, IW1, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\left\{ \begin{array}{l} \text{32 ビット整数版では INTEGER(4)} \\ \text{64 ビット整数版では INTEGER(8)} \end{array} \right\}$

項番	引数名	型	大きさ	入出力	内　容
1	A	$\left\{\begin{matrix} \text{D} \\ \text{R} \end{matrix}\right\}$	LNA, N	入 力	実行列 A(2 次元配列型)
				出 力	入力時の内容は保存されない
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A の次数
4	E	$\left\{\begin{matrix} \text{Z} \\ \text{C} \end{matrix}\right\}$	N	出 力	固有値 (注意事項 (a), (b) 参照)
5	IW1	I	N	ワーク	作業領域
6	W1	$\left\{\begin{matrix} \text{D} \\ \text{R} \end{matrix}\right\}$	N	ワーク	作業領域
7	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \text{N} \leq \text{LNA}$

(5) **エラーインディケータ**

IERR の値	意　味	処 理 内 容
0	正常終了.	
1000	N = 1 であった.	$\text{E}(1) \leftarrow \text{A}(1,1)$ とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
$5000 + i$	固有値を求める段階で収束しなかった. $(1 \leq i \leq \text{N})$	E の 第 $(i+1), \cdots,$ 第 N 要素にそれまでに正しく求まった固有値が入る.

(6) **注意事項**

(a) 第 j 要素の固有値が複素数であれば, それと共役な複素固有値が第 (j+1) 要素に格納される. ただし, 正の虚部が先に格納される.

(b) 固有値は添字の大きい方から求まり, j 番目に求まった固有値は E の第 (N−j+1) 要素に格納される. ただし, 求まる順番と固有値の大小は無関係である.

4.4 複素行列 (2 次元配列型) (実数引数型)

4.4.1 ZCGEAA, CCGEAA
複素行列の全固有値・全固有ベクトル

(1) **機 能**

複素行列 A=(AR, AI)(2 次元配列型)(実数引数型) の全固有値とそれに対応する全固有ベクトルを基本相似変換, QR 法により求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZCGEAA (AR, AI, LNA, N, ER, EI, VR, VI, LNV, W1, IERR)

単精度サブルーチン:

CALL CCGEAA (AR, AI, LNA, N, ER, EI, VR, VI, LNV, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\left\{\begin{array}{l}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{array}\right\}$

項番	引数名	型	大きさ	入出力	内　容
1	AR	D R	LNA, N	入 力	複素行列 A の実部 (2 次元配列型)
				出 力	入力時の内容は保存されない
2	AI	D R	LNA, N	入 力	複素行列 A の虚部 (2 次元配列型)
				出 力	入力時の内容は保存されない
3	LNA	I	1	入 力	配列 AR, AI の整合寸法
4	N	I	1	入 力	行列 A の次数
5	ER	D R	N	出 力	固有値の実部 (注意事項 (a) 参照)
6	EI	D R	N	出 力	固有値の虚部 (注意事項 (a) 参照)
7	VR	D R	LNV, N	出 力	固有値 (ER, EI) に対応する固有ベクトルの実部 (列ベクトル) (注意事項 (b), (c) 参照)
8	VI	D R	LNV, N	出 力	固有値 (ER, EI) に対応する固有ベクトルの虚部 (列ベクトル) (注意事項 (b), (c) 参照)
9	LNV	I	1	入 力	配列 VR, VI の整合寸法
10	W1	D R	3×N	ワーク	作業領域
11	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNA}, \mathrm{LNV}$

(5) **エラーインディケータ**

IERR の値	意味	処理内容
0	正常終了.	
1000	N = 1 であった.	ER(1) ← AR(1, 1), EI(1) ← AI(1, 1), VR(1, 1) ← 1.0, VI(1, 1) ← 0.0 とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
$5000 + i$	固有値を求める段階で収束しなかった. $(1 \leq i \leq \mathrm{N})$	ER, EI の 第 $(i+1), \cdots,$ 第 N 要素にそれまでに正しく求まった固有値が入る. このとき固有ベクトルは求まらない.

(6) **注意事項**

(a) 固有値の実部は ER に, 虚部は EI に格納される. このとき固有値は添字の大きい方から求まり, j 番目に求まった固有値は ER, EI の第 (N−j+1) 要素に格納される. ただし, 求まる順番と固有値の大小は無関係である.

(b) 固有値 (ER, EI) の第 k 要素に対応する固有ベクトルの実部, 虚部は, それぞれ VR, VI の第 k 列に格納される.

(c) 固有ベクトルはそのユークリッドノルムが, $\|\mathrm{x}\|_2 = 1.0$ となるように正規化される.

(d) 固有ベクトルを必要としないときは, 4.4.2 $\left\{\begin{array}{c}\mathrm{ZCGEAN}\\ \mathrm{CCGEAN}\end{array}\right\}$ を使用する.

(7) **使用例**

(a) 問 題

$$A = \begin{bmatrix} 5+9i & 5+5i & -6-6i & -7-7i \\ 3+3i & 6+10i & -5-5i & -6-6i \\ 2+2i & 3+3i & -1+3i & -5-5i \\ 1+i & 2+2i & -3-3i & 4i \end{bmatrix}$$

の全固有値とそれに対応する固有ベクトルを求める.

(b) 入力データ

行列 A の実部 AR, 虚部 AI, LNA=11, N=4, LNV=11

(c) 主プログラム

```
      PROGRAM ACGEAA
! *** EXAMPLE OF ZCGEAA ***
      IMPLICIT REAL(8)  (A-H,O-Z)
      CHARACTER*80 FMT
      PARAMETER ( LNA = 11, LNV = 11 )
      DIMENSION  AR(LNA,LNA),AI(LNA,LNA),ER(LNA),EI(LNA),&
                 VR(LNV,LNV),VI(LNV,LNV),W1(3*LNA)
!
      READ(5,*) N
      DO 10 I=1, N
         READ(5,*) (AR(I,J), AI(I,J), J=1, N)
   10 CONTINUE
!
      WRITE(6,1000) N
      DO 20 I=1, N
         WRITE(FMT,1100) N
         WRITE(6,FMT) (AR(I,J), AI(I,J), J=1, N)
   20 CONTINUE
!
```

```
      CALL ZCGEAA(AR,AI,LNA,N,ER,EI,VR,VI,LNV,W1,IERR)
!
      WRITE(6,1200) IERR
!
      DO 40 J=1, N-1, 2
         WRITE(6,1300) (’EIGENVALUE’, I=1, 2)
         WRITE(6,1400) ER(J), EI(J), ER(J+1), EI(J+1)
         WRITE(6,1300) (’EIGENVECTOR’, I=1, 2)
         DO 30 I=1, N
            WRITE(6,1500) VR(I,J), VI(I,J), VR(I,J+1), VI(I,J+1)
   30    CONTINUE
   40 CONTINUE
      IF(MOD(N,2).NE.0) THEN
         WRITE(6,1300) ’EIGENVALUE ’
         WRITE(6,1400) ER(N), EI(N)
         WRITE(6,1300) ’EIGENVECTOR’
         DO 50 I=1, N
            WRITE(6,1500) VR(I,N), VI(I,N)
   50    CONTINUE
      ENDIF
      STOP
!
 1000 FORMAT(’ ’,/,/,&
             ’ ***  ZCGEAA  ***’,/,/,&
             ’  **  INPUT  **’,/,/,&
             ’      N = ’, I4,/,/,&
             ’      INPUT MATRIX A ( REAL,IMAGINARY )’,/)
 1100 FORMAT(’( ,5X,’,I2,’(’’(’’,F5.1,’’ , ’’, F5.1,’’) ’’))’)
 1200 FORMAT(’ ’,/,/,&
             ’  **  OUTPUT  **’,/,/,&
             ’      IERR = ’, I4)
 1300 FORMAT(’ ’,/, 2(14X, A11, 9X))
 1400 FORMAT(’ ’, 2(4X, 1PD13.6, ’ , ’, 1PD13.6, 1X))
 1500 FORMAT(’ ’, 2(4X, F13.10, ’ , ’, F13.10, 1X))
      END
```

(d) 出力結果

```
***  ZCGEAA  ***

 **  INPUT  **

     N =    4

     INPUT MATRIX A ( REAL,IMAGINARY )

    (  5.0 ,   9.0) (  5.0 ,   5.0) ( -6.0 ,  -6.0) ( -7.0 ,  -7.0)
    (  3.0 ,   3.0) (  6.0 ,  10.0) ( -5.0 ,  -5.0) ( -6.0 ,  -6.0)
    (  2.0 ,   2.0) (  3.0 ,   3.0) ( -1.0 ,   3.0) ( -5.0 ,  -5.0)
    (  1.0 ,   1.0) (  2.0 ,   2.0) ( -3.0 ,  -3.0) (  0.0 ,   4.0)


 **  OUTPUT  **

     IERR =    0

              EIGENVALUE                         EIGENVALUE
     4.000000D+00 ,  8.000000D+00      2.000000D+00 ,  6.000000D+00

             EIGENVECTOR                        EIGENVECTOR
     0.5419737851 , -0.1989918330      0.3438224256 , -0.1569817904
     0.5419737851 , -0.1989918330      0.6876448512 , -0.3139635808
     0.5419737851 , -0.1989918330      0.3438224256 , -0.1569817904
    -0.0000000000 ,  0.0000000000      0.3438224256 , -0.1569817904

              EIGENVALUE                         EIGENVALUE
     3.000000D+00 ,  7.000000D+00      1.000000D+00 ,  5.000000D+00

             EIGENVECTOR                        EIGENVECTOR
    -0.2921601411 ,  0.4979716712     -0.3883659462 ,  0.6485371718
    -0.2921601411 ,  0.4979716712     -0.1941829731 ,  0.3242685859
     0.0000000000 , -0.0000000000     -0.1941829731 ,  0.3242685859
    -0.2921601411 ,  0.4979716712     -0.1941829731 ,  0.3242685859
```

4.4.2 ZCGEAN, CCGEAN
複素行列の全固有値

(1) 機 能

複素行列 A=(AR, AI)(2 次元配列型)(実数引数型) の全固有値を基本相似変換, QR 法により求める.

(2) 使用法

倍精度サブルーチン:

CALL ZCGEAN (AR, AI, LNA, N, ER, EI, IERR)

単精度サブルーチン:

CALL CCGEAN (AR, AI, LNA, N, ER, EI, IERR)

(3) 引 数

D:倍精度実数型 Z:倍精度複素数型
R:単精度実数型 C:単精度複素数型

I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\ \text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$

項番	引数名	型	大きさ	入出力	内 容
1	AR	$\begin{Bmatrix}\mathrm{D}\\ \mathrm{R}\end{Bmatrix}$	LNA, N	入 力	複素行列 A の実部 (2 次元配列型)
				出 力	入力時の内容は保存されない
2	AI	$\begin{Bmatrix}\mathrm{D}\\ \mathrm{R}\end{Bmatrix}$	LNA, N	入 力	複素行列 A の虚部 (2 次元配列型)
				出 力	入力時の内容は保存されない
3	LNA	I	1	入 力	配列 AR, AI の整合寸法
4	N	I	1	入 力	行列 A の次数
5	ER	$\begin{Bmatrix}\mathrm{D}\\ \mathrm{R}\end{Bmatrix}$	N	出 力	固有値の実部 (注意事項 (a) 参照)
6	EI	$\begin{Bmatrix}\mathrm{D}\\ \mathrm{R}\end{Bmatrix}$	N	出 力	固有値の虚部 (注意事項 (a) 参照)
7	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a) $0 < \mathrm{N} \leq \mathrm{LNA}$

(5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
1000	$\mathrm{N} = 1$ であった.	$\mathrm{ER}(1) \leftarrow \mathrm{AR}(1,1)$, $\mathrm{EI}(1) \leftarrow \mathrm{AI}(1,1)$ とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
$5000 + i$	固有値を求める段階で収束しなかった. $(1 \leq i \leq \mathrm{N})$	ER, EI の 第 $(i+1), \ldots,$ 第 N 要素にそれまでに正しく求まった固有値が入る.

(6) **注意事項**

(a) 固有値の実部は ER に, 虚部は EI に格納される. このとき, 固有値は添字の大きい方から求まり, j 番目に求まった固有値は ER, EI の第 (N−j+1) 要素に格納される. ただし, 求まる順番と固有値の大小は無関係である.

4.5 複素行列 (2次元配列型) (複素引数型)

4.5.1 ZCGNAA, CCGNAA
複素行列の全固有値・全固有ベクトル

(1) **機 能**

複素行列 A(2次元配列型)(複素引数型) の全固有値とそれに対応する全固有ベクトルを基本相似変換, QR法により求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZCGNAA (A, LNA, N, E, VE, LNV, W1, WK, IERR)

単精度サブルーチン:

CALL CCGNAA (A, LNA, N, E, VE, LNV, W1, WK, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: {32ビット整数版ではINTEGER(4) / 64ビット整数版ではINTEGER(8)}

項番	引数名	型	大きさ	入出力	内容
1	A	Z C	LNA, N	入 力	複素行列 A(2次元配列型)
				出 力	入力時の内容は保存されない
2	LNA	I	1	入 力	配列Aの整合寸法
3	N	I	1	入 力	行列 A の次数
4	E	Z C	N	出 力	固有値 (注意事項 (a) 参照)
5	VE	Z C	LNV, N	出 力	固有値 (E) に対応する固有ベクトル (列ベクトル) (注意事項 (b), (c) 参照)
6	LNV	I	1	入 力	配列VEの整合寸法
7	W1	D R	N	ワーク	作業領域
8	WK	Z C	N	ワーク	作業領域
9	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNA}, \mathrm{LNV}$

(5) **エラーインディケータ**

IERR の値	意　　　　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	E(1) ← A(1, 1), VE(1, 1) ← 1.0 とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
$5000 + i$	固有値を求める段階で収束しなかった. $(1 \le i \le \mathrm{N})$	E の 第 $(i+1), \ldots,$ 第 N 要素にそれまでに正しく求まった固有値が入る. このとき固有ベクトルは求まらない.

(6) **注意事項**

(a) 固有値は添字の大きい方から求まり, j 番目に求まった固有値は E の第 (N−j+1) 要素に格納される. ただし, 求まる順番と固有値の大小は無関係である.

(b) 固有値の第 k 要素に対応する固有ベクトルは, 配列 VE の第 k 列に格納される.

(c) 固有ベクトルはそのユークリッドノルムが, $\|\mathrm{x}\|_2 = 1.0$ となるように正規化される.

(d) 固有ベクトルを必要としないときは, 4.5.2 $\left\{\begin{matrix}\mathrm{ZCGNAN}\\ \mathrm{CCGNAN}\end{matrix}\right\}$ を使用する.

(7) **使用例**

(a) 問 題

$$A = \begin{bmatrix} 5+9i & 5+5i & -6-6i & -7-7i \\ 3+3i & 6+10i & -5-5i & -6-6i \\ 2+2i & 3+3i & -1+3i & -5-5i \\ 1+i & 2+2i & -3-3i & 4i \end{bmatrix}$$

の全固有値とそれに対応する固有ベクトルを求める.

(b) 入力データ

行列 A, LNA=11, N=4, LNV=11

(c) 主プログラム

```
      PROGRAM ACGNAA
! *** EXAMPLE OF ZCGNAA ***
      IMPLICIT REAL(8)  (A-H,O-Z)
      CHARACTER*80 FMT
      PARAMETER ( LNA = 11, LNV = 11 )
      COMPLEX(8) A,E,VE,WK
      DIMENSION  A(LNA,LNA),E(LNA),VE(LNV,LNV),W1(LNA),WK(LNA)
!
      READ(5,*) N
      DO 10 I=1, N
         READ(5,*) (A(I,J), J=1, N)
   10 CONTINUE
!
      WRITE(6,1000) N
      DO 20 I=1, N
         WRITE(FMT,1100) N
         WRITE(6,FMT) (A(I,J), J=1, N)
   20 CONTINUE
!
      CALL ZCGNAA(A,LNA,N,E,VE,LNV,W1,WK,IERR)
!
      WRITE(6,1200) IERR
!
      DO 40 J=1, N-1, 2
         WRITE(6,1300) ('EIGENVALUE', I=1, 2)
         WRITE(6,1400) E(J), E(J+1)
         WRITE(6,1300) ('EIGENVECTOR', I=1, 2)
         DO 30 I=1, N
```

```
            WRITE(6,1500) VE(I,J), VE(I,J+1)
   30    CONTINUE
   40 CONTINUE
      IF(MOD(N,2).NE.0) THEN
         WRITE(6,1300) ’EIGENVALUE ’
         WRITE(6,1400) E(N)
         WRITE(6,1300) ’EIGENVECTOR’
         DO 50 I=1, N
            WRITE(6,1500) VE(I,N)
   50    CONTINUE
      ENDIF
      STOP
!
 1000 FORMAT(’ ’,/,/,&
             ’ ***  ZCGNAA  ***’,/,/,&
             ’  **  INPUT  **’,/,/,&
             ’      N = ’, I4,/,/,&
             ’      INPUT MATRIX A ( REAL,IMAGINARY )’,/)
 1100 FORMAT(’( ,5X,’,I2,’(’’(’’,F5.1,’’ , ’’, F5.1,’’) ’’))’)
 1200 FORMAT(’ ’,/,/,&
             ’  **  OUTPUT  **’,/,/,&
             ’      IERR = ’, I4)
 1300 FORMAT(’ ’,/, 2(14X, A11, 9X))
 1400 FORMAT(’ ’, 2(4X, 1PD13.6, ’ , ’, 1PD13.6, 1X))
 1500 FORMAT(’ ’, 2(4X, F13.10, ’ , ’, F13.10, 1X))
      END
```

(d) 出力結果

```
***  ZCGNAA  ***

 **  INPUT  **

     N =    4

     INPUT MATRIX A ( REAL,IMAGINARY )

    (  5.0 ,   9.0) (  5.0 ,   5.0) ( -6.0 ,  -6.0) ( -7.0 ,  -7.0)
    (  3.0 ,   3.0) (  6.0 ,  10.0) ( -5.0 ,  -5.0) ( -6.0 ,  -6.0)
    (  2.0 ,   2.0) (  3.0 ,   3.0) ( -1.0 ,   3.0) ( -5.0 ,  -5.0)
    (  1.0 ,   1.0) (  2.0 ,   2.0) ( -3.0 ,  -3.0) (  0.0 ,   4.0)


 **  OUTPUT  **

     IERR =    0

              EIGENVALUE                        EIGENVALUE
     4.000000D+00 ,  8.000000D+00      2.000000D+00 ,  6.000000D+00

             EIGENVECTOR                       EIGENVECTOR
     0.5419737851 , -0.1989918330     -0.3496783544 , -0.1434649481
     0.5419737851 , -0.1989918330     -0.6993567087 , -0.2869298963
     0.5419737851 , -0.1989918330     -0.3496783544 , -0.1434649481
     0.0000000000 ,  0.0000000000     -0.3496783544 , -0.1434649481

              EIGENVALUE                        EIGENVALUE
     3.000000D+00 ,  7.000000D+00      1.000000D+00 ,  5.000000D+00

             EIGENVECTOR                       EIGENVECTOR
    -0.2921601411 ,  0.4979716712     -0.3172488495 ,  0.6861353649
    -0.2921601411 ,  0.4979716712     -0.1586244247 ,  0.3430676824
     0.0000000000 ,  0.0000000000     -0.1586244247 ,  0.3430676824
    -0.2921601411 ,  0.4979716712     -0.1586244247 ,  0.3430676824
```

4.5.2 ZCGNAN, CCGNAN
複素行列の全固有値

(1) **機 能**

複素行列 A(2 次元配列型)(複素引数型) の全固有値を基本相似変換, QR 法により求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZCGNAN (A, LNA, N, E, W1, IERR)

単精度サブルーチン:

CALL CCGNAN (A, LNA, N, E, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\left\{\begin{array}{l}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{array}\right\}$

項番	引数名	型	大きさ	入出力	内 容
1	A	$\left\{\begin{array}{c}\text{Z}\\\text{C}\end{array}\right\}$	LNA, N	入 力	複素行列 A(2 次元配列型)
				出 力	入力時の内容は保存されない
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A の次数
4	E	$\left\{\begin{array}{c}\text{Z}\\\text{C}\end{array}\right\}$	N	出 力	固有値 (注意事項 (a) 参照)
5	W1	$\left\{\begin{array}{c}\text{D}\\\text{R}\end{array}\right\}$	N	ワーク	作業領域
6	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \text{N} \leq \text{LNA}$

(5) **エラーインディケータ**

IERR の値	意 味	処 理 内 容
0	正常終了.	
1000	N = 1 であった.	E(1) ← A(1, 1) とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
$5000 + i$	固有値を求める段階で収束しなかった. ($1 \leq i \leq$ N)	E の 第 $(i+1), \cdots,$ 第 N 要素にそれまでに正しく求まった固有値が入る.

(6) **注意事項**

(a) 固有値は添字の大きい方から求まり, j 番目に求まった固有値は E の第 (N−j+1) 要素に格納される. ただし, 求まる順番と固有値の大小は無関係である.

4.6　実対称行列 (2 次元配列型) (上三角型)

4.6.1　DCSMAA, RCSMAA
実対称行列の全固有値・全固有ベクトル

(1) **機 能**

実対称行列 A(2 次元配列型)(上三角型) の全固有値とそれに対応する全固有ベクトルをハウスホルダー法, QR 法により求める.

(2) **使用法**

倍精度サブルーチン:

CALL DCSMAA (A, LNA, N, E, W1, IERR)

単精度サブルーチン:

CALL RCSMAA (A, LNA, N, E, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\left\{\begin{array}{l}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{array}\right\}$

項番	引数名	型	大きさ	入出力	内　　容
1	A	$\left\{\begin{matrix}D\\R\end{matrix}\right\}$	LNA, N	入 力	実対称行列 A(2 次元配列型)(上三角型)
				出 力	各固有値に対応する固有ベクトル (列ベクトル)
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A の次数
4	E	$\left\{\begin{matrix}D\\R\end{matrix}\right\}$	N	出 力	固有値
5	W1	$\left\{\begin{matrix}D\\R\end{matrix}\right\}$	N	ワーク	作業領域
6	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNA}$

(5) **エラーインディケータ**

IERR の値	意　　味	処 理 内 容
0	正常終了.	
1000	$\mathrm{N} = 1$ であった.	$\mathrm{E}(1) \leftarrow \mathrm{A}(1,1)$, $\mathrm{A}(1,1) \leftarrow 1.0$ とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
$5000 + i$	固有値, 固有ベクトルを求める段階で収束しなかった. $(1 \leq i \leq \mathrm{N})$	$\mathrm{E}(1), \cdots, \mathrm{E}(i-1)$ にそれまでに求まった固有値, A にそれに対応する固有ベクトルが入る (ただし順不同).

(6) **注意事項**

(a) 配列 A には, 上三角部分のみにデータが格納されていればよい.

(b) 固有値は小さい順に格納される.

(c) 固有ベクトルは正規直交系である.

(d) 固有ベクトルを必要としないときは, 4.6.2 $\begin{Bmatrix} \text{DCSMAN} \\ \text{RCSMAN} \end{Bmatrix}$を使用する.

(7) **使用例**

(a) 問 題

$$A = \begin{bmatrix} 6 & 4 & 4 & 1 \\ 4 & 6 & 1 & 4 \\ 4 & 1 & 6 & 4 \\ 1 & 4 & 4 & 6 \end{bmatrix}$$

の全固有値とそれに対応する固有ベクトルを求める.

(b) 入力データ

行列 A, LNA=11, N=4

(c) 主プログラム

```
      PROGRAM BCSMAA
! *** EXAMPLE OF DCSMAA ***
      IMPLICIT REAL(8) (A-H,O-Z)
      PARAMETER ( LNA = 11 )
      DIMENSION A(LNA,LNA), E(LNA), W1(LNA)
!
      READ(5,*) N
      DO 10 I=1, N
         READ(5,*) (A(I,J), J=I, N)
   10 CONTINUE
!
      WRITE(6,1000) N
      DO 20 I=1, N
         WRITE(6,1100) (A(J,I), J=1, I-1), (A(I,J), J=I, N)
   20 CONTINUE
!
      CALL DCSMAA(A,LNA,N,E,W1,IERR)
!
      WRITE(6,1200) IERR
!
      DO 40 K=1, N-3, 4
         WRITE(6,1300) ('EIGENVALUE ', I=1, 4)
         WRITE(6,1400) (E(I), I=K, K+3)
         WRITE(6,1300) ('EIGENVECTOR', I=1, 4)
         DO 30 J=1, N
            WRITE(6,1500) (A(J,I), I=K, K+3)
   30    CONTINUE
   40 CONTINUE
      IF(MOD(N,4).NE.0) THEN
         WRITE(6,1300) ('EIGENVALUE ', I=N/4*4+1, N)
         WRITE(6,1400) (E(I), I=N/4*4+1, N)
         WRITE(6,1300) ('EIGENVECTOR', I=N/4*4+1, N)
         DO 50 J=1, N
            WRITE(6,1500) (A(J,I), I=N/4*4+1, N)
   50    CONTINUE
      ENDIF
      STOP
!
 1000 FORMAT(' ',/,/,&
             ' ***  DCSMAA  ***',/,/,&
             '  **  INPUT  **',/,/,&
             '      N = ', I2,/,/,&
             '      INPUT MATRIX A',/)
 1100 FORMAT(7X, 11(F7.1))
 1200 FORMAT(' ',/,/,&
             '  **  OUTPUT  **',/,/,&
             '      IERR = ', I4)
 1300 FORMAT(' ',/,1X, 4(5X, A11, 2X))
 1400 FORMAT(3X, 4(2X, 1PD14.7, 2X))
 1500 FORMAT(2X, 4(F14.8, 4X))
      END
```

(d) 出力結果

```
***  DCSMAA  ***

 **  INPUT  **

     N =  4

     INPUT MATRIX A

          6.0    4.0    4.0    1.0
          4.0    6.0    1.0    4.0
          4.0    1.0    6.0    4.0
          1.0    4.0    4.0    6.0

 **  OUTPUT  **

     IERR =    0

     EIGENVALUE         EIGENVALUE         EIGENVALUE         EIGENVALUE
    -1.0000000D+00      5.0000000D+00      5.0000000D+00      1.5000000D+01

     EIGENVECTOR        EIGENVECTOR        EIGENVECTOR        EIGENVECTOR
     0.50000000         0.70710678         0.00000000         0.50000000
    -0.50000000         0.00000000        -0.70710678         0.50000000
    -0.50000000        -0.00000000         0.70710678         0.50000000
     0.50000000        -0.70710678         0.00000000         0.50000000
```

4.6.2 DCSMAN, RCSMAN
実対称行列の全固有値

(1) **機 能**

実対称行列 A(2 次元配列型)(上三角型) の全固有値をハウスホルダー法, 無平方根 QR 法により求める.

(2) **使用法**

倍精度サブルーチン:

CALL DCSMAN (A, LNA, N, E, W1, IERR)

単精度サブルーチン:

CALL RCSMAN (A, LNA, N, E, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: { 32 ビット整数版では INTEGER(4) / 64 ビット整数版では INTEGER(8) }

項番	引数名	型	大きさ	入出力	内　　容
1	A	$\begin{Bmatrix} \mathrm{D} \\ \mathrm{R} \end{Bmatrix}$	LNA, N	入 力	実対称行列 A (2 次元配列型) (上三角型)
				出 力	入力時の内容は保存されない
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A の次数
4	E	$\begin{Bmatrix} \mathrm{D} \\ \mathrm{R} \end{Bmatrix}$	N	出 力	固有値
5	W1	$\begin{Bmatrix} \mathrm{D} \\ \mathrm{R} \end{Bmatrix}$	N	ワーク	作業領域
6	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNA}$

(5) **エラーインディケータ**

IERR の値	意　　味	処 理 内 容
0	正常終了.	
1000	N = 1 であった.	$\mathrm{E}(1) \leftarrow \mathrm{A}(1,1)$ とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
$5000 + i$	固有値を求める段階で収束しなかった. $(1 \leq i \leq \mathrm{N})$	$\mathrm{E}(1), \cdots, \mathrm{E}(i-1)$ にそれまでに求まった固有値が入る (ただし順不同).

(6) **注意事項**

(a) 配列 A には, 上三角部分のみにデータが格納されていればよい.

(b) 固有値は小さい順に格納される.

4.6.3 DCSMSS, RCSMSS
実対称行列の固有値・固有ベクトル

(1) **機 能**

実対称行列 A(2 次元配列型)(上三角型) の固有値をハウスホルダー法, 無平方根 QR 法またはバイセクション法により, 大きい方から m 個, または小さい方から m 個求め, それに対応する固有ベクトルを逆反復法により求める.

(2) **使用法**

倍精度サブルーチン:

CALL DCSMSS (A, LNA, N, EPS, E, M, VE, LNV, ISW, IW1, W1, IERR)

単精度サブルーチン:

CALL RCSMSS (A, LNA, N, EPS, E, M, VE, LNV, ISW, IW1, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\left\{\begin{array}{l}\text{32 ビット整数版では INTEGER(4)}\\ \text{64 ビット整数版では INTEGER(8)}\end{array}\right\}$

項番	引数名	型	大きさ	入出力	内容
1	A	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	LNA, N	入 力	実対称行列 A (2 次元配列型) (上三角型)
				出 力	入力時の内容は保存されない
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A の次数
4	EPS	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	1	入 力	固有値の収束判定のための絶対誤差の上限を与えるパラメータ (注意事項 (d) 参照)
5	E	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	M	出 力	固有値
6	M	I	1	入 力	求めたい固有値の個数 m
7	VE	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	LNV, M	出 力	各固有値に対応する固有ベクトル (列ベクトル)
8	LNV	I	1	入 力	配列 VE の整合寸法
9	ISW	I	1	入 力	処理スイッチ ISW $\geq$ 0: 大きい方から固有値を求める. ISW $<$ 0: 小さい方から固有値を求める.
10	IW1	I	M	出 力	固有ベクトルフラグ (注意事項 (e) 参照)
11	W1	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	$8 \times$ N	ワーク	作業領域
12	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \text{N} \leq \text{LNA}, \text{LNV}$

(b) $0 < \text{M} \leq \text{N}$

(5) **エラーインディケータ**

IERR の値	意　　　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	$E(1) \leftarrow A(1,1)$, $VE(1,1) \leftarrow 1.0$ とする.
2000	固有ベクトルを求める逆反復で最大反復回数をこえた.	固有ベクトルに一部精度の低いものがあるが, 処理は続行する (注意事項 (e) 参照).
3000	制限条件 (a) または (b) を満足しなかった.	処理を打ち切る.

(6) **注意事項**

(a) 配列 A には, 上三角部分のみにデータが格納されていればよい.

(b) 固有値は ISW≥0 のときには大きい順に, ISW<0 のときには小さい順に格納される.

(c) 固有値は無平方根 QR 法とバイセクション法を内部で適切に切り分け計算している.

(d) EPS ≤ 0 のときは内部で自動的に最適値を設定する. 通常は自動設定されるように負の値を設定されたい. なお, EPS はバイセクション法で固有値を求めるときに使用される.

(e) 逆反復法で最大反復回数をこえた場合 (IERR=2000 出力時) について

- IW1(i) = 0 の場合 :
  i 番目の固有ベクトル計算は正常終了している.
- IW1(i) $\neq$ 0 の場合 :
  i 番目の固有ベクトル計算は収束条件が満たされず, 固有ベクトルの精度は低い.
  この場合 IW1(i) は, 反復回数が設定される.

なお正常終了時 (IERR=0 出力時) は, IW1(i) ← 0 が設定される.

(f) 固有ベクトルは正規直交系である.

(g) 固有ベクトルを必要としないときは, 4.6.4 $\begin{Bmatrix}\text{DCSMSN}\\ \text{RCSMSN}\end{Bmatrix}$ を使用する.

(7) **使用例**

(a) 問 題

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

の固有値を小さい順に 3 個とそれに対応する固有ベクトルを求める.

(b) 入力データ

行列 A, LNA=11, N=6, EPS=−1.0, M=3, LNV=11, ISW=−1

(c) 主プログラム

```
      PROGRAM BCSMSS
! *** EXAMPLE OF DCSMSS ***
      IMPLICIT REAL(8) (A-H,O-Z)
      PARAMETER ( LNA = 11, LNV = 11 )
      DIMENSION A(LNA,LNA), E(LNA), VE(LNV,LNV), IW1(LNA), W1(8*LNA)
!
      READ(5,*) N, M
      DO 10 I=1, N
         READ(5,*) (A(I,J), J=I, N)
   10 CONTINUE
!
      WRITE(6,1000) N, M
      DO 20 I=1, N
         WRITE(6,1100) (A(J,I), J=1, I-1), (A(I,J), J=I, N)
   20 CONTINUE
!
      ISW = -1
      EPS = -1.0D0
!
      CALL DCSMSS(A,LNA,N,EPS,E,M,VE,LNV,ISW,IW1,W1,IERR)
!
      WRITE(6,1200) IERR
!
      DO 40 K=1, M-3, 4
         WRITE(6,1300) ('EIGENVALUE ', I=1, 4)
         WRITE(6,1400) (E(I), I=K, K+3)
         WRITE(6,1300) ('EIGENVECTOR', I=1, 4)
         DO 30 J=1, N
            WRITE(6,1500) (VE(J,I), I=K, K+3)
   30    CONTINUE
   40 CONTINUE
      IF(MOD(M,4).NE.0) THEN
         WRITE(6,1300) ('EIGENVALUE ', I=M/4*4+1, M)
         WRITE(6,1400) (E(I), I=M/4*4+1, M)
         WRITE(6,1300) ('EIGENVECTOR', I=M/4*4+1, M)
         DO 50 J=1, N
            WRITE(6,1500) (VE(J,I), I=M/4*4+1, M)
   50    CONTINUE
      ENDIF
      STOP
!
 1000 FORMAT(' ',/,/,&
             ' ***  DCSMSS  ***',/,/,&
             '  **  INPUT  **',/,/,&
             '      N = ', I2,/,/,&
             '      M = ', I2,/,/,&
             '      INPUT MATRIX A',/)
 1100 FORMAT(7X, 11(F7.1))
 1200 FORMAT(' ',/,/,&
             '  **  OUTPUT  **',/,/,&
             '      IERR = ', I4)
 1300 FORMAT(' ',/,1X, 4(5X, A11, 2X))
 1400 FORMAT(3X, 4(2X, 1PD14.7, 2X))
 1500 FORMAT(2X, 4(F14.8, 4X))
      END
```

(d) 出力結果

```
***  DCSMSS  ***

 **  INPUT  **

     N =  6

     M =  3

     INPUT MATRIX A

          0.0    1.0    0.0    0.0    0.0    1.0
          1.0    0.0    1.0    0.0    0.0    0.0
          0.0    1.0    0.0    1.0    0.0    0.0
          0.0    0.0    1.0    0.0    1.0    0.0
          0.0    0.0    0.0    1.0    0.0    1.0
          1.0    0.0    0.0    0.0    1.0    0.0


 **  OUTPUT  **

     IERR =    0

     EIGENVALUE        EIGENVALUE        EIGENVALUE
    -2.0000000D+00    -1.0000000D+00    -1.0000000D+00

     EIGENVECTOR       EIGENVECTOR       EIGENVECTOR
     0.40824829        0.00000000        0.57735027
    -0.40824829       -0.50000000       -0.28867513
     0.40824829        0.50000000       -0.28867513
    -0.40824829       -0.00000000        0.57735027
     0.40824829       -0.50000000       -0.28867513
    -0.40824829        0.50000000       -0.28867513
```

4.6.4 DCSMSN, RCSMSN
実対称行列の固有値

(1) **機 能**

実対称行列 A(2 次元配列型)(上三角型) の固有値をハウスホルダー法, 無平方根 QR 法またはバイセクション法により, 大きい方から m 個, または小さい方から m 個求める.

(2) **使用法**

倍精度サブルーチン:

CALL DCSMSN (A, LNA, N, EPS, E, M, ISW, W1, IERR)

単精度サブルーチン:

CALL RCSMSN (A, LNA, N, EPS, E, M, ISW, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: { 32 ビット整数版では INTEGER(4)
64 ビット整数版では INTEGER(8) }

項番	引数名	型	大きさ	入出力	内　容
1	A	D R	LNA, N	入 力	実対称行列 A (2 次元配列型) (上三角型)
				出 力	入力時の内容は保存されない
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A の次数
4	EPS	D R	1	入 力	固有値の収束判定のための絶対誤差の上限を与えるパラメータ (注意事項 (d) 参照)
5	E	D R	M	出 力	固有値
6	M	I	1	入 力	求めたい固有値の個数 m
7	ISW	I	1	入 力	処理スイッチ ISW $\geq$ 0: 大きい方から固有値を求める. ISW $<$ 0: 小さい方から固有値を求める.
8	W1	D R	5 × N	ワーク	作業領域
9	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNA}$

(b) $0 < \mathrm{M} \leq \mathrm{N}$

(5) エラーインディケータ

IERR の値	意　　　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	$E(1) \leftarrow A(1,1)$ とする.
3000	制限条件 (a) または (b) を満足しなかった.	処理を打ち切る.

(6) 注意事項

(a) 配列 A には, 上三角部分のみにデータが格納されていればよい.

(b) 固有値は ISW≥0 のときには大きい順に, ISW<0 のときには小さい順に格納される.

(c) 固有値は無平方根 QR 法とバイセクション法を内部で適切に切り分けて計算している.

(d) EPS ≤ 0 のときは内部で自動的に最適値を設定する. 通常は自動設定されるように負の値を設定されたい. なお, EPS はバイセクション法で固有値を求めるときに使用される.

4.6.5 DCSMEE, RCSMEE
実対称行列の固有値・固有ベクトル (区間指定)

(1) **機 能**

実対称行列 A(2 次元配列型)(上三角型) の指定した区間の固有値をハウスホルダー法, バイセクション法により, 昇順で m 個, もしくは降順で m 個求め, それに対応する固有ベクトルを逆反復法により求める.

(2) **使用法**

倍精度サブルーチン:

CALL DCSMEE (A, LNA, N, EPS, E, M, E1, E2, VE, LNV, IW1, W1, IERR)

単精度サブルーチン:

CALL RCSMEE (A, LNA, N, EPS, E, M, E1, E2, VE, LNV, IW1, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: { 32 ビット整数版では INTEGER(4)
64 ビット整数版では INTEGER(8) }

項番	引数名	型	大きさ	入出力	内　容
1	A	D R	LNA, N	入 力	実対称行列 A (2 次元配列型) (上三角型)
				出 力	入力時の内容は保存されない
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A の次数
4	EPS	D R	1	入 力	固有値の収束判定のための絶対誤差の上限を与えるパラメータ (注意事項 (b) 参照)
5	E	D R	M	出 力	固有値
6	M	I	1	入 力	求めたい固有値の個数 m
				出 力	求めた固有値の個数
7	E1	D R	1	入 力	E1 < E2: E1 から昇順で固有値を求める. (E2 は, 上限) E1 > E2: E1 から降順で固有値を求める. (E2 は, 下限) (注意事項 (c)(d) 参照)
8	E2	D R	1	入 力	
9	VE	D R	LNV, M	出 力	各固有値に対応する固有ベクトル (列ベクトル)
10	LNV	I	1	入 力	配列 VE の整合寸法
11	IW1	I	M	出 力	固有ベクトルフラグ (注意事項 (e) 参照)
12	W1	D R	8 × N	ワーク	作業領域
13	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNA}, \mathrm{LNV}$

(b) $0 < \mathrm{M} \leq \mathrm{N}$

(5) **エラーインディケータ**

IERR の値	意　　　　　味	処　理　内　容
0	正常終了.	
1000	$N = 1$ であった.	$E(1) \leftarrow A(1,1)$, $VE(1,1) \leftarrow 1.0$ とする.
1500	区間 [E1, E2] に存在する固有値は M 個未満であった.	区間 [E1, E2] 内にある全ての固有値, 固有ベクトルを求める. 引数 M には求めた固有値の個数を出力する.
2000	固有ベクトルを求める逆反復で最大反復回数をこえた.	固有ベクトルに一部精度の低いものがあるが, 処理は続行する (注意事項 (e) 参照).
3000	制限条件 (a) または (b) を満足しなかった.	処理を打ち切る.

(6) **注意事項**

(a) 配列 A には, 上三角部分のみにデータが格納されていればよい.

(b) $EPS \leq 0$ のときは内部で自動的に最適値を設定する. 通常は自動設定されるように負の値を設定されたい. なお, EPS はバイセクション法で固有値を求めるときに使用される.

(c) 固有値は E1<E2 のときには小さい順, E1>E2 のときには大きい順に格納される.

(d) E1=E2 のとき, 区間 [E1-EPS, E1+EPS] にある固有値が求まる. しかし, 通常は E1≠E2 となるように設定されたい.

(e) 逆反復法で最大反復回数をこえた場合 (IERR=2000 出力時) について

- $IW1(i) = 0$ の場合 :
  i 番目の固有ベクトル計算は正常終了している.
- $IW1(i) \neq 0$ の場合 :
  i 番目の固有ベクトル計算は収束条件が満たされず, 固有ベクトルの精度は低い.
  この場合 $IW1(i)$ は, 反復回数が設定される.

なお正常終了時 (IERR=0 出力時) は, $IW1(i) \leftarrow 0$ が設定される.

(f) 固有ベクトルは正規直交系である.

(g) 固有ベクトルを必要としないときは, 4.6.6 $\left\{\begin{matrix}\text{DCSMEN}\\\text{RCSMEN}\end{matrix}\right\}$ を使用する.

(7) **使用例**

(a) 問 題

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

の固有値を区間 [0, 5] で 3 個求め (昇順), それに対応する固有ベクトルを求める.

(b) 入力データ

行列 A, LNA=11, N=6, EPS=−1.0, M=3, E1=0, E2=5, LNV=11

(c) 主プログラム

```
      PROGRAM BCSMEE
! *** EXAMPLE OF DCSMEE ***
      IMPLICIT REAL(8) (A-H,O-Z)
      PARAMETER ( LNA = 11, LNV = 11 )
      DIMENSION A(LNA,LNA), E(LNA), VE(LNV,LNV), IW1(LNA), W1(8*LNA)
!
      READ(5,*) N, M, E1, E2
      DO 10 I=1, N
         READ(5,*) (A(I,J), J=I, N)
   10 CONTINUE
!
      WRITE(6,1000) N, M, E1, E2
      DO 20 I=1, N
         WRITE(6,1100) (A(J,I), J=1, I-1), (A(I,J), J=I, N)
   20 CONTINUE
!
      EPS = -1.0D0
!
      CALL DCSMEE(A,LNA,N,EPS,E,M,E1,E2,VE,LNV,IW1,W1,IERR)
!
      WRITE(6,1200) IERR
!
      DO 40 K=1, M-3, 4
         WRITE(6,1300) ('EIGENVALUE ', I=1, 4)
         WRITE(6,1400) (E(I), I=K, K+3)
         WRITE(6,1300) ('EIGENVECTOR', I=1, 4)
         DO 30 J=1, N
            WRITE(6,1500) (VE(J,I), I=K, K+3)
   30    CONTINUE
   40 CONTINUE
      IF(MOD(M,4).NE.0) THEN
         WRITE(6,1300) ('EIGENVALUE ', I=M/4*4+1, M)
         WRITE(6,1400) (E(I), I=M/4*4+1, M)
         WRITE(6,1300) ('EIGENVECTOR', I=M/4*4+1, M)
         DO 50 J=1, N
            WRITE(6,1500) (VE(J,I), I=M/4*4+1, M)
   50    CONTINUE
      ENDIF
      STOP
!
 1000 FORMAT(1X,/,/,&
             1X,'***  DCSMEE  ***',/,/,&
             1X,' **  INPUT  **',/,/,&
             1X,'N = ', I4,'  M = ', I4,/,/,&
             1X,'E1= ', 1PD14.7,'  E2= ', 1PD14.7,/,/,&
             1X,'     INPUT MATRIX A',/)
 1100 FORMAT(1X, 6X, 11(F7.1))
 1200 FORMAT(1X,/,/,&
             1X,' **  OUTPUT  **',/,/,&
             1X,'     IERR = ', I4)
 1300 FORMAT(1X,/,1X, 4(5X, A11, 2X))
 1400 FORMAT(1X, 2X, 4(2X, 1PD14.7, 2X))
 1500 FORMAT(1X, 1X, 4(F14.8, 4X))
      END
```

(d) 出力結果

```
***  DCSMEE  ***

 **  INPUT  **

N =    6  M =    3

E1=  0.0000000D+00  E2=  5.0000000D+00

     INPUT MATRIX A

          0.0    1.0    0.0    0.0    0.0    1.0
          1.0    0.0    1.0    0.0    0.0    0.0
```

```
          0.0    1.0    0.0    1.0    0.0    0.0
          0.0    0.0    1.0    0.0    1.0    0.0
          0.0    0.0    0.0    1.0    0.0    1.0
          1.0    0.0    0.0    0.0    1.0    0.0

 **  OUTPUT  **

     IERR =    0

     EIGENVALUE          EIGENVALUE          EIGENVALUE
     1.0000000D+00       1.0000000D+00       2.0000000D+00

     EIGENVECTOR         EIGENVECTOR         EIGENVECTOR
     0.57735027          0.00000000         -0.40824829
     0.28867513         -0.50000000         -0.40824829
    -0.28867513         -0.50000000         -0.40824829
    -0.57735027         -0.00000000         -0.40824829
    -0.28867513          0.50000000         -0.40824829
     0.28867513          0.50000000         -0.40824829
```

4.6.6 DCSMEN, RCSMEN
実対称行列の固有値 (区間指定)

(1) **機 能**

実対称行列A(2次元配列型)(上三角型)の指定した区間の固有値をハウスホルダー法, バイセクション法により, 昇順でm個, もしくは降順でm個求める.

(2) **使用法**

倍精度サブルーチン:
CALL DCSMEN (A, LNA, N, EPS, E, M, E1, E2, W1, IERR)
単精度サブルーチン:
CALL RCSMEN (A, LNA, N, EPS, E, M, E1, E2, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$

項番	引数名	型	大きさ	入出力	内容
1	A	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	LNA, N	入 力	実対称行列A(2次元配列型)(上三角型)
				出 力	入力時の内容は保存されない
2	LNA	I	1	入 力	配列Aの整合寸法
3	N	I	1	入 力	行列Aの次数
4	EPS	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	1	入 力	固有値の収束判定のための絶対誤差の上限を与えるパラメータ (注意事項 (b) 参照)
5	E	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	M	出 力	固有値
6	M	I	1	入 力	求めたい固有値の個数m
				出 力	求めた固有値の個数
7	E1	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	1	入 力	E1 < E2の場合: E1から昇順で固有値を求める. (E2は, 上限) E1 > E2の場合: E1から降順で固有値を求める. (E2は, 下限) (注意事項 (c)(d) 参照)
8	E2	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	1	入 力	
9	W1	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	5×N	ワーク	作業領域
10	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNA}$

(b) $0 < \mathrm{M} \leq \mathrm{N}$

(5) **エラーインディケータ**

IERR の値	意　　　　　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	$E(1) \leftarrow A(1,1)$ とする.
1500	区間 [E1, E2] に存在する固有値は M 個未満であった.	区間 [E1, E2] 内にある全ての固有値, 固有ベクトルを求める. 引数 M には求めた固有値の個数を出力する.
3000	制限条件 (a) または (b) を満足しなかった.	処理を打ち切る.

(6) **注意事項**

(a) 配列 A には, 上三角部分のみにデータが格納されていればよい.

(b) EPS $\leq$ 0 のときは内部で自動的に最適値を設定する. 通常は自動設定されるように負の値を設定されたい. なお, EPS はバイセクション法で固有値を求めるときに使用される.

(c) 固有値は E1<E2 のときには小さい順, E1>E2 のときには大きい順に格納される.

(d) E1=E2 のとき, 区間 [E1-EPS, E1+EPS] にある固有値が求まる. しかし, 通常は E1$\neq$E2 となるように設定されたい.

4.7 エルミート行列 (2次元配列型) (上三角型) (実数引数型)

4.7.1 ZCHRAA, CCHRAA
エルミート行列の全固有値・全固有ベクトル

(1) **機能**

エルミート行列 $A = (\mathrm{AR}, \mathrm{AI})$ (2次元配列型) (上三角型) (実数引数型) の全固有値とそれに対応する全固有ベクトルをハウスホルダー法, QR法により求める.

(2) **使用法**

倍精度サブルーチン:
　CALL ZCHRAA (AR, AI, LNA, N, E, VR, VI, LNV, W1, IERR)
単精度サブルーチン:
　CALL CCHRAA (AR, AI, LNA, N, E, VR, VI, LNV, W1, IERR)

(3) **引数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: $\left\{\begin{matrix}\text{32ビット整数版では INTEGER(4)}\\\text{64ビット整数版では INTEGER(8)}\end{matrix}\right\}$

項番	引数名	型	大きさ	入出力	内容
1	AR	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	LNA, N	入力	エルミート行列 A の実部 (2次元配列型) (上三角型)
2	AI	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	LNA, N	入力	エルミート行列 A の虚部 (2次元配列型) (上三角型)
3	LNA	I	1	入力	配列 AR, AI の整合寸法
4	N	I	1	入力	行列 A の次数
5	E	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	N	出力	固有値
6	VR	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	LNV, N	出力	各固有値に対応する固有ベクトルの実部 (列ベクトル)
7	VI	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	LNV, N	出力	各固有値に対応する固有ベクトルの虚部 (列ベクトル)
8	LNV	I	1	入力	配列 VR, VI の整合寸法
9	W1	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	$3 \times \mathrm{N}$	ワーク	作業領域
10	IERR	I	1	出力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNA}, \mathrm{LNV}$

(5) エラーインディケータ

IERR の値	意　　味	処 理 内 容
0	正常終了.	
1000	N = 1 であった.	$E(1) \leftarrow A(1,1)$, $VR(1,1) \leftarrow 1.0$, $VI(1,1) \leftarrow 0.0$ とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
$5000+i$	固有値を求める段階で収束しなかった. $(1 \leq i \leq \mathrm{N})$	$E(1), \cdots, E(i-1)$ にそれまでに求まった固有値が入る (ただし順不同). このとき固有ベクトルは求まらない.

(6) 注意事項

(a) 配列 AR, AI には, エルミート行列の実部, 虚部のそれぞれ上三角部分のみが格納されていればよい (付録 B 参照).

(b) 固有値は小さい順に格納される.

(c) 固有ベクトルは正規直交系である.

(d) 固有ベクトルを必要としないときは, 4.7.2 $\left\{\begin{matrix}\text{ZCHRAN}\\ \text{CCHRAN}\end{matrix}\right\}$を使用する.

(7) 使用例

(a) 問 題

$$A = \begin{bmatrix} 7 & 3 & 1+2i & -1+2i \\ 3 & 7 & 1-2i & -1-2i \\ 1-2i & 1+2i & 7 & -3 \\ -1-2i & -1+2i & -3 & 7 \end{bmatrix}$$

の全固有値とそれに対応する固有ベクトルを求める.

(b) 入力データ

行列 A の実部 AR, 虚部 AI, LNA=11, N=4, LNV=10

(c) 主プログラム

```
      PROGRAM ACHRAA
! *** EXAMPLE OF ZCHRAA ***
      IMPLICIT REAL(8) (A-H,O-Z)
      CHARACTER*80 FMT
      PARAMETER ( LNA = 11, LNV = 11 )
      DIMENSION AR(LNA,LNA), AI(LNA,LNA), E(LNA),&
                VR(LNV,LNV), VI(LNV,LNV), W1(3*LNA)
!
      READ(5,*) N
      DO 10 I=1, N
         READ(5,*) (AR(I,J), AI(I,J), J=I, N)
   10 CONTINUE
!
      WRITE(6,1000) N
      DO 20 I=1, N
         WRITE(FMT,1100) N
         WRITE(6,FMT) (AR(J,I), -AI(J,I), J=1, I-1),&
                      (AR(I,J),  AI(I,J), J=I, N)
   20 CONTINUE
!
      CALL ZCHRAA(AR,AI,LNA,N,E,VR,VI,LNV,W1,IERR)
!
      WRITE(6,1200) IERR
!
      DO 40 J=1, N-1, 2
         WRITE(6,1300) ('EIGENVALUE ', I=1, 2)
```

```
         WRITE(6,1400) E(J), E(J+1)
         WRITE(6,1300) ('EIGENVECTOR', I=1, 2)
         DO 30 I=1, N
            WRITE(6,1500) VR(I,J), VI(I,J), VR(I,J+1), VI(I,J+1)
   30    CONTINUE
   40 CONTINUE
      IF(MOD(N,2).NE.0) THEN
         WRITE(6,1300) 'EIGENVALUE '
         WRITE(6,1400) E(N)
         WRITE(6,1300) 'EIGENVECTOR'
         DO 50 I=1, N
            WRITE(6,1500) VR(I,N), VI(I,N)
   50    CONTINUE
      ENDIF
      STOP
!
 1000 FORMAT(' ',/,/,&
             ' ***  ZCHRAA  ***',/,/,&
             '  **  INPUT  **',/,/,&
             '      N = ', I4,/,/,&
             '      INPUT MATRIX A ( REAL,IMAGINARY )',/)
 1100 FORMAT('( ,5X,', I2,'(''('',F5.1,'' ,'',F5.1,'') ''))')
 1200 FORMAT(' ',/,/,&
             '  **  OUTPUT  **',/,/,&
             '      IERR = ', I4)
 1300 FORMAT(' ',/, 2(14X, A11, 8X))
 1400 FORMAT(' ', 2(12X, 1PD14.7, 7X))
 1500 FORMAT(' ', 2(5X, F12.8, ' ,', F12.8, 2X))
      END
```

(d) 出力結果

```
***  ZCHRAA  ***

 **  INPUT  **

     N =    4

     INPUT MATRIX A ( REAL,IMAGINARY )

    (  7.0 ,  0.0) (  3.0 ,  0.0) (  1.0 ,  2.0) ( -1.0 ,  2.0)
    (  3.0 ,  0.0) (  7.0 ,  0.0) (  1.0 , -2.0) ( -1.0 , -2.0)
    (  1.0 , -2.0) (  1.0 ,  2.0) (  7.0 ,  0.0) ( -3.0 ,  0.0)
    ( -1.0 , -2.0) ( -1.0 ,  2.0) ( -3.0 ,  0.0) (  7.0 ,  0.0)


 **  OUTPUT  **

     IERR =    0

             EIGENVALUE                       EIGENVALUE
             0.0000000D+00                    8.0000000D+00

             EIGENVECTOR                      EIGENVECTOR
       0.50000000 ,  0.00000000        -0.70710678 , -0.00000000
      -0.50000000 ,  0.00000000         0.00000000 , -0.00000000
       0.00000000 ,  0.50000000         0.35355339 ,  0.35355339
      -0.00000000 ,  0.50000000        -0.35355339 ,  0.35355339

             EIGENVALUE                       EIGENVALUE
             8.0000000D+00                    1.2000000D+01

             EIGENVECTOR                      EIGENVECTOR
       0.00000000 ,  0.00000000         0.50000000 ,  0.00000000
      -0.09987868 ,  0.70001732         0.50000000 ,  0.00000000
      -0.30006932 , -0.39994800         0.50000000 , -0.00000000
      -0.39994800 ,  0.30006932        -0.50000000 , -0.00000000
```

4.7.2 ZCHRAN, CCHRAN
エルミート行列の全固有値

(1) **機 能**

エルミート行列 $A = (\mathrm{AR}, \mathrm{AI})$ (2 次元配列型) (上三角型) (実数引数型) の全固有値をハウスホルダー法, 無平方根 QR 法により求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZCHRAN (AR, AI, LNA, N, E, W1, IERR)

単精度サブルーチン:

CALL CCHRAN (AR, AI, LNA, N, E, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: { 32 ビット整数版では INTEGER(4)
　 64 ビット整数版では INTEGER(8) }

項番	引数名	型	大きさ	入出力	内　　容
1	AR	D R	LNA, N	入 力	エルミート行列 A の実部 (2 次元配列型) (上三角型)
				出 力	入力時の内容は保存されない
2	AI	D R	LNA, N	入 力	エルミート行列 A の虚部 (2 次元配列型) (上三角型)
				出 力	入力時の内容は保存されない
3	LNA	I	1	入 力	配列 AR, AI の整合寸法
4	N	I	1	入 力	行列 A の次数
5	E	D R	N	出 力	固有値
6	W1	D R	3×N	ワーク	作業領域
7	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNA}$

(5) **エラーインディケータ**

IERR の値	意　　味	処　理　内　容
0	正常終了.	
1000	$\mathrm{N} = 1$ であった.	$\mathrm{E}(1) \leftarrow \mathrm{AR}(1,1)$ とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
$5000 + i$	固有値を求める段階で収束しなかった. $(1 \leq i \leq \mathrm{N})$	$\mathrm{E}(1), \cdots, \mathrm{E}(i-1)$ にそれまでに求まった固有値が入る (ただし順不同).

(6) **注意事項**

(a) 配列 AR, AI には, エルミート行列の実部, 虚部のそれぞれ上三角部分のみが格納されていればよい (付録 B 参照).

(b) 固有値は小さい順に格納される.

4.7.3 ZCHRSS, CCHRSS
エルミート行列の固有値・固有ベクトル

(1) **機 能**

エルミート行列 $A = (\mathrm{AR}, \mathrm{AI})$ (2 次元配列型) (上三角型) (実数引数型) の全固有値をハウスホルダー法, 無平方根 QR 法またはバイセクション法により, 大きい方から m 個, または小さい方から m 個求め, それに対応する固有ベクトルを逆反復法により求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZCHRSS (AR, AI, LNA, N, EPS, E, M, VR, VI, LNV, ISW, IW1, W1, IERR)

単精度サブルーチン:

CALL CCHRSS (AR, AI, LNA, N, EPS, E, M, VR, VI, LNV, ISW, IW1, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\left\{\begin{array}{l}\text{32 ビット整数版では INTEGER(4)}\\ \text{64 ビット整数版では INTEGER(8)}\end{array}\right\}$

項番	引数名	型	大きさ	入出力	内　容
1	AR	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	LNA, N	入 力	エルミート行列 A の実部 (2 次元配列型) (上三角型)
				出 力	入力時の内容は保存されない
2	AI	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	LNA, N	入 力	エルミート行列 A の虚部 (2 次元配列型) (上三角型)
				出 力	入力時の内容は保存されない
3	LNA	I	1	入 力	配列 AR, AI の整合寸法
4	N	I	1	入 力	行列 A の次数
5	EPS	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	1	入 力	固有値の収束判定のための絶対誤差の上限を与えるパラメータ (注意事項 (d) 参照)
6	E	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	M	出 力	固有値
7	M	I	1	入 力	求めたい固有値の個数 m
8	VR	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	LNV, M	出 力	各固有値に対応する固有ベクトルの実部 (列ベクトル)
9	VI	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	LNV, M	出 力	各固有値に対応する固有ベクトルの虚部 (列ベクトル)
10	LNV	I	1	入 力	配列 VR, VI の整合寸法
11	ISW	I	1	入 力	処理スイッチ ISW ≥ 0: 大きい方から固有値を求める. ISW < 0: 小さい方から固有値を求める.
12	IW1	I	M	出 力	固有ベクトルフラグ (注意事項 (e) 参照)
13	W1	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	$10 \times \mathrm{N}$	ワーク	作業領域
14	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a) $0 < \mathrm{N} \leq \mathrm{LNA}, \mathrm{LNV}$

(b) $0 < \mathrm{M} \leq \mathrm{N}$

(5) エラーインディケータ

IERR の値	意　　　　味	処　理　内　容
0	正常終了.	
1000	$\mathrm{N} = 1$ であった.	$\mathrm{E}(1) \leftarrow \mathrm{AR}(1,1)$, $\mathrm{VR}(1,1) \leftarrow 1.0$, $\mathrm{VI}(1,1) \leftarrow 0.0$ とする.
2000	固有ベクトルを求める逆反復で最大反復回数をこえた.	固有ベクトルに一部精度の低いものがあるが, 処理は続行する (注意事項 (e) 参照).
3000	制限条件 (a) または (b) を満足しなかった.	処理を打ち切る.

(6) 注意事項

(a) 配列 AR, AI には, エルミート行列の実部, 虚部のそれぞれ上三角部分のみが格納されていればよい (付録 B 参照).

(b) 固有値は ISW≥0 のときには大きい順に, ISW<0 のときには小さい順に格納される.

(c) 固有値は無平方根 QR 法とバイセクション法を内部で適切に切り分けて計算している.

(d) $\mathrm{EPS} \leq 0$ のときは内部で自動的に最適値を設定する. 通常は自動設定されるように負の値を設定されたい. なお, EPS はバイセクション法で固有値を求めるときに使用される.

(e) 逆反復法で最大反復回数をこえた場合 (IERR=2000 出力時) について

- $\mathrm{IW1}(i) = 0$ の場合 :
  i 番目の固有ベクトル計算は正常終了している.
- $\mathrm{IW1}(i) \neq 0$ の場合 :
  i 番目の固有ベクトル計算は収束条件が満たされず, 固有ベクトルの精度は低い.
  この場合 $\mathrm{IW1}(i)$ は, 反復回数が設定される.

なお正常終了時 (IERR=0 出力時) は, $\mathrm{IW1}(i) \leftarrow 0$ が設定される.

(f) 固有ベクトルは正規直交系である.

(g) 固有ベクトルを必要としないときは, 4.7.4 $\left\{\begin{matrix}\mathrm{ZCHRSN}\\ \mathrm{CCHRSN}\end{matrix}\right\}$ を使用する.

(7) 使用例

(a) 問 題

$$A = \begin{bmatrix} 7 & 3 & 1+2i & -1+2i \\ 3 & 7 & 1-2i & -1-2i \\ 1-2i & 1+2i & 7 & -3 \\ -1-2i & -1+2i & -3 & 7 \end{bmatrix}$$

のとき, エルミート行列 A の固有値を大きい順に 3 個とそれに対応する固有ベクトルを求める.

(b) 入力データ

行列 A の実部 AR, 虚部 AI, LNA=11, N=4, EPS=−1.0, M=3, LNV=11, ISW=1

(c) 主プログラム

```
      PROGRAM ACHRSS
! *** EXAMPLE OF ZCHRSS ***
      IMPLICIT REAL(8) (A-H,O-Z)
      CHARACTER*80 FMT
      PARAMETER ( LNA = 11, LNV = 11 )
      DIMENSION AR(LNA,LNA), AI(LNA,LNA), E(LNA),&
                VR(LNV,LNV), VI(LNV,LNV), IW1(LNA), W1(10*LNA)
!
      READ(5,*) N, M
      DO 10 I=1, N
         READ(5,*) (AR(I,J), AI(I,J), J=I, N)
   10 CONTINUE
!
      WRITE(6,1000) N, M
      DO 20 I=1, N
         WRITE(FMT,1100) N
         WRITE(6,FMT) (AR(J,I), -AI(J,I), J=1, I-1),&
                      (AR(I,J),  AI(I,J), J=I, N)
   20 CONTINUE
!
      ISW = 1
      EPS = -1.0D0
!
      CALL ZCHRSS(AR,AI,LNA,N,EPS,E,M,VR,VI,LNV,ISW,IW1,W1,IERR)
!
      WRITE(6,1200) IERR
!
      DO 40 J=1, M-1, 2
         WRITE(6,1300) ('EIGENVALUE ', I=1, 2)
         WRITE(6,1400) E(J), E(J+1)
         WRITE(6,1300) ('EIGENVECTOR', I=1, 2)
         DO 30 I=1, N
            WRITE(6,1500) VR(I,J), VI(I,J), VR(I,J+1), VI(I,J+1)
   30    CONTINUE
   40 CONTINUE
      IF(MOD(M,2).NE.0) THEN
         WRITE(6,1300) 'EIGENVALUE '
         WRITE(6,1400) E(M)
         WRITE(6,1300) 'EIGENVECTOR'
         DO 50 I=1, N
            WRITE(6,1500) VR(I,M), VI(I,M)
   50    CONTINUE
      ENDIF
      STOP
!
 1000 FORMAT(' ',/,/,&
             ' ***  ZCHRSS  ***',/,/,&
             '  **  INPUT  **',/,/,&
             '      N = ', I4,/,/,&
             '      M = ', I4,/,/,&
             '      INPUT MATRIX A ( REAL,IMAGINARY )',/)
 1100 FORMAT('( ,5X,', I2,'(''('',F5.1,'' ,'',F5.1,'') ''))')
 1200 FORMAT(' ',/,/,&
             '  **  OUTPUT  **',/,/,&
             '      IERR = ', I4)
 1300 FORMAT(' ',/, 2(14X, A11, 8X))
 1400 FORMAT(' ', 2(12X, 1PD14.7, 7X))
 1500 FORMAT(' ', 2(5X, F12.8, ' ,', F12.8, 2X))
      END
```

(d) 出力結果

```
***  ZCHRSS  ***

 **  INPUT  **

     N =    4

     M =    3

     INPUT MATRIX A ( REAL,IMAGINARY )

    (  7.0 ,  0.0) (  3.0 ,  0.0) (  1.0 ,  2.0) ( -1.0 ,  2.0)
    (  3.0 ,  0.0) (  7.0 ,  0.0) (  1.0 , -2.0) ( -1.0 , -2.0)
    (  1.0 , -2.0) (  1.0 ,  2.0) (  7.0 ,  0.0) ( -3.0 ,  0.0)
    ( -1.0 , -2.0) ( -1.0 ,  2.0) ( -3.0 ,  0.0) (  7.0 ,  0.0)


 **  OUTPUT  **

     IERR =    0

             EIGENVALUE                       EIGENVALUE
             1.2000000D+01                    8.0000000D+00

             EIGENVECTOR                      EIGENVECTOR
      0.50000000 ,  0.00000000         0.00000000 ,  0.00000000
      0.50000000 ,  0.00000000        -0.09987868 ,  0.70001732
      0.50000000 , -0.00000000        -0.30006932 , -0.39994800
     -0.50000000 , -0.00000000        -0.39994800 ,  0.30006932
```

```
        EIGENVALUE
        8.0000000D+00

        EIGENVECTOR
  0.70710678 ,  0.00000000
 -0.00000000 , -0.00000000
 -0.35355339 , -0.35355339
  0.35355339 , -0.35355339
```

4.7.4 ZCHRSN, CCHRSN
エルミート行列の固有値

(1) **機 能**

エルミート行列 $A = (\mathrm{AR}, \mathrm{AI})$ (2 次元配列型) (上三角型) (実数引数型) の固有値をハウスホルダー法, 無平方根 QR 法またはバイセクション法により, 大きい方から m 個, または小さい方から m 個求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZCHRSN (AR, AI, LNA, N, EPS, E, M, ISW, W1, IERR)

単精度サブルーチン:

CALL CCHRSN (AR, AI, LNA, N, EPS, E, M, ISW, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\left\{\begin{array}{l}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{array}\right\}$

項番	引数名	型	大きさ	入出力	内　容
1	AR	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	LNA, N	入 力	エルミート行列 A の実部 (2 次元配列型) (上三角型)
				出 力	入力時の内容は保存されない
2	AI	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	LNA, N	入 力	エルミート行列 A の虚部 (2 次元配列型) (上三角型)
				出 力	入力時の内容は保存されない
3	LNA	I	1	入 力	配列 AR, AI の整合寸法
4	N	I	1	入 力	行列 A の次数
5	EPS	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	1	入 力	固有値の収束判定のための絶対誤差の上限を与えるパラメータ (注意事項 (d) 参照)
6	E	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	M	出 力	固有値
7	M	I	1	入 力	求めたい固有値の個数 m
8	ISW	I	1	入 力	処理スイッチ ISW $\geq$ 0: 大きい方から固有値を求める. ISW $<$ 0: 小さい方から固有値を求める.
9	W1	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	5×N	ワーク	作業領域
10	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNA}$

(b) $0 < \mathrm{M} \leq \mathrm{N}$

(5) エラーインディケータ

IERR の値	意　　　　味	処 理 内 容
0	正常終了.	
1000	N = 1 であった.	E(1) ← AR(1, 1) とする.
3000	制限条件 (a) または (b) を満足しなかった.	処理を打ち切る.

(6) 注意事項

(a) 配列 AR, AI には, エルミート行列の実部, 虚部のそれぞれ上三角部分のみが格納されていればよい (付録 B 参照).

(b) 固有値は ISW$\geq$0 のときには大きい順に, ISW$<$0 のときには小さい順に格納される.

(c) 固有値は平方根 QR 法とバイセクション法を内部で適切に切り分けて計算している.

(d) EPS ≤ 0 のときは内部で自動的に最適値を設定する. 通常は自動設定されるように負の値を設定されたい. なお, EPS はバイセクション法で固有値を求めるときに使用される.

4.7.5 ZCHREE, CCHREE
エルミート行列の固有値・固有ベクトル (区間指定)

(1) **機 能**

エルミート行列 $A = (\mathrm{AR}, \mathrm{AI})$(2 次元配列型)(上三角型)(実数引数型) の指定した区間の固有値をハウスホルダー法, バイセクション法により, 昇順で m 個, もしくは降順で m 個求め, それに対応する固有ベクトルを逆反復法により求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZCHREE (AR, AI, LNA, N, EPS, E, M, E1, E2, VR, VI, LNV, IW1, W1, IERR)

単精度サブルーチン:

CALL CCHREE (AR, AI, LNA, N, EPS, E, M, E1, E2, VR, VI, LNV, IW1, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型　I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\ \text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$
R:単精度実数型　C:単精度複素数型

項番	引数名	型	大きさ	入出力	内　　容
1	AR	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	LNA, N	入 力	エルミート行列 A の実部 (2 次元配列型) (上三角型)
				出 力	入力時の内容は保存されない
2	AI	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	LNA, N	入 力	エルミート行列 A の虚部 (2 次元配列型) (上三角型)
				出 力	入力時の内容は保存されない
3	LNA	I	1	入 力	配列 AR, AI の整合寸法
4	N	I	1	入 力	行列 A の次数
5	EPS	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	1	入 力	固有値の収束判定のための絶対誤差の上限を与えるパラメータ (注意事項 (b) 参照)
6	E	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	M	出 力	固有値
7	M	I	1	入 力	求めたい固有値の個数 m
				出 力	求めた固有値の個数
8	E1	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	1	入 力	E1 $<$ E2 の場合：E1 から昇順で固有値を求める. (E2 は, 上限) E1 $>$ E2 の場合：E1 から降順で固有値を求める. (E2 は, 下限) (注意事項 (c)(d) 参照)
9	E2	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	1	入 力	
10	VR	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	LNV, M	出 力	各固有値に対応する固有ベクトルの実部 (列ベクトル)
11	VI	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	LNV, M	出 力	各固有値に対応する固有ベクトルの虚部 (列ベクトル)
12	LNV	I	1	入 力	配列 VR, VI の整合寸法
13	IW1	I	M	出 力	固有ベクトルフラグ (注意事項 (e) 参照)
14	W1	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	10 × N	ワーク	作業領域
15	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNA}, \mathrm{LNV}$

(b) $0 < \mathrm{M} \leq \mathrm{N}$

(5) **エラーインディケータ**

IERR の値	意　　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	$E(1) \leftarrow AR(1,1)$, $VR(1,1) \leftarrow 1.0$, $VI(1,1) \leftarrow 0.0$ とする.
1500	区間 [E1, E2] に存在する固有値は M 個未満であった.	区間 [E1, E2] 内にある全ての固有値, 固有ベクトルを求める. 引数 M には求めた固有値の個数を出力する.
2000	固有ベクトルを求める逆反復で最大反復回数をこえた.	固有ベクトルに一部精度の低いものがあるが, 処理は続行する (注意事項 (e) 参照).
3000	制限条件 (a) または (b) を満足しなかった.	処理を打ち切る.

(6) **注意事項**

(a) 配列 AR, AI には, エルミート行列の実部, 虚部のそれぞれ上三角部分のみが格納されていればよい (付録 B 参照).

(b) $EPS \leq 0$ のときは内部で自動的に最適値を設定する. 通常は自動設定されるように負の値を設定されたい. なお, EPS はバイセクション法で固有値を求めるときに使用される.

(c) 固有値は E1<E2 のときには小さい順, E1>E2 のときには大きい順に格納される.

(d) E1=E2 のとき, 区間 [E1-EPS, E1+EPS] にある固有値が求まる. しかし, 通常は E1≠E2 となるように設定されたい.

(e) 逆反復法で最大反復回数をこえた場合 (IERR=2000 出力時) について

- $IW1(i) = 0$ の場合 :
  i 番目の固有ベクトル計算は正常終了している.
- $IW1(i) \neq 0$ の場合 :
  i 番目の固有ベクトル計算は収束条件が満たされず, 固有ベクトルの精度は低い.
  この場合 IW1(i) は, 反復回数が設定される.

なお正常終了時 (IERR=0 出力時) は, IW1(i) ← 0 が設定される.

(f) 固有ベクトルは正規直交系である.

(g) 固有ベクトルを必要としないときは, 4.7.6 $\left\{\begin{matrix}\text{ZCHREN}\\ \text{CCHREN}\end{matrix}\right\}$ を使用する.

(7) **使用例**

(a) 問 題

$$A = \begin{bmatrix} 7 & 3 & 1+2i & -1+2i \\ 3 & 7 & 1-2i & -1-2i \\ 1-2i & 1+2i & 7 & -3 \\ -1-2i & -1+2i & -3 & 7 \end{bmatrix}$$

のとき, エルミート行列 A の固有値を区間 [5, 15] 内で 15 に近いものから 3 個を降順で求め, それに対応する固有ベクトルを求める.

(b) 入力データ

行列 A の実部 AR, 虚部 AI, LNA=11, N=4, EPS=−1.0, M=3, E1=15, E2=5, LNV=11

(c) 主プログラム

```
      PROGRAM ACHREE
! *** EXAMPLE OF ZCHREE ***
      IMPLICIT REAL(8) (A-H,O-Z)
      CHARACTER*80 FMT
      PARAMETER ( LNA = 11, LNV = 11 )
      DIMENSION AR(LNA,LNA), AI(LNA,LNA), E(LNA),&
                VR(LNV,LNV), VI(LNV,LNV), IW1(LNA), W1(10*LNA)
!
      READ(5,*) N, M, E1, E2
      DO 10 I=1, N
         READ(5,*) (AR(I,J), AI(I,J), J=I, N)
   10 CONTINUE
!
      WRITE(6,1000) N, M, E1, E2
      DO 20 I=1, N
         WRITE(FMT,1100) N
         WRITE(6,FMT) (AR(J,I), -AI(J,I), J=1, I-1),&
                      (AR(I,J),  AI(I,J), J=I, N)
   20 CONTINUE
!
!     ISW = 1
      EPS = -1.0D0
!
      CALL ZCHREE(AR,AI,LNA,N,EPS,E,M,E1,E2,VR,VI,LNV,IW1,W1,IERR)
!
      WRITE(6,1200) IERR
!
      DO 40 J=1, M-1, 2
         WRITE(6,1300) (’EIGENVALUE ’, I=1, 2)
         WRITE(6,1400) E(J), E(J+1)
         WRITE(6,1300) (’EIGENVECTOR’, I=1, 2)
         DO 30 I=1, N
            WRITE(6,1500) VR(I,J), VI(I,J), VR(I,J+1), VI(I,J+1)
   30    CONTINUE
   40 CONTINUE
      IF(MOD(M,2).NE.0) THEN
         WRITE(6,1300) ’EIGENVALUE ’
         WRITE(6,1400) E(M)
         WRITE(6,1300) ’EIGENVECTOR’
         DO 50 I=1, N
            WRITE(6,1500) VR(I,M), VI(I,M)
   50    CONTINUE
      ENDIF
      STOP
!
 1000 FORMAT(1X,/,/,&
             1X,’***  ZCHREE  ***’,/,/,&
             1X,’ **  INPUT  **’,/,/,&
             1X,’N = ’, I4,’  M = ’, I4,/,/,&
             1X,’E1= ’, 1PD14.7,’  E2= ’, 1PD14.7,/,/,&
             1X,’     INPUT MATRIX A ( REAL,IMAGINARY )’,/)
 1100 FORMAT(’(1X,5X,’, I2,’(’’(’’,F5.1,’’ ,’’,F5.1,’’) ’’))’)
 1200 FORMAT(1X,/,/,&
             1X,’ **  OUTPUT  **’,/,/,&
             1X,’     IERR = ’, I4)
 1300 FORMAT(1X,/, 2(14X, A11, 8X))
 1400 FORMAT(1X, 2(12X, 1PD14.7, 7X))
 1500 FORMAT(1X, 2(5X, F12.8, ’ ,’, F12.8, 2X))
      END
```

(d) 出力結果

```
***  ZCHREE  ***

 **  INPUT  **

N =    4  M =    3

E1=  1.5000000D+01  E2=  5.0000000D+00

     INPUT MATRIX A ( REAL,IMAGINARY )

     (  7.0 ,  0.0) (  3.0 ,  0.0) (  1.0 ,  2.0) ( -1.0 ,  2.0)
     (  3.0 ,  0.0) (  7.0 ,  0.0) (  1.0 , -2.0) ( -1.0 , -2.0)
     (  1.0 , -2.0) (  1.0 ,  2.0) (  7.0 ,  0.0) ( -3.0 ,  0.0)
     ( -1.0 , -2.0) ( -1.0 ,  2.0) ( -3.0 ,  0.0) (  7.0 ,  0.0)


 **  OUTPUT  **

     IERR =    0

             EIGENVALUE                       EIGENVALUE
             1.2000000D+01                    8.0000000D+00

             EIGENVECTOR                      EIGENVECTOR
       0.50000000 ,  0.00000000         0.70710678 ,  0.00000000
       0.50000000 ,  0.00000000        -0.00000000 , -0.00000000
       0.50000000 , -0.00000000        -0.35355339 , -0.35355339
      -0.50000000 , -0.00000000         0.35355339 , -0.35355339

             EIGENVALUE
             8.0000000D+00
```

```
         EIGENVECTOR
  0.00000000 ,  0.00000000
 -0.09987868 ,  0.70001732
 -0.30006932 , -0.39994800
 -0.39994800 ,  0.30006932
```

4.7.6 ZCHREN, CCHREN
エルミート行列の固有値 (区間指定)

(1) **機 能**

エルミート行列 $A = (\mathrm{AR}, \mathrm{AI})$(2 次元配列型)(上三角型)(実数引数型) の指定した区間の固有値をハウスホルダー法, バイセクション法により, 昇順で m 個, もしくは降順で m 個求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZCHREN (AR, AI, LNA, N, EPS, E, M, E1, E2, W1, IERR)

単精度サブルーチン:

CALL CCHREN (AR, AI, LNA, N, EPS, E, M, E1, E2, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: {32 ビット整数版では INTEGER(4), 64 ビット整数版では INTEGER(8)}

項番	引数名	型	大きさ	入出力	内　容
1	AR	{D, R}	LNA, N	入 力	エルミート行列 A の実部 (2 次元配列型) (上三角型)
				出 力	入力時の内容は保存されない
2	AI	{D, R}	LNA, N	入 力	エルミート行列 A の虚部 (2 次元配列型) (上三角型)
				出 力	入力時の内容は保存されない
3	LNA	I	1	入 力	配列 AR, AI の整合寸法
4	N	I	1	入 力	行列 A の次数
5	EPS	{D, R}	1	入 力	固有値の収束判定のための絶対誤差の上限を与えるパラメータ (注意事項 (b) 参照)
6	E	{D, R}	M	出 力	固有値
7	M	I	1	入 力	求めたい固有値の個数 m
				出 力	求めた固有値の個数
8	E1	{D, R}	1	入 力	E1 < E2 の場合 : E1 から昇順で固有値を求める. (E2 は, 上限) E1 > E2 の場合 : E1 から降順で固有値を求める. (E2 は, 下限) (注意事項 (c)(d) 参照)
9	E2	{D, R}	1	入 力	
10	W1	{D, R}	5 × N	ワーク	作業領域
11	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNA}$

(b) $0 < \mathrm{M} \leq \mathrm{N}$

(5) **エラーインディケータ**

IERR の値	意　　味	処 理 内 容
0	正常終了.	
1000	N = 1 であった.	E(1) ← AR(1, 1) とする.
1500	区間 [E1, E2] に存在する固有値は M 個未満であった.	区間 [E1, E2] 内にある全ての固有値, 固有ベクトルを求める. 引数 M には求めた固有値の個数を出力する.
3000	制限条件 (a) または (b) を満足しなかった.	処理を打ち切る.

(6) **注意事項**

(a) 配列 AR, AI には, エルミート行列の実部, 虚部のそれぞれ上三角部分のみが格納されていればよい (付録 B 参照).

(b) EPS ≤ 0 のときは内部で自動的に最適値を設定する. 通常は自動設定されるように負の値を設定されたい. なお, EPS はバイセクション法で固有値を求めるときに使用される.

(c) 固有値は E1<E2 のときには小さい順, E1>E2 のときには大きい順に格納される.

(d) E1=E2 のとき, 区間 [E1-EPS, E1+EPS] にある固有値が求まる. しかし, 通常は E1≠E2 となるように設定されたい.

4.8 エルミート行列 (2 次元配列型) (上三角型)(複素引数型)

4.8.1 ZCHEAA, CCHEAA
エルミート行列の全固有値・全固有ベクトル

(1) **機 能**

エルミート行列 A (2 次元配列型)(上三角型)(複素引数型) の全固有値とそれに対応する全固有ベクトルをハウスホルダー法, QR 法により求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZCHEAA (A, LNA, N, E, W1, W2, IERR)

単精度サブルーチン:

CALL CCHEAA (A, LNA, N, E, W1, W2, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型　I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$

R:単精度実数型　C:単精度複素数型

項番	引数名	型	大きさ	入出力	内　　容
1	A	Z C	LNA, N	入 力	エルミート行列 A (2 次元配列型) (上三角型)
				出 力	各固有値に対応する固有ベクトル (列ベクトル)
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A の次数
4	E	D R	N	出 力	固有値
5	W1	D R	N	ワーク	作業領域
6	W2	Z C	N	ワーク	作業領域
7	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNA}$

(5) エラーインディケータ

IERR の値	意　　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	E(1) ← A(1, 1), A(1, 1) ← (1.0, 0.0) とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
$5000 + i$	固有値を求める段階で収束しなかった. ($1 \le i \le$ N)	E(1), $\cdots$, E($i-1$) にそれまでに求まった固有値が入る (ただし順不同). このとき固有ベクトルは求まらない.

(6) 注意事項

(a) 配列 A には, エルミート行列の上三角部分のみが格納されていればよい (付録 B 参照).

(b) 固有値は小さい順に格納される.

(c) 固有ベクトルは正規直交系である.

(d) 固有ベクトルを必要としないときは, 4.8.2 $\left\{\begin{matrix}\text{ZCHEAN}\\\text{CCHEAN}\end{matrix}\right\}$を使用する.

(7) 使用例

(a) 問 題

$$A = \begin{bmatrix} 7 & 3 & 1+2i & -1+2i \\ 3 & 7 & 1-2i & -1-2i \\ 1-2i & 1+2i & 7 & -3 \\ -1-2i & -1+2i & -3 & 7 \end{bmatrix}$$

の全固有値とそれに対応する固有ベクトルを求める.

(b) 入力データ

行列 A, LNA=11, N=4

(c) 主プログラム

```
      PROGRAM ACHEAA
! *** EXAMPLE OF ZCHEAA ***
      IMPLICIT REAL(8) (A-H,O-Z)
      CHARACTER*80 FMT
      COMPLEX(8) A,W2
      PARAMETER ( LNA = 11 )
      DIMENSION A(LNA,LNA), E(LNA), W1(LNA), W2(LNA)
!
      READ(5,*) N
      DO 10 I=1, N
         READ(5,*) (A(I,J), J=I, N)
   10 CONTINUE
!
      WRITE(6,1000) N
      DO 20 I=1, N
         WRITE(FMT,1100) N
         WRITE(6,FMT) (DCONJG(A(J,I)), J=1, I-1), (A(I,J), J=I, N)
   20 CONTINUE
!
      CALL ZCHEAA(A,LNA,N,E,W1,W2,IERR)
!
      WRITE(6,1200) IERR
!
      DO 40 J=1, N-1, 2
         WRITE(6,1300) (’EIGENVALUE ’, I=1, 2)
         WRITE(6,1400) E(J), E(J+1)
         WRITE(6,1300) (’EIGENVECTOR’, I=1, 2)
         DO 30 I=1, N
```

```
            WRITE(6,1500) A(I,J), A(I,J+1)
   30    CONTINUE
   40 CONTINUE
      IF(MOD(N,2).NE.0) THEN
         WRITE(6,1300) ’EIGENVALUE ’
         WRITE(6,1400) E(N)
         WRITE(6,1300) ’EIGENVECTOR’
         DO 50 I=1, N
            WRITE(6,1500) A(I,N)
   50    CONTINUE
      ENDIF
      STOP
!
 1000 FORMAT(’ ’,/,/,&
             ’ ***  ZCHEAA  ***’,/,/,&
             ’  **  INPUT  **’,/,/,&
             ’      N = ’, I4,/,/,&
             ’      INPUT MATRIX A ( REAL,IMAGINARY )’,/)
 1100 FORMAT(’( ,5X,’, I2,’(’’(’’,F5.1,’’ ,’’,F5.1,’’) ’’))’)
 1200 FORMAT(’ ’,/,/,&
             ’  **  OUTPUT  **’,/,/,&
             ’      IERR = ’, I4)
 1300 FORMAT(’ ’,/, 2(14X, A11, 8X))
 1400 FORMAT(’ ’, 2(12X, 1PD14.7, 7X))
 1500 FORMAT(’ ’, 2(5X, F12.8, ’ ,’, F12.8, 2X))
      END
```

(d) 出力結果

```
***  ZCHEAA  ***

 **  INPUT  **

     N =    4

     INPUT MATRIX A ( REAL,IMAGINARY )

    (  7.0 ,  0.0) (  3.0 ,  0.0) (  1.0 ,  2.0) ( -1.0 ,  2.0)
    (  3.0 ,  0.0) (  7.0 ,  0.0) (  1.0 , -2.0) ( -1.0 , -2.0)
    (  1.0 , -2.0) (  1.0 ,  2.0) (  7.0 ,  0.0) ( -3.0 ,  0.0)
    ( -1.0 , -2.0) ( -1.0 ,  2.0) ( -3.0 ,  0.0) (  7.0 ,  0.0)


 **  OUTPUT  **

     IERR =    0

             EIGENVALUE                       EIGENVALUE
             0.0000000D+00                    8.0000000D+00

             EIGENVECTOR                      EIGENVECTOR
       0.50000000 ,  0.00000000        -0.70710678 , -0.00000000
      -0.50000000 ,  0.00000000         0.00000000 , -0.00000000
       0.00000000 ,  0.50000000         0.35355339 ,  0.35355339
      -0.00000000 ,  0.50000000        -0.35355339 ,  0.35355339

             EIGENVALUE                       EIGENVALUE
             8.0000000D+00                    1.2000000D+01

             EIGENVECTOR                      EIGENVECTOR
       0.00000000 ,  0.00000000         0.50000000 ,  0.00000000
      -0.09987868 ,  0.70001732         0.50000000 ,  0.00000000
      -0.30006932 , -0.39994800         0.50000000 , -0.00000000
      -0.39994800 ,  0.30006932        -0.50000000 , -0.00000000
```

4.8.2 ZCHEAN, CCHEAN エルミート行列の全固有値

(1) **機 能**

エルミート行列 A(2 次元配列型)(上三角型)(複素引数型) の全固有値をハウスホルダー法, 無平方根 QR 法により求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZCHEAN (A, LNA, N, E, W1, W2, IERR)

単精度サブルーチン:

CALL CCHEAN (A, LNA, N, E, W1, W2, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\left\{\begin{array}{l}\text{32 ビット整数版では INTEGER(4)}\\ \text{64 ビット整数版では INTEGER(8)}\end{array}\right\}$

項番	引数名	型	大きさ	入出力	内　　　容
1	A	$\left\{\begin{array}{c}\text{Z}\\ \text{C}\end{array}\right\}$	LNA, N	入 力	エルミート行列 A (2 次元配列型) (上三角型)
				出 力	入力時の内容は保存されない
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A の次数
4	E	$\left\{\begin{array}{c}\text{D}\\ \text{R}\end{array}\right\}$	N	出 力	固有値
5	W1	$\left\{\begin{array}{c}\text{D}\\ \text{R}\end{array}\right\}$	N	ワーク	作業領域
6	W2	$\left\{\begin{array}{c}\text{Z}\\ \text{C}\end{array}\right\}$	N	ワーク	作業領域
7	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNA}$

(5) **エラーインディケータ**

IERR の値	意　　　味	処 理 内 容
0	正常終了.	
1000	$\mathrm{N} = 1$ であった.	$\mathrm{E}(1) \leftarrow \mathrm{A}(1, 1)$ とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
$5000 + i$	固有値を求める段階で収束しなかった. $(1 \leq i \leq \mathrm{N})$	$\mathrm{E}(1), \ldots, \mathrm{E}(i-1)$ にそれまでに求まった固有値が入る (ただし順不同).

(6) **注意事項**

(a) 配列 A には, エルミート行列の上三角部分のみが格納されていればよい (付録 B 参照).

(b) 固有値は小さい順に格納される.

4.8.3 ZCHESS, CCHESS エルミート行列の固有値・固有ベクトル

(1) **機 能**

エルミート行列 A (2 次元配列型)(上三角型)(複素引数型) の固有値をハウスホルダー法, 無平方根 QR 法またはバイセクション法により, 大きい方から m 個, または小さい方から m 個求め, それに対応する固有ベクトルを逆反復法により求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZCHESS (A, LNA, N, EPS, E, M, VE, LNV, ISW, IW1, W1, W2, IERR)

単精度サブルーチン:

CALL CCHESS (A, LNA, N, EPS, E, M, VE, LNV, ISW, IW1, W1, W2, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: { 32 ビット整数版では INTEGER(4)
64 ビット整数版では INTEGER(8) }

項番	引数名	型	大きさ	入出力	内　　容
1	A	Z C	LNA, N	入 力	エルミート行列 A (2 次元配列型)(上三角型)
				出 力	入力時の内容は保存されない
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A の次数
4	EPS	D R	1	入 力	固有値の収束判定のための絶対誤差の上限を与えるパラメータ (注意事項 (d) 参照)
5	E	D R	M	出 力	固有値
6	M	I	1	入 力	求めたい固有値の個数 m
7	VE	Z C	LNV, M	出 力	各固有値に対応する固有ベクトル (列ベクトル)
8	LNV	I	1	入 力	配列 VE の整合寸法
9	ISW	I	1	入 力	処理スイッチ ISW ≥ 0: 大きい方から固有値を求める. ISW < 0: 小さい方から固有値を求める.
10	IW1	I	M	出 力	固有ベクトルフラグ (注意事項 (e) 参照)
11	W1	D R	8 × N	ワーク	作業領域
12	W2	Z C	N	ワーク	作業領域
13	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNA}, \mathrm{LNV}$

(b) $0 < \mathrm{M} \leq \mathrm{N}$

(5) **エラーインディケータ**

IERR の値	意　　味	処　理　内　容
0	正常終了.	
1000	$N = 1$ であった.	$E(1) \leftarrow A(1,1)$, $VE(1,1) \leftarrow (1.0, 0.0)$ とする.
2000	固有ベクトルを求める逆反復で最大反復回数をこえた.	固有ベクトルに一部精度の低いものがあるが, 処理は続行する (注意事項 (e) 参照).
3000	制限条件 (a) または (b) を満足しなかった.	処理を打ち切る.

(6) **注意事項**

(a) 配列 A には, エルミート行列の上三角部分のみが格納されていればよい (付録 B 参照).

(b) 固有値は ISW≥0 のときには大きい順に, ISW<0 のときには小さい順に格納される.

(c) 固有値は無平方根 QR 法とバイセクション法を内部で適切に切り分けて計算している.

(d) $\mathrm{EPS} \leq 0$ のときは内部で自動的に最適値を設定する. 通常は自動設定されるように負の値を設定されたい. なお, EPS はバイセクション法で固有値を求めるときに使用される.

(e) 逆反復法で最大反復回数をこえた場合 (IERR=2000 出力時) について

- $\mathrm{IW1}(i) = 0$ の場合 :
  i 番目の固有ベクトル計算は正常終了している.
- $\mathrm{IW1}(i) \neq 0$ の場合 :
  i 番目の固有ベクトル計算は収束条件が満たされず, 固有ベクトルの精度は低い.
  この場合 $\mathrm{IW1}(i)$ は, 反復回数が設定される.

なお正常終了時 (IERR=0 出力時) は, $\mathrm{IW1}(i) \leftarrow 0$ が設定される.

(f) 固有ベクトルは正規直交系である.

(g) 固有ベクトルを必要としないときは, 4.8.4 $\left\{\begin{matrix}\mathrm{ZCHESN}\\ \mathrm{CCHESN}\end{matrix}\right\}$を使用する.

(7) **使用例**

(a) 問 題

$$A = \begin{bmatrix} 7 & 3 & 1+2i & -1+2i \\ 3 & 7 & 1-2i & -1-2i \\ 1-2i & 1+2i & 7 & -3 \\ -1-2i & -1+2i & -3 & 7 \end{bmatrix}$$

のとき, エルミート行列 A の固有値を大きい順に 3 個とそれに対応する固有ベクトルを求める.

(b) 入力データ

行列 A, LNA=11, N=4, EPS=−1.0, M=3, LNV=11, ISW=1

(c) 主プログラム

```
      PROGRAM ACHESS
! *** EXAMPLE OF ZCHESS ***
      IMPLICIT REAL(8) (A-H,O-Z)
      CHARACTER*80 FMT
      COMPLEX(8) A,VE,W2
      PARAMETER ( LNA = 11, LNV = 11 )
      DIMENSION A(LNA,LNA), E(LNA), VE(LNV,LNV),&
                IW1(LNA), W1(8*LNA), W2(LNA)
!
      READ(5,*) N, M
      DO 10 I=1, N
         READ(5,*) (A(I,J), J=I, N)
   10 CONTINUE
!
      WRITE(6,1000) N, M
      DO 20 I=1, N
         WRITE(FMT,1100) N
         WRITE(6,FMT) (DCONJG(A(J,I)), J=1, I-1), (A(I,J), J=I, N)
   20 CONTINUE
!
      ISW = 1
      EPS = -1.0D0
!
      CALL ZCHESS(A,LNA,N,EPS,E,M,VE,LNV,ISW,IW1,W1,W2,IERR)
!
      WRITE(6,1200) IERR
!
      DO 40 J=1, M-1, 2
         WRITE(6,1300) ('EIGENVALUE ', I=1, 2)
         WRITE(6,1400) E(J), E(J+1)
         WRITE(6,1300) ('EIGENVECTOR', I=1, 2)
         DO 30 I=1, N
            WRITE(6,1500) VE(I,J), VE(I,J+1)
   30    CONTINUE
   40 CONTINUE
      IF(MOD(M,2).NE.0) THEN
         WRITE(6,1300) 'EIGENVALUE '
         WRITE(6,1400) E(M)
         WRITE(6,1300) 'EIGENVECTOR'
         DO 50 I=1, N
            WRITE(6,1500) VE(I,M)
   50    CONTINUE
      ENDIF
      STOP
!
 1000 FORMAT(' ',/,/,&
             ' ***  ZCHESS  ***',/,/,&
             '  **  INPUT  **',/,/,&
             '      N = ', I4,/,/,&
             '      M = ', I4,/,/,&
             '      INPUT MATRIX A ( REAL,IMAGINARY )',/)
 1100 FORMAT('( ,5X,', I2,'(''('',F5.1,'' ,'',F5.1,'') ''))')
 1200 FORMAT(' ',/,/,&
             '  **  OUTPUT  **',/,/,&
             '      IERR = ', I4)
 1300 FORMAT(' ',/, 2(14X, A11, 8X))
 1400 FORMAT(' ', 2(12X, 1PD14.7, 7X))
 1500 FORMAT(' ', 2(5X, F12.8, ' ,', F12.8, 2X))
      END
```

(d) 出力結果

```
***  ZCHESS  ***

 **  INPUT  **

     N =    4

     M =    3

     INPUT MATRIX A ( REAL,IMAGINARY )

    (  7.0 ,  0.0) (  3.0 ,  0.0) (  1.0 ,  2.0) ( -1.0 ,  2.0)
    (  3.0 ,  0.0) (  7.0 ,  0.0) (  1.0 , -2.0) ( -1.0 , -2.0)
    (  1.0 , -2.0) (  1.0 ,  2.0) (  7.0 ,  0.0) ( -3.0 ,  0.0)
    ( -1.0 , -2.0) ( -1.0 ,  2.0) ( -3.0 ,  0.0) (  7.0 ,  0.0)


 **  OUTPUT  **

     IERR =    0

             EIGENVALUE                       EIGENVALUE
            1.2000000D+01                     8.0000000D+00

             EIGENVECTOR                      EIGENVECTOR
       0.50000000 ,  0.00000000         0.00000000 ,  0.00000000
       0.50000000 ,  0.00000000        -0.09987868 ,  0.70001732
       0.50000000 , -0.00000000        -0.30006932 , -0.39994800
      -0.50000000 , -0.00000000        -0.39994800 ,  0.30006932

             EIGENVALUE
            8.0000000D+00

             EIGENVECTOR
       0.70710678 ,  0.00000000
      -0.00000000 ,  0.00000000
      -0.35355339 , -0.35355339
       0.35355339 , -0.35355339
```

4.8.4 ZCHESN, CCHESN
エルミート行列の固有値

(1) 機 能

エルミート行列 A (2 次元配列型)(上三角型)(複素引数型) の固有値をハウスホルダー法, 無平方根 QR 法またはバイセクション法により, 大きい方から m 個, または小さい方から m 個求める.

(2) 使用法

倍精度サブルーチン:

CALL ZCHESN (A, LNA, N, EPS, E, M, ISW, W1, W2, IERR)

単精度サブルーチン:

CALL CCHESN (A, LNA, N, EPS, E, M, ISW, W1, W2, IERR)

(3) 引 数

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: $\left\{\begin{array}{l}\text{32 ビット整数版では INTEGER(4)}\\ \text{64 ビット整数版では INTEGER(8)}\end{array}\right\}$

項番	引数名	型	大きさ	入出力	内　容
1	A	$\left\{\begin{matrix}\mathrm{Z}\\ \mathrm{C}\end{matrix}\right\}$	LNA, N	入 力	エルミート行列 A (2 次元配列型) (上三角型)
				出 力	入力時の内容は保存されない
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A の次数
4	EPS	$\left\{\begin{matrix}\mathrm{D}\\ \mathrm{R}\end{matrix}\right\}$	1	入 力	固有値の収束判定のための絶対誤差の上限を与えるパラメータ (注意事項 (d) 参照)
5	E	$\left\{\begin{matrix}\mathrm{D}\\ \mathrm{R}\end{matrix}\right\}$	M	出 力	固有値
6	M	I	1	入 力	求めたい固有値の個数 m
7	ISW	I	1	入 力	処理スイッチ ISW $\geq$ 0: 大きい方から固有値を求める. ISW $<$ 0: 小さい方から固有値を求める.
8	W1	$\left\{\begin{matrix}\mathrm{D}\\ \mathrm{R}\end{matrix}\right\}$	$2 \times$ N	ワーク	作業領域
9	W2	$\left\{\begin{matrix}\mathrm{Z}\\ \mathrm{C}\end{matrix}\right\}$	N	ワーク	作業領域
10	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a) $0 <$ N $\leq$ LNA

(b) $0 <$ M $\leq$ N

(5) **エラーインディケータ**

IERR の値	意　　　　　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	$\mathrm{E}(1) \leftarrow \mathrm{A}(1,1)$ とする.
3000	制限条件 (a) または (b) を満足しなかった.	処理を打ち切る.

(6) **注意事項**

(a) 配列 A には, エルミート行列の上三角部分のみが格納されていればよい (付録 B 参照).

(b) 固有値は ISW≥0 のときには大きい順に, ISW<0 のときには小さい順に格納される.

(c) 固有値は無平方根 QR 法とバイセクション法を内部で適切に切り分けて計算している.

(d) EPS ≤ 0 のときは内部で自動的に最適値を設定する. 通常は自動設定されるように負の値を設定されたい. なお, EPS はバイセクション法で固有値を求めるときに使用される.

4.8.5 ZCHEEE, CCHEEE
エルミート行列の固有値・固有ベクトル (区間指定)

(1) **機 能**

エルミート行列 A (2 次元配列型) (上三角型) (複素引数型) の指定した区間の固有値をハウスホルダー法, バイセクション法により, 昇順で m 個, もしくは降順で m 個求め, それに対応する固有ベクトルを逆反復法により求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZCHEEE (A, LNA, N, EPS, E, M, E1, E2, VE, LNV, IW1, W1, W2, IERR)

単精度サブルーチン:

CALL CCHEEE (A, LNA, N, EPS, E, M, E1, E2, VE, LNV, IW1, W1, W2, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型　I: {32 ビット整数版では INTEGER(4) / 64 ビット整数版では INTEGER(8)}
R:単精度実数型　C:単精度複素数型

項番	引数名	型	大きさ	入出力	内容
1	A	Z C	LNA, N	入 力	エルミート行列 A (2 次元配列型)(上三角型)
				出 力	入力時の内容は保存されない
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A の次数
4	EPS	D R	1	入 力	固有値の収束判定のための絶対誤差の上限を与えるパラメータ (注意事項 (b) 参照)
5	E	D R	M	出 力	固有値
6	M	I	1	入 力	求めたい固有値の個数 m
				出 力	求めた固有値の個数
7	E1	D R	1	入 力	E1 < E2 の場合： E1 から昇順で固有値を求める. (E2 は, 上限) E1 > E2 の場合： E1 から降順で固有値を求める. (E2 は, 下限) (注意事項 (c)(d) 参照)
8	E2	D R	1	入 力	
9	VE	Z C	LNV, M	出 力	各固有値に対応する固有ベクトル (列ベクトル)
10	LNV	I	1	入 力	配列 VE の整合寸法
11	IW1	I	M	出 力	固有ベクトルフラグ (注意事項 (e) 参照)
12	W1	D R	8×N	ワーク	作業領域
13	W2	Z C	N	ワーク	作業領域
14	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a) $0 < \mathrm{N} \le \mathrm{LNA}, \mathrm{LNV}$

(b) $0 < \mathrm{M} \le \mathrm{N}$

(5) エラーインディケータ

IERR の値	意　　　味	処　理　内　容
0	正常終了.	
1000	$\mathrm{N} = 1$ であった.	$\mathrm{E}(1) \leftarrow \mathrm{A}(1,1)$, $\mathrm{VE}(1,1) \leftarrow (1.0, 0.0)$ とする.
1500	区間 [E1, E2] に存在する固有値は M 個未満であった.	区間 [E1, E2] 内にある全ての固有値, 固有ベクトルを求める. 引数 M には求めた固有値の個数を出力する.
2000	固有ベクトルを求める逆反復で最大反復回数をこえた.	固有ベクトルに一部精度の低いものがあるが, 処理は続行する (注意事項 (e) 参照).
3000	制限条件 (a) または (b) を満足しなかった.	処理を打ち切る.

(6) 注意事項

(a) 配列 A には, エルミート行列の上三角部分のみが格納されていればよい (付録 B 参照).

(b) $\mathrm{EPS} \le 0$ のときは内部で自動的に最適値を設定する. 通常は自動設定されるように負の値を設定されたい. なお, EPS はバイセクション法で固有値を求めるときに使用される.

(c) 固有値は E1<E2 のときには小さい順, E1>E2 のときには大きい順に格納される.

(d) E1=E2 のとき, 区間 [E1-EPS, E1+EPS] にある固有値が求まる. しかし, 通常は E1≠E2 となるように設定されたい.

(e) 逆反復法で最大反復回数をこえた場合 (IERR=2000 出力時) について

- $\mathrm{IW1}(i) = 0$ の場合 :
  i 番目の固有ベクトル計算は正常終了している.
- $\mathrm{IW1}(i) \neq 0$ の場合 :
  i 番目の固有ベクトル計算は収束条件が満たされず, 固有ベクトルの精度は低い.
  この場合 $\mathrm{IW1}(i)$ は, 反復回数が設定される.

なお正常終了時 (IERR=0 出力時) は, $\mathrm{IW1}(i) \leftarrow 0$ が設定される.

(f) 固有ベクトルは正規直交系である.

(g) 固有ベクトルを必要としないときは, 4.8.6 $\begin{Bmatrix} \text{ZCHEEN} \\ \text{CCHEEN} \end{Bmatrix}$ を使用する.

(7) **使用例**

(a) 問 題

$$A = \begin{bmatrix} 7 & 3 & 1+2i & -1+2i \\ 3 & 7 & 1-2i & -1-2i \\ 1-2i & 1+2i & 7 & -3 \\ -1-2i & -1+2i & -3 & 7 \end{bmatrix}$$

のとき, エルミート行列 A の固有値を区間 [5, 15] 内で 15 に近いものから 3 個を降順で求め, それに対応する固有ベクトルを求める.

(b) 入力データ

行列 A, LNA=11, N=4, EPS=-1.0, M=3, E1=15, E2=5, LNV=11

(c) 主プログラム

```
      PROGRAM ACHEEE
! *** EXAMPLE OF ZCHEEE ***
      IMPLICIT REAL(8) (A-H,O-Z)
      CHARACTER*80 FMT
      COMPLEX(8) A,VE,W2
      PARAMETER ( LNA = 11, LNV = 11 )
      DIMENSION A(LNA,LNA), E(LNA), VE(LNV,LNV),&
                IW1(LNA), W1(8*LNA), W2(LNA)
!
      READ(5,*) N, M, E1, E2
      DO 10 I=1, N
         READ(5,*) (A(I,J), J=I, N)
   10 CONTINUE
!
      WRITE(6,1000) N, M, E1, E2
      DO 20 I=1, N
         WRITE(FMT,1100) N
         WRITE(6,FMT) (DCONJG(A(J,I)), J=1, I-1), (A(I,J), J=I, N)
   20 CONTINUE
!
!     ISW = 1
      EPS = -1.0D0
!
      CALL ZCHEEE(A,LNA,N,EPS,E,M,E1,E2,VE,LNV,IW1,W1,W2,IERR)
!
      WRITE(6,1200) IERR
!
      DO 40 J=1, M-1, 2
         WRITE(6,1300) ('EIGENVALUE ', I=1, 2)
         WRITE(6,1400) E(J), E(J+1)
         WRITE(6,1300) ('EIGENVECTOR', I=1, 2)
         DO 30 I=1, N
            WRITE(6,1500) VE(I,J), VE(I,J+1)
   30    CONTINUE
   40 CONTINUE
      IF(MOD(M,2).NE.0) THEN
         WRITE(6,1300) 'EIGENVALUE '
         WRITE(6,1400) E(M)
         WRITE(6,1300) 'EIGENVECTOR'
         DO 50 I=1, N
            WRITE(6,1500) VE(I,M)
   50    CONTINUE
      ENDIF
      STOP
!
 1000 FORMAT(1X,/,/,&
             1X,'***  ZCHEEE  ***',/,/,&
             1X,' **  INPUT  **',/,/,&
             1X,'N = ', I4,'  M = ', I4,/,/,&
             1X,'E1= ', 1PD14.7,'  E2= ', 1PD14.7,/,/,&
             1X,'     INPUT MATRIX A ( REAL,IMAGINARY )',/)
 1100 FORMAT('(1X,5X,', I2,'(''('',F5.1,'' ,'',F5.1,'') ''))')
 1200 FORMAT(1X,/,/,&
             1X,' **  OUTPUT  **',/,/,&
             1X,'     IERR = ', I4)
 1300 FORMAT(1X,/, 2(14X, A11, 8X))
 1400 FORMAT(1X, 2(12X, 1PD14.7, 7X))
 1500 FORMAT(1X, 2(5X, F12.8, ' ,', F12.8, 2X))
      END
```

(d) 出力結果

```
***  ZCHEEE  ***

 **  INPUT  **

N =    4  M =    3

E1=  1.5000000D+01  E2=  5.0000000D+00

     INPUT MATRIX A ( REAL,IMAGINARY )
```

```
    (  7.0 ,  0.0) (  3.0 ,  0.0) (  1.0 ,  2.0) ( -1.0 ,  2.0)
    (  3.0 ,  0.0) (  7.0 ,  0.0) (  1.0 , -2.0) ( -1.0 , -2.0)
    (  1.0 , -2.0) (  1.0 ,  2.0) (  7.0 ,  0.0) ( -3.0 ,  0.0)
    ( -1.0 , -2.0) ( -1.0 ,  2.0) ( -3.0 ,  0.0) (  7.0 ,  0.0)

**  OUTPUT  **

    IERR =    0

            EIGENVALUE                       EIGENVALUE
            1.2000000D+01                    8.0000000D+00

            EIGENVECTOR                      EIGENVECTOR
      0.50000000 ,  0.00000000         0.70710678 ,  0.00000000
      0.50000000 ,  0.00000000        -0.00000000 ,  0.00000000
      0.50000000 , -0.00000000        -0.35355339 , -0.35355339
     -0.50000000 , -0.00000000         0.35355339 , -0.35355339

            EIGENVALUE
            8.0000000D+00

            EIGENVECTOR
      0.00000000 ,  0.00000000
     -0.09987868 ,  0.70001732
     -0.30006932 , -0.39994800
     -0.39994800 ,  0.30006932
```

4.8.6 ZCHEEN, CCHEEN
エルミート行列の固有値 (区間指定)

(1) **機 能**

エルミート行列 A (2 次元配列型) (上三角型) (複素引数型) の指定した区間の固有値をハウスホルダー法, バイセクション法により, 昇順で m 個, もしくは降順で m 個求める.

(2) **使用法**

倍精度サブルーチン:
CALL ZCHEEN (A, LNA, N, EPS, E, M, E1, E2, W1, W2, IERR)
単精度サブルーチン:
CALL CCHEEN (A, LNA, N, EPS, E, M, E1, E2, W1, W2, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$

項番	引数名	型	大きさ	入出力	内　容
1	A	$\begin{Bmatrix}\mathrm{Z}\\\mathrm{C}\end{Bmatrix}$	LNA, N	入 力	エルミート行列 A (2 次元配列型) (上三角型)
				出 力	入力時の内容は保存されない
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A の次数
4	EPS	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	1	入 力	固有値の収束判定のための絶対誤差の上限を与えるパラメータ (注意事項 (b) 参照)
5	E	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	M	出 力	固有値
6	M	I	1	入 力	求めたい固有値の個数 m
				出 力	求めた固有値の個数
7	E1	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	1	入 力	E1 $<$ E2 の場合： E1 から昇順で固有値を求める. (E2 は, 上限)
8	E2	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	1	入 力	E1 $>$ E2 の場合： E1 から降順で固有値を求める. (E2 は, 下限) (注意事項 (c)(d) 参照)
9	W1	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	$2 \times \mathrm{N}$	ワーク	作業領域
10	W2	$\begin{Bmatrix}\mathrm{Z}\\\mathrm{C}\end{Bmatrix}$	N	ワーク	作業領域
11	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNA}$

(b) $0 < \mathrm{M} \leq \mathrm{N}$

(5) **エラーインディケータ**

IERR の値	意　味	処　理　内　容
0	正常終了.	
1000	$N = 1$ であった.	$E(1) \leftarrow A(1, 1)$ とする.
1500	区間 [E1, E2] に存在する固有値は M 個未満であった.	区間 [E1, E2] 内にある全ての固有値, 固有ベクトルを求める. 引数 M には求めた固有値の個数を出力する.
3000	制限条件 (a) または (b) を満足しなかった.	処理を打ち切る.

(6) **注意事項**

(a) 配列 A には, エルミート行列の上三角部分のみが格納されていればよい (付録 B 参照).

(b) EPS ≤ 0 のときは内部で自動的に最適値を設定する. 通常は自動設定されるように負の値を設定されたい. なお, EPS はバイセクション法で固有値を求めるときに使用される.

(c) 固有値は E1<E2 のときには小さい順, E1>E2 のときには大きい順に格納される.

(d) E1=E2 のとき, 区間 [E1-EPS, E1+EPS] にある固有値が求まる. しかし, 通常は E1≠E2 となるように設定されたい.

4.9 実対称バンド行列 (対称バンド型)

4.9.1 DCSBAA, RCSBAA
実対称バンド行列の全固有値・全固有ベクトル

(1) **機能**

実対称バンド行列 A(対称バンド型) の全固有値とそれに対応する全固有ベクトルをギブンス法またはハウスホルダー法及び QR 法により求める.

(2) **使用法**

倍精度サブルーチン:
CALL DCSBAA (A, LMA, N, MB, E, VE, LNV, W1, IERR)
単精度サブルーチン:
CALL RCSBAA (A, LMA, N, MB, E, VE, LNV, W1, IERR)

(3) **引数**

D:倍精度実数型 Z:倍精度複素数型
R:単精度実数型 C:単精度複素数型
I: 32 ビット整数版では INTEGER(4)
64 ビット整数版では INTEGER(8)

項番	引数名	型	大きさ	入出力	内容
1	A	D R	LMA, N	入力	実対称バンド行列 A(対称バンド型) (注意事項 (a) 参照)
				出力	入力時の内容は保存されない
2	LMA	I	1	入力	配列 A の整合寸法
3	N	I	1	入力	行列 A の次数
4	MB	I	1	入力	行列 A のバンド幅
5	E	D R	N	出力	固有値
6	VE	D R	LNV, N	出力	各固有値に対応する固有ベクトル (列ベクトル)
7	LNV	I	1	入力	配列 VE の整合寸法
8	W1	D R	N	ワーク	作業領域
9	IERR	I	1	出力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNV}$

(b) $0 \leq \mathrm{MB} < \mathrm{N}$

(c) $\mathrm{MB} < \mathrm{LMA}$

(5) エラーインディケータ

IERR の値	意　　　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	E(1) ← A(1, 1), VE(1, 1) ← 1.0 とする.
1100	MB = 0 であった.	E(i) ← A(1, i)　(i = 1, . . . , N), VE ← I　(I は単位行列) とする.
3000	制限条件 (a), (b) または (c) を満足しなかった.	処理を打ち切る.
$5000 + i$	固有値を求める段階で収束しなかった. ($1 \le i \le$ N)	E(1), · · · , E($i-1$) にそれまでに求まった固有値, A にそれに対応する固有ベクトルが入る (ただし順不同).

(6) 注意事項

(a) 配列 A には, 実対称バンド行列を対称バンド型 ((MB+1) 行 N 列) に圧縮して格納する (付録 B 参照).

(b) 固有値は小さい順に格納される.

(c) 固有ベクトルは正規直交系である.

(d) 固有ベクトルを必要としないときは, 4.9.2 $\begin{Bmatrix}\text{DCSBAN}\\ \text{RCSBAN}\end{Bmatrix}$を使用する.

(7) 使用例

(a) 問 題

$$A = \begin{bmatrix} 5 & -4 & 1 & & & & \\ -4 & 6 & -4 & 1 & & 0 & \\ 1 & -4 & 6 & -4 & 1 & & \\ & 1 & -4 & 6 & -4 & 1 & \\ & & 1 & -4 & 6 & -4 & 1 \\ & 0 & & 1 & -4 & 6 & -4 \\ & & & & 1 & -4 & 5 \end{bmatrix}$$

の全固有値とそれに対応する固有ベクトルを求める.

(b) 入力データ

行列 A, LMA=11, N=7, MB=2, LNV=11

(c) 主プログラム

```
      PROGRAM BCSBAA
! *** EXAMPLE OF DCSBAA ***
      IMPLICIT REAL(8) (A-H,O-Z)
      CHARACTER*80 FMT
      PARAMETER ( LMA = 11, LNV = 11 )
      DIMENSION A(LMA,LMA), E(LMA), VE(LNV,LNV), W1(LMA)
!
      READ(5,*) N, MB
      DO 10 J=1, MB+1
         READ(5,*) (A(J,I), I=MB-J+2, N)
   10 CONTINUE
!
```

```
      WRITE(6,1000) N, MB
      DO 20 J=1, MB+1
         WRITE(FMT,1100) (MB-J+1)*7+1, N-MB+J-1
         WRITE(6,FMT) (A(J,I), I=MB-J+2, N)
   20 CONTINUE
!
      CALL DCSBAA(A,LMA,N,MB,E,VE,LNV,W1,IERR)
!
      WRITE(6,1200) IERR
!
      DO 40 K=1, N-3, 4
         WRITE(6,1300) ('EIGENVALUE ', I=1, 4)
         WRITE(6,1400) (E(I), I=K, K+3)
         WRITE(6,1300) ('EIGENVECTOR', I=1, 4)
         DO 30 J=1, N
            WRITE(6,1500) (VE(J,I), I=K, K+3)
   30    CONTINUE
   40 CONTINUE
      IF(MOD(N,4).NE.0) THEN
         WRITE(6,1300) ('EIGENVALUE ', I=N/4*4+1, N)
         WRITE(6,1400) (E(I), I=N/4*4+1, N)
         WRITE(6,1300) ('EIGENVECTOR', I=N/4*4+1, N)
         DO 50 J=1, N
            WRITE(6,1500) (VE(J,I), I=N/4*4+1, N)
   50    CONTINUE
      ENDIF
      STOP
!
 1000 FORMAT(' ',/,/,&
             ' ***  DCSBAA  ***',/,/,&
             '  **  INPUT  **',/,/,&
             '      N = ', I2,/,/,&
             '      BAND WIDTH = ', I2,/,/,&
             '      INPUT MATRIX A',/)
 1100 FORMAT('( ,', I3,'X,', I2,'(F7.1))')
 1200 FORMAT(' ',/,/,&
             '  **  OUTPUT  **',/,/,&
             '      IERR = ', I4)
 1300 FORMAT(' ',/,1X, 4(5X, A11, 2X))
 1400 FORMAT(3X, 4(2X, 1PD14.7, 2X))
 1500 FORMAT(2X, 4(F14.8, 4X))
      END
```

(d) 出力結果

```
***  DCSBAA  ***

 **  INPUT  **

     N =  7

     BAND WIDTH =  2

     INPUT MATRIX A

                  1.0    1.0    1.0    1.0    1.0
          -4.0   -4.0   -4.0   -4.0   -4.0   -4.0
    5.0    6.0    6.0    6.0    6.0    6.0    5.0


 **  OUTPUT  **

     IERR =    0

     EIGENVALUE        EIGENVALUE        EIGENVALUE        EIGENVALUE
     2.3177302D-02     3.4314575D-01     1.5243190D+00     4.0000000D+00

     EIGENVECTOR       EIGENVECTOR       EIGENVECTOR       EIGENVECTOR
     0.19134172       -0.35355339       -0.46193977       -0.50000000
     0.35355339       -0.50000000       -0.35355339       -0.00000000
     0.46193977       -0.35355339        0.19134172        0.50000000
     0.50000000       -0.00000000        0.50000000        0.00000000
     0.46193977        0.35355339        0.19134172       -0.50000000
     0.35355339        0.50000000       -0.35355339        0.00000000
     0.19134172        0.35355339       -0.46193977        0.50000000

     EIGENVALUE        EIGENVALUE        EIGENVALUE
     7.6472539D+00     1.1656854D+01     1.4805250D+01

     EIGENVECTOR       EIGENVECTOR       EIGENVECTOR
    -0.46193977       -0.35355339        0.19134172
     0.35355339        0.50000000       -0.35355339
     0.19134172       -0.35355339        0.46193977
    -0.50000000        0.00000000       -0.50000000
     0.19134172        0.35355339        0.46193977
     0.35355339       -0.50000000       -0.35355339
    -0.46193977        0.35355339        0.19134172
```

4.9.2 DCSBAN, RCSBAN
実対称バンド行列の全固有値

(1) **機 能**

実対称バンド行列 A(対称バンド型) の全固有値をギブンス法, 無平方根 QR 法により求める.

(2) **使用法**

倍精度サブルーチン:

CALL DCSBAN (A, LMA, N, MB, E, W1, IERR)

単精度サブルーチン:

CALL RCSBAN (A, LMA, N, MB, E, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$

項番	引数名	型	大きさ	入出力	内　容
1	A	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	LMA, N	入 力	実対称バンド行列 A(対称バンド型) (注意事項 (a) 参照)
				出 力	入力時の内容は保存されない
2	LMA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A の次数
4	MB	I	1	入 力	行列 A のバンド幅
5	E	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	N	出 力	固有値
6	W1	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	N	ワーク	作業領域
7	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 \leq \text{MB} < \text{N}$

(b) $\text{MB} < \text{LMA}$

(5) エラーインディケータ

IERR の値	意　　　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	$E(1) \leftarrow A(1,1)$ とする.
1100	MB = 0 であった.	$E(i) \leftarrow A(1,i) \quad (i = 1,\ldots,N)$ とする.
3000	制限条件 (a) または (b) を満足しなかった.	処理を打ち切る.
$5000 + i$	固有値を求める段階で収束しなかった. $(1 \le i \le N)$	$E(1),\cdots,E(i-1)$ にそれまでに求まった固有値が入る (ただし順不同).

(6) 注意事項

(a) 配列 A には, 実対称バンド行列を対称バンド型 ((MB+1) 行 N 列) に圧縮して格納する (付録 B 参照).

(b) 固有値は小さい順に格納される.

4.9.3 DCSBSS, RCSBSS
実対称バンド行列の固有値・固有ベクトル

(1) **機 能**

実対称バンド行列 A(対称バンド型) の固有値をギブンス法, 無平方根 QR またはバイセクション法により大きい方から m 個, または小さい方から m 個求め, それに対応する固有ベクトルを逆反復法により求める.

(2) **使用法**

倍精度サブルーチン:
CALL DCSBSS (A, LMA, N, MB, EPS, E, M, VE, LNV, ISW, IW1, W1, IERR)
単精度サブルーチン:
CALL RCSBSS (A, LMA, N, MB, EPS, E, M, VE, LNV, ISW, IW1, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\left\{\begin{array}{l}\text{32 ビット整数版では INTEGER(4)}\\ \text{64 ビット整数版では INTEGER(8)}\end{array}\right\}$

項番	引数名	型	大きさ	入出力	内　容
1	A	$\left\{\begin{matrix}D\\R\end{matrix}\right\}$	LMA, N	入 力	実対称バンド行列 A(対称バンド型)(注意事項 (a) 参照)
				出 力	入力時の内容は保存されない
2	LMA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A の次数
4	MB	I	1	入 力	行列 A のバンド幅
5	EPS	$\left\{\begin{matrix}D\\R\end{matrix}\right\}$	1	入 力	固有値の収束判定のための絶対誤差の上限を与えるパラメータ (注意事項 (d) 参照)
6	E	$\left\{\begin{matrix}D\\R\end{matrix}\right\}$	M	出 力	固有値
7	M	I	1	入 力	求めたい固有値の個数 m
8	VE	$\left\{\begin{matrix}D\\R\end{matrix}\right\}$	LNV, M	出 力	各固有値に対応する固有ベクトル (列ベクトル)
9	LNV	I	1	入 力	配列 VE の整合寸法
10	ISW	I	1	入 力	処理スイッチ ISW ≥ 0: 大きい方から固有値を求める. ISW < 0: 小さい方から固有値を求める.
11	IW1	I	M	出 力	固有ベクトルフラグ (注意事項 (e) 参照)
12	W1	$\left\{\begin{matrix}D\\R\end{matrix}\right\}$	内容参照	ワーク	作業領域 **大きさ:** $(\mathrm{N} \times (3 \times \mathrm{MB} + 6))$
13	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \le \mathrm{LNV}$

(b) $0 < \mathrm{M} \le \mathrm{N}$

(c) $0 \le \mathrm{MB} < \mathrm{N}$

(d) $\mathrm{MB} < \mathrm{LMA}$

(5) **エラーインディケータ**

IERR の値	意　　味	処　理　内　容
0	正常終了.	
1000	$\mathrm{N} = 1$ であった.	$\mathrm{E}(1) \leftarrow \mathrm{A}(1,1)$, $\mathrm{VE}(1,1) \leftarrow 1.0$ とする.
1100	$\mathrm{MB} = 0$ であった.	$\mathrm{E}(i) \leftarrow \mathrm{A}(1,i) \quad (i = 1,\ldots,\mathrm{N})$ とする. VE の各列ベクトルの適当な成分に 1.0, 残りの成分に 0.0 が入る.
2000	固有ベクトルを求める逆反復で最大反復回数をこえた.	固有ベクトルに一部精度の低いものがあるが, 処理は続行する (注意事項 (e) 参照).
3000	制限条件 (a), (b), (c) または (d) を満足しなかった.	処理を打ち切る.

(6) **注意事項**

(a) 配列 A には, 実対称バンド行列を対称バンド型 ((MB+1) 行 N 列) に圧縮して格納する (付録 B 参照).

(b) 固有値は ISW≥0 のときには大きい順に, ISW<0 のときには小さい順に格納される.

(c) 固有値は無平方根 QR とバイセクション法を内部で適切に切り分けて計算している.

(d) $\mathrm{EPS} \le 0$ のときは内部で自動的に最適値を設定する. 通常は自動設定されるように負の値を設定されたい. なお, EPS はバイセクション法で固有値を求めるときに使用される.

(e) 逆反復法で最大反復回数をこえた場合 (IERR=2000 出力時) について

- $\mathrm{IW1}(i) = 0$ の場合 :
  i 番目の固有ベクトル計算は正常終了している.
- $\mathrm{IW1}(i) \neq 0$ の場合 :
  i 番目の固有ベクトル計算は収束条件が満たされず, 固有ベクトルの精度は低い.
  この場合 $\mathrm{IW1}(i)$ は, 反復回数が設定される.

なお正常終了時 (IERR=0 出力時) は, $\mathrm{IW1}(i) \leftarrow 0$ が設定される.

(f) 固有ベクトルは正規直交系である.

(g) 固有ベクトルを必要としないときは, 4.9.4 $\left\{\begin{matrix}\mathrm{DCSBSN}\\ \mathrm{RCSBSN}\end{matrix}\right\}$ を使用する.

(7) **使用例**

(a) 問 題

$$
A = \begin{bmatrix}
5 & 2 & 1 & & & & & & & \\
2 & 6 & 3 & 1 & & & 0 & & & \\
1 & 3 & 6 & 3 & 1 & & & & & \\
 & 1 & 3 & 6 & 3 & 1 & & & & \\
 & & 1 & 3 & 6 & 3 & 1 & & & \\
 & & & 1 & 3 & 6 & 3 & 1 & & \\
 & & & & 1 & 3 & 6 & 3 & 1 & \\
 & & 0 & & & 1 & 3 & 6 & 3 & 1 \\
 & & & & & & 1 & 3 & 6 & 2 \\
 & & & & & & & 1 & 2 & 5
\end{bmatrix}
$$

の固有値を小さい順に3個とそれに対応する固有ベクトルを求める.

(b) 入力データ

行列 A, LNA=11, N=10, MB=2, EPS=−1.0, M=3, LNV=10, ISW=−1

(c) 主プログラム

```
      PROGRAM BCSBSS
! *** EXAMPLE OF DCSBSS ***
      IMPLICIT REAL(8) (A-H,O-Z)
      CHARACTER*80 FMT
      PARAMETER ( LMA = 11, LNV = 11 )
      DIMENSION A(LMA,LMA), E(LMA), VE(LNV,LNV), IW1(LMA), W1(36*LMA)
!
      READ(5,*) N, MB, M
      DO 10 J=1, MB+1
         READ(5,*) (A(J,I), I=MB-J+2, N)
   10 CONTINUE
!
      WRITE(6,1000) N, MB, M
      DO 20 J=1, MB+1
         WRITE(FMT,1100) (MB-J+1)*7+1, N-MB+J-1
         WRITE(6,FMT) (A(J,I), I=MB-J+2, N)
   20 CONTINUE
!
      ISW = -1
      EPS = -1.0D0
!
      CALL DCSBSS(A,LMA,N,MB,EPS,E,M,VE,LNV,ISW,IW1,W1,IERR)
!
      WRITE(6,1200)  IERR
!
      DO 40 K=1, M-3, 4
         WRITE(6,1300) ('EIGENVALUE ', I=1, 4)
         WRITE(6,1400) (E(I), I=K, K+3)
         WRITE(6,1300) ('EIGENVECTOR', I=1, 4)
         DO 30 J=1, N
            WRITE(6,1500) (VE(J,I), I=K, K+3)
   30    CONTINUE
   40 CONTINUE
      IF(MOD(M,4).NE.0) THEN
         WRITE(6,1300) ('EIGENVALUE ', I=M/4*4+1, M)
         WRITE(6,1400) (E(I), I=M/4*4+1, M)
         WRITE(6,1300) ('EIGENVECTOR', I=M/4*4+1, M)
         DO 50 J=1, N
            WRITE(6,1500) (VE(J,I), I=M/4*4+1, M)
   50    CONTINUE
      ENDIF
      STOP
!
 1000 FORMAT(' ',/,/,&
             ' ***  DCSBSS  ***',/,/,&
             '  **  INPUT  **',/,/,&
             '      N = ', I2,/,/,&
             '      BAND WIDTH = ', I2,/,/,&
             '      M = ', I2,/,/,&
             '      INPUT MATRIX A',/)
 1100 FORMAT('( ,', I3,'X,', I2,'(F7.1))')
 1200 FORMAT(' ',/,/,&
             '  **  OUTPUT  **',/,/,&
             '      IERR = ', I4)
 1300 FORMAT(' ',/,1X, 4(5X, A11, 2X))
 1400 FORMAT(3X, 4(2X, 1PD14.7, 2X))
 1500 FORMAT(2X, 4(F14.8, 4X))
      END
```

(d) 出力結果

```
***  DCSBSS  ***

 **  INPUT  **

     N = 10

     BAND WIDTH =  2

     M =  3

     INPUT MATRIX A

                  1.0    1.0    1.0    1.0    1.0    1.0    1.0    1.0
           2.0    3.0    3.0    3.0    3.0    3.0    3.0    3.0    2.0
    5.0    6.0    6.0    6.0    6.0    6.0    6.0    6.0    6.0    5.0

 **  OUTPUT  **

     IERR =    0

     EIGENVALUE        EIGENVALUE        EIGENVALUE
     1.8799058D+00     1.8926451D+00     2.2578112D+00

     EIGENVECTOR       EIGENVECTOR       EIGENVECTOR
    -0.01172823        0.05600768        0.12429215
    -0.20382326       -0.01048668       -0.41411041
     0.44423970       -0.15306238        0.48738830
    -0.46949161        0.39024431       -0.16106987
     0.20136345       -0.56659903       -0.22265032
     0.20136345        0.56659903        0.22265032
    -0.46949161       -0.39024431        0.16106987
     0.44423970        0.15306238       -0.48738830
    -0.20382326        0.01048668        0.41411041
    -0.01172823       -0.05600768       -0.12429215
```

4.9.4 DCSBSN, RCSBSN 実対称バンド行列の固有値

(1) **機 能**

実対称バンド行列 A(対称バンド型) の固有値をギブンス法, 無平方根 QR 法またはバイセクション法により大きい方から m 個, または小さい方から m 個求める.

(2) **使用法**

倍精度サブルーチン:
　CALL DCSBSN (A, LMA, N, MB, EPS, E, M, ISW, W1, IERR)
単精度サブルーチン:
　CALL RCSBSN (A, LMA, N, MB, EPS, E, M, ISW, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: { 32 ビット整数版では INTEGER(4)
　 64 ビット整数版では INTEGER(8) }

項番	引数名	型	大きさ	入出力	内　容
1	A	D R	LMA, N	入 力	実対称バンド行列 A(対称バンド型) (注意事項 (a) 参照)
				出 力	入力時の内容は保存されない
2	LMA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A の次数
4	MB	I	1	入 力	行列 A のバンド幅
5	EPS	D R	1	入 力	固有値の収束判定のための絶対誤差の上限を与えるパラメータ (注意事項 (d) 参照)
6	E	D R	M	出 力	固有値
7	M	I	1	入 力	求めたい固有値の個数 m
8	ISW	I	1	入 力	処理スイッチ ISW $\geq$ 0: 大きい方から固有値を求める. ISW $<$ 0: 小さい方から固有値を求める.
9	W1	D R	5×N	ワーク	作業領域
10	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{M} \leq \mathrm{N}$

(b) $0 \leq \mathrm{MB} < \mathrm{N}$

(c) $\mathrm{MB} < \mathrm{LMA}$

(5) エラーインディケータ

IERR の値	意　　　味	処　理　内　容
0	正常終了.	
1000	N $=1$ であった.	E$(1) \leftarrow$ A$(1,1)$ とする.
1100	MB $=0$ であった.	E$(i) \leftarrow$ A$(1,i)$ $(i=1,\ldots,$N$)$ とする.
3000	制限条件 (a), (b) または (c) を満足しなかった.	処理を打ち切る.

(6) 注意事項

(a) 配列 A には, 実対称バンド行列を対称バンド型 ((MB+1) 行 N 列) に圧縮して格納する (付録 B 参照).

(b) 固有値は ISW≥0 のときには大きい順に, ISW<0 のときには小さい順に格納される.

(c) 固有値は無平方根 QR 法とバイセクション法を内部で適切に切り分けて計算している.

(d) EPS ≤ 0 のときは内部で自動的に最適値を設定する. 通常は自動設定されるように負の値を設定されたい. なお, EPS はバイセクション法で固有値を求めるときに使用される.

4.9.5 DCSBFF, RCSBFF
実対称バンド行列の固有値・固有ベクトル

(1) 機 能

実対称バンド行列 A(対称バンド型) の固有値とそれに対応する固有ベクトルをサブスペース法により, 固有値の絶対値が小さい方から m 個, または大きい方から m 個求める.

(2) 使用法

倍精度サブルーチン:

CALL DCSBFF (A, LMA, N, MB, M, ITOL, NITE, E, VE, LNV, MST, IS1, IS2, W1, IW1, IERR)

単精度サブルーチン:

CALL RCSBFF (A, LMA, N, MB, M, ITOL, NITE, E, VE, LNV, MST, IS1, IS2, W1, IW1, IERR)

(3) 引 数

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: { 32 ビット整数版では INTEGER(4) / 64 ビット整数版では INTEGER(8) }

項番	引数名	型	大きさ	入出力	内　容
1	A	{D R}	LMA, N	入 力	実対称バンド行列 A(対称バンド型) (付録 B 参照)
				出 力	入力時の内容は保存されない
2	LMA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A の次数
4	MB	I	1	入 力	行列 A のバンド幅
5	M	I	1	入 力	求めたい固有値の個数 m
6	ITOL	I	1	入 力	収束判定用トレランス (注意事項 (b) 参照)
7	NITE	I	1	入 力	最大反復回数 (注意事項 (d) 参照)
8	E	{D R}	内容参照	出 力	固有値 **大きさ:** $(\min(2\times m,\mathrm{N},m+8))$
9	VE	{D R}	内容参照	出 力	各固有値に対応する固有ベクトル (列ベクトル) **大きさ:** $((\mathrm{LNV},\min(2\times m,\mathrm{N},m+8)))$
10	LNV	I	1	入 力	配列 VE の整合寸法
11	MST	I	1	出 力	計算されなかった固有値数 (注意事項 (e) 参照)
12	IS1	I	1	入 力	処理スイッチ IS1 ≤ 0: 固有値を絶対値の小さい方から求める. IS1 > 0: 固有値を絶対値の大きい方から求める.
13	IS2	I	1	入 力	スツルム列チェックスイッチ IS2 ≤ 0: チェックを行わない. IS2 > 0: チェックを行う.

項番	引数名	型	大きさ	入出力	内　　　容
14	W1	$\left\{\begin{array}{c}\mathrm{D}\\\mathrm{R}\end{array}\right\}$	内容参照	ワーク	作業領域 **大きさ**: IS2 ≤ 0 の時： $\mathrm{N}\times q+q\times q+2\times q+\mathrm{N}$ IS2 > 0 の時： $\mathrm{N}\times q+q\times q+2\times q+\mathrm{N}+\mathrm{N}\times(\mathrm{MB}+1)$ ただし, ここで $q=\min(2\times m,\mathrm{N},m+8)$ である.
15	IW1	I	N	ワーク	作業領域
16	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \le \mathrm{LNV}$

(b) $0 \le \mathrm{MB} < \mathrm{N}$

(c) $\mathrm{MB} < \mathrm{LMA}$

(d) $0 < \mathrm{M} \le \mathrm{N}$

(5) **エラーインディケータ**

IERR の値	意　　　味	処 理 内 容
0	正常終了.	
1000	$\mathrm{N}=1$ であった.	$\mathrm{E}(1) \leftarrow \mathrm{A}(1,1)$, $\mathrm{VE}(1,1) \leftarrow 1.0$ とする.
3000	制限条件 (a), (b), (c) または (d) を満足しなかった.	処理を打ち切る.
4000	処理途中でエラーが起きた.	
$5000+i$	指定された回数以内で収束しなかった.	i 番目までの固有値, 固有ベクトルは求められている. 処理を打ち切る.

(6) **注意事項**

(a) このサブルーチンは, 求めたい固有値数が行列の次数と比較してごく小さい場合 (m≪N) に有効である. それ以外の場合には, 他のサブルーチン 4.9.1 $\left\{\begin{array}{c}\text{DCSBAA}\\\text{RCSBAA}\end{array}\right\}$, 4.9.3 $\left\{\begin{array}{c}\text{DCSBSS}\\\text{RCSBSS}\end{array}\right\}$等を使用した方がよい.

(b) このサブルーチンでは, 以下の条件が満たされた時に固有値は収束したとみなす. この時, 固有ベクトルは ITOL/2 以上の精度を持つ.

$$\left|\frac{a_i^n - a_i^{n-1}}{a_i^n}\right| \le 10.0^{-\mathrm{ITOL}} \qquad (a_i^n\text{:n 回反復後の第 i 番目の固有値})$$

ITOL の入力値として 0 以下または $-$ LOG10 (ε) より大きい数が与えられた場合, 内部で最適値が採用される. (ε:誤差判定のための単位).

(c) 固有値は IS1≤0 のときには絶対値の小さい順に, IS1>0 のときには大きい順に配列 E に格納される.

(d) NITE の入力値として 0 以下の数が与えられた場合は, 既定値として 20 を採用する.

(e) このサブルーチンには, 算出された固有値に対し, スツルム列の性質を利用したチェックを行う機能がある. これにより, 計算されなかった固有値が算出されるが, この時演算回数は N × MB^2 程度増加する.

例

6, 5, 3, 2, 1 を固有値として持つような固有値問題に対して, 絶対値最小のものより 3 個の固有値を求めたとする. この時, 固有値の解として 5, 2, 1 が求められたとすると, 3 が解として求められなかったので MST には 1 が返される.

なお, この機能は固有値がすべて正の時のみ有効である.

(7) 使用例

(a) 問 題

$$A = \begin{bmatrix} 611 & 196 & -192 & 407 & -8 & 0 & 0 & 0 \\ 196 & 899 & 113 & -192 & -71 & -43 & 0 & 0 \\ -192 & 113 & 899 & 196 & 61 & 49 & 8 & 0 \\ 407 & -192 & 196 & 611 & 8 & 44 & 59 & -23 \\ -8 & -72 & 61 & 8 & 411 & -599 & 208 & 208 \\ 0 & -43 & 49 & 44 & -599 & 411 & 208 & 208 \\ 0 & 0 & 8 & 59 & 208 & 208 & 99 & -911 \\ 0 & 0 & 0 & -23 & 208 & 208 & -911 & 99 \end{bmatrix}$$

の固有値とそれに対応する固有ベクトルを, 固有値の絶対値最小のものから 2 個求める.

(b) 入力データ

行列 A, LMA=11, N=8, MB=4, M=2, LNV=10

(c) 主プログラム

```
      PROGRAM BCSBFF
! *** EXAMPLE OF DCSBFF ***
      IMPLICIT REAL(8)(A-H,O-Z)
      CHARACTER*80 FMT
      PARAMETER ( LMA=11, LNV=10, LN=10, LNQ=10 )
      PARAMETER ( LW=LNQ*(LN+LNQ+2)+LN*(LN+1) )
      DIMENSION A(LMA,LN), E(LN), VE(LNV,LNQ), W1(LW), IW1(LN)
!
      READ(5,*) N, MB, M
      DO 10 J=1, MB+1
         READ(5,*) (A(J,I), I=MB-J+2, N)
   10 CONTINUE
!
      WRITE(6,1000) N, MB, M
      DO 20 J=1, MB+1
         WRITE(FMT,1100) (MB-J+1)*8+2, N-MB+J-1
         WRITE(6,FMT) (A(J,I), I=MB-J+2, N)
   20 CONTINUE
!
      CALL DCSBFF(A,LMA,N,MB,M,0,0,E,VE,LNV,MST,0,1,W1,IW1,IERR)
!
      WRITE(6,1200) IERR
!
      DO 40 K=1, M-3, 4
         WRITE(6,1300) ('EIGENVALUE ', I=1, 4)
         WRITE(6,1400) (E(I), I=K, K+3)
         WRITE(6,1300) ('EIGENVECTOR', I=1, 4)
         DO 30 J=1, N
            WRITE(6,1500) (VE(J,I), I=K, K+3)
   30    CONTINUE
   40 CONTINUE
      IF(MOD(M,4).NE.0) THEN
         WRITE(6,1300) ('EIGENVALUE ', I=M/4*4+1, M)
         WRITE(6,1400) (E(I), I=M/4*4+1, M)
         WRITE(6,1300) ('EIGENVECTOR', I=M/4*4+1, M)
         DO 50 J=1, N
            WRITE(6,1500) (VE(J,I), I=M/4*4+1, M)
   50    CONTINUE
      ENDIF
      WRITE(6,1600) MST
      STOP
!
 1000 FORMAT(' ',/,/,&
             ' ***  DCSBFF  ***',/,/,&
             '  **  INPUT  **',/,/,&
             '      N = ', I2,/,/,&
             '      BAND WIDTH = ', I2,/,/,&
             '      M = ', I2,/,/,&
```

```
            '      INPUT MATRIX A',/)
1100 FORMAT('( ,', I3,'X,', I2,'(F8.1))')
1200 FORMAT(' ',/,/,&
            '  **  OUTPUT  **',/,/,&
            '      IERR = ', I4)
1300 FORMAT(' ',/,1X, 4(5X, A11, 2X))
1400 FORMAT(3X, 4(2X, 1PD14.7, 2X))
1500 FORMAT(2X, 4(F14.8, 4X))
1600 FORMAT(' ',/,1X,'     MISSED EIGENVALUES = ', I2)
     END
```

(d) 出力結果

```
***  DCSBFF  ***

 **  INPUT  **

     N =  8

     BAND WIDTH =  4

     M =  2

     INPUT MATRIX A

                                     -8.0   -43.0     8.0   -23.0
                            407.0   -71.0    49.0    59.0   208.0
                  -192.0  -192.0    61.0    44.0   208.0   208.0
           196.0   113.0   196.0     8.0  -599.0   208.0  -911.0
    611.0   899.0   899.0   611.0   411.0   411.0    99.0    99.0


 **  OUTPUT  **

     IERR =    0

     EIGENVALUE        EIGENVALUE
    -5.9306717D+00      2.2278823D+01

     EIGENVECTOR       EIGENVECTOR
     0.42488478        0.46703726
    -0.26681179       -0.17958149
     0.26654024        0.17944823
    -0.39884494       -0.49603483
    -0.45953441        0.42706413
    -0.43824501        0.44863165
    -0.22420209        0.22834768
    -0.25429564        0.18862045

     MISSED EIGENVALUES =  1
```

4.10 実対称 3 重対角行列 (ベクトル型)

4.10.1 DCSTAA, RCSTAA
実対称 3 重対角行列の全固有値・全固有ベクトル

(1) **機 能**

実対称 3 重対角行列 A(ベクトル型) の全固有値とそれに対応する固有ベクトルを QR 法により求める.

(2) **使用法**

倍精度サブルーチン:

CALL DCSTAA (D, N, SD, E, VE, LNV, IERR)

単精度サブルーチン:

CALL RCSTAA (D, N, SD, E, VE, LNV, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: { 32 ビット整数版では INTEGER(4)
　 64 ビット整数版では INTEGER(8) }

項番	引数名	型	大きさ	入出力	内　容
1	D	{D R}	N	入 力	実対称 3 重対角行列 A の対角成分
2	N	I	1	入 力	行列 A の次数
3	SD	{D R}	N	入 力	実対称 3 重対角行列 A の副対角成分
				出 力	入力時の内容は保存されない
4	E	{D R}	N	出 力	固有値
5	VE	{D R}	LNV, N	出 力	各固有値に対応する固有ベクトル (列ベクトル)
6	LNV	I	1	入 力	配列 VE の整合寸法
7	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNV}$

(5) **エラーインディケータ**

IERR の値	意　味	処 理 内 容
0	正常終了.	
1000	N = 1 であった.	E(1) ← D(1), VE(1, 1) ← 1.0 とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
$5000 + i$	固有値, 固有ベクトルを求める段階で収束しなかった. $(1 \leq i \leq \mathrm{N})$	$\mathrm{E}(1), \cdots, \mathrm{E}(i-1)$ にそれまでに求まった固有値, VE にそれに対応する固有ベクトルが入る (ただし順不同).

(6) **注意事項**

(a) 1 次元配列 D, SD に, 実対称 3 重対角行列のそれぞれ対角成分, 副対角成分を格納する. SD(N) は任意でよい (付録 B を参照).

(b) 固有値は小さい順に格納される.

(c) 固有ベクトルは正規直交系である.

(d) 固有ベクトルを必要としないときは, 4.10.2 $\left\{\begin{matrix}\text{DCSTAN}\\ \text{RCSTAN}\end{matrix}\right\}$ を使用する.

(7) **使用例**

(a) 問 題

$$A = \begin{bmatrix} 4 & 1 & 0 & 0 \\ 1 & 3 & 1 & 0 \\ 0 & 1 & 3 & 1 \\ 0 & 0 & 1 & 4 \end{bmatrix}$$

の全固有値とそれに対応する固有ベクトルを求める.

(b) 入力データ

行列 A の対角成分 D, N=4, 行列 A の副対角成分 SD, LNV=10

(c) 主プログラム

```
      PROGRAM BCSTAA
! *** EXAMPLE OF DCSTAA ***
      IMPLICIT REAL(8) (A-H,O-Z)
      PARAMETER (LNV = 10)
      DIMENSION D(LNV), SD(LNV), E(LNV), VE(LNV,LNV)
!
      READ(5,*) N
      READ(5,*) (D(I), I=1, N)
      READ(5,*) (SD(I), I=1, N-1)
!
      WRITE(6,1000) N
      WRITE(6,1100) ’DIAGONAL   ’, (D(I), I=1, N)
      WRITE(6,1100) ’SUBDIAGONAL’, (SD(I), I=1, N-1)
!
      CALL DCSTAA(D,N,SD,E,VE,LNV,IERR)
!
      WRITE(6,1200) IERR
!
      DO 20 K=1, N-3, 4
         WRITE(6,1300) (’EIGENVALUE ’, I=1, 4)
         WRITE(6,1400) (E(I), I=K, K+3)
         WRITE(6,1300) (’EIGENVECTOR’, I=1, 4)
         DO 10 J=1, N
            WRITE(6,1500) (VE(J,I), I=K, K+3)
   10    CONTINUE
   20 CONTINUE
      IF(MOD(N,4).NE.0) THEN
         WRITE(6,1300) (’EIGENVALUE ’, I=N/4*4+1, N)
         WRITE(6,1400) (E(I), I=N/4*4+1, N)
         WRITE(6,1300) (’EIGENVECTOR’, I=N/4*4+1, N)
         DO 30 J=1, N
            WRITE(6,1500) (VE(J,I), I=N/4*4+1, N)
   30    CONTINUE
      ENDIF
      STOP
!
 1000 FORMAT(’ ’,/,/,&
             ’ ***  DCSTAA  ***’,/,/,&
             ’  **  INPUT  **’,/,/,&
             ’      N = ’, I2,/,/,&
             ’      INPUT MATRIX A’)
 1100 FORMAT(’ ’,/,6X, A11,/,&
             5X, 11(F7.1),/)
 1200 FORMAT(’ ’,/,/,&
             ’  **  OUTPUT  **’,/,/,&
             ’      IERR = ’, I4)
 1300 FORMAT(’ ’,/,1X, 4(5X, A11, 2X))
 1400 FORMAT(3X, 4(2X, 1PD14.7, 2X))
 1500 FORMAT(2X, 4(F14.8, 4X))
      END
```

(d) 出力結果

```
***  DCSTAA  ***

 **  INPUT  **

     N =  4

     INPUT MATRIX A

     DIAGONAL
        4.0    3.0    3.0    4.0

     SUBDIAGONAL
        1.0    1.0    1.0


 **  OUTPUT  **

     IERR =    0

     EIGENVALUE         EIGENVALUE         EIGENVALUE         EIGENVALUE
     1.5857864D+00      3.0000000D+00      4.4142136D+00      5.0000000D+00

     EIGENVECTOR        EIGENVECTOR        EIGENVECTOR        EIGENVECTOR
    -0.27059805        -0.50000000        -0.65328148        -0.50000000
     0.65328148         0.50000000        -0.27059805        -0.50000000
    -0.65328148         0.50000000         0.27059805        -0.50000000
     0.27059805        -0.50000000         0.65328148        -0.50000000
```

4.10.2 DCSTAN, RCSTAN
実対称 3 重対角行列の全固有値

(1) **機 能**

実対称 3 重対角行列 A(ベクトル型) の全固有値を無平方根 QR 法により求める.

(2) **使用法**

倍精度サブルーチン:
CALL DCSTAN (D, N, SD, E, IERR)
単精度サブルーチン:
CALL RCSTAN (D, N, SD, E, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$

項番	引数名	型	大きさ	入出力	内　　容
1	D	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	N	入 力	実対称 3 重対角行列 A の対角成分
2	N	I	1	入 力	行列 A の次数
3	SD	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	N	入 力	実対称 3 重対角行列 A の副対角成分
				出 力	入力時の内容は保存されない
4	E	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	N	出 力	固有値
5	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $\mathrm{N} > 0$

(5) **エラーインディケータ**

IERR の値	意　　味	処　理　内　容
0	正常終了.	
1000	$\mathrm{N} = 1$ であった.	$\mathrm{E}(1) \leftarrow \mathrm{D}(1)$ とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
$5000 + i$	固有値を求める段階で収束しなかった. $(1 \le i \le \mathrm{N})$	$\mathrm{E}(1), \cdots, \mathrm{E}(i-1)$ にそれまでに求まった固有値が入る (ただし順不同).

(6) **注意事項**

(a) 1 次元配列 D, SD に, 実対称 3 重対角行列のそれぞれ対角成分, 副対角成分を格納する. SD (N) は任意でよい (付録 B を参照).

(b) 固有値は小さい順に格納される.

4.10.3 DCSTSS, RCSTSS
実対称 3 重対角行列の固有値・固有ベクトル

(1) **機 能**

実対称 3 重対角行列 A(ベクトル型) の固有値を無平方根 QR 法またはバイセクション法により, 大きい方から m 個, または小さい方から m 個求めそれに対応する固有ベクトルを逆反復法により求める.

(2) **使用法**

倍精度サブルーチン:
　CALL DCSTSS (D, N, SD, EPS, E, M, VE, LNV, ISW, IW1, W1, IERR)
単精度サブルーチン:
　CALL RCSTSS (D, N, SD, EPS, E, M, VE, LNV, ISW, IW1, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

$$\text{I:} \left\{ \begin{array}{l} \text{32 ビット整数版では INTEGER(4)} \\ \text{64 ビット整数版では INTEGER(8)} \end{array} \right\}$$

項番	引数名	型	大きさ	入出力	内　容
1	D	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	N	入 力	実対称 3 重対角行列 A の対角成分
2	N	I	1	入 力	行列 A の次数
3	SD	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	N	入 力	実対称 3 重対角行列 A の副対角成分
4	EPS	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	1	入 力	固有値の収束判定のための絶対誤差の上限を与えるパラメータ (注意事項 (d) 参照)
5	E	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	M	出 力	固有値
6	M	I	1	入 力	求めたい固有値の個数 m
7	VE	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	LNV, M	出 力	各固有値に対応する固有ベクトル (列ベクトル)
8	LNV	I	1	入 力	配列 VE の整合寸法
9	ISW	I	1	入 力	処理スイッチ ISW $\geq$ 0: 大きい方から固有値を求める. ISW $<$ 0: 小さい方から固有値を求める.
10	IW1	I	M	出 力	固有ベクトルフラグ (注意事項 (e) 参照)
11	W1	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	6 × N	ワーク	作業領域
12	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNV}$

(b) $0 < \mathrm{M} \leq \mathrm{N}$

(5) **エラーインディケータ**

IERR の値	意　　　　　　味	処　理　内　容
0	正常終了.	
1000	$\mathrm{N} = 1$ であった.	$\mathrm{E}(1) \leftarrow \mathrm{D}(1)$, $\mathrm{VE}(1, 1) \leftarrow 1.0$ とする.
2000	固有ベクトルを求める逆反復で最大反復回数をこえた.	固有ベクトルに一部精度の低いものがあるが, 処理は続行する (注意事項 (e) 参照).
3000	制限条件 (a) または (b) を満足しなかった.	処理を打ち切る.

(6) **注意事項**

(a) 1 次元配列 D, SD に, 実対称 3 重対角行列のそれぞれ対角成分, 副対角成分を格納する. SD (N) は任意でよい (付録 B を参照).

(b) 固有値は ISW≥0 のときには大きい順に, ISW<0 のときには小さい順に格納される.

(c) 固有値は無平方根 QR 法とバイセクション法を内部で適切に切り分けて計算している.

(d) $\mathrm{EPS} \leq 0$ のときは内部で自動的に最適値を設定する. 通常は自動設定されるように負の値を設定されたい. なお, EPS はバイセクション法で固有値を求めるときに使用される.

(e) 逆反復法で最大反復回数をこえた場合 (IERR=2000 出力時) について

- $\mathrm{IW1}(i) = 0$ の場合 :
  i 番目の固有ベクトル計算は正常終了している.
- $\mathrm{IW1}(i) \neq 0$ の場合 :
  i 番目の固有ベクトル計算は収束条件が満たされず, 固有ベクトルの精度は低い.
  この場合 $\mathrm{IW1}(i)$ は, 反復回数が設定される.

なお正常終了時 (IERR=0 出力時) は, $\mathrm{IW1}(i) \leftarrow 0$ が設定される.

(f) 固有ベクトルは正規直交系である.

(g) 固有ベクトルを必要としないときは, 4.10.4 $\begin{Bmatrix} \mathrm{DCSTSN} \\ \mathrm{RCSTSN} \end{Bmatrix}$ を使用する.

(7) **使用例**

(a) 問 題

$$A = \begin{bmatrix} 5 & 3 & & & & & & & & \\ 3 & 2 & 3 & & & & & & & \\ & 3 & 2 & 3 & & & & 0 & & \\ & & 3 & 2 & 3 & & & & & \\ & & & 3 & 2 & 3 & & & & \\ & & & & 3 & 2 & 3 & & & \\ & & & & & 3 & 2 & 3 & & \\ & & 0 & & & & 3 & 2 & 3 & \\ & & & & & & & 3 & 2 & 3 \\ & & & & & & & & 3 & 5 \end{bmatrix}$$

の固有値を大きい順に 2 個とそれに対応する固有ベクトルを求める.

(b) 入力データ

行列 A の対角成分 D, N=10, 行列 A の副対角成分 SD, EPS=−1.0, M=2, LNV=10, ISW=1

(c) 主プログラム

```
      PROGRAM BCSTSS
! *** EXAMPLE OF DCSTSS ***
      IMPLICIT REAL(8) (A-H,O-Z)
      PARAMETER ( LNV = 10 )
      DIMENSION D(LNV), SD(LNV), E(LNV), VE(LNV,LNV),&
                IW1(LNV), W1(6*LNV)
!
      READ(5,*) N, M
      READ(5,*) (D(I), I=1, N)
      READ(5,*) (SD(I), I=1, N-1)
!
      WRITE(6,1000) N, M
      WRITE(6,1100) ’DIAGONAL   ’, (D(I), I=1, N)
      WRITE(6,1100) ’SUBDIAGONAL’, (SD(I), I=1, N-1)
!
      ISW = 1
      EPS = -1.0D0
!
      CALL DCSTSS(D,N,SD,EPS,E,M,VE,LNV,ISW,IW1,W1,IERR)
!
      WRITE(6,1200) IERR
!
      DO 20 K=1, M-3, 4
         WRITE(6,1300) (’EIGENVALUE ’, I=1, 4)
         WRITE(6,1400) (E(I), I=K, K+3)
         WRITE(6,1300) (’EIGENVECTOR’, I=1, 4)
         DO 10 J=1, N
            WRITE(6,1500) (VE(J,I), I=K, K+3)
   10    CONTINUE
   20 CONTINUE
      IF(MOD(M,4).NE.0) THEN
         WRITE(6,1300) (’EIGENVALUE ’, I=M/4*4+1, M)
         WRITE(6,1400) (E(I), I=M/4*4+1, M)
         WRITE(6,1300) (’EIGENVECTOR’, I=M/4*4+1, M)
         DO 30 J=1, N
            WRITE(6,1500) (VE(J,I), I=M/4*4+1, M)
   30    CONTINUE
      ENDIF
      STOP
!
 1000 FORMAT(’ ’,/,/,&
             ’ ***  DCSTSS  ***’,/,/,&
             ’  **  INPUT  **’,/,/,&
             ’      N = ’, I2,/,/,&
             ’      M = ’, I2,/,/,&
             ’      INPUT MATRIX A’)
 1100 FORMAT(’ ’,/,6X, A11,/,&
             5X, 11(F7.1),/)
 1200 FORMAT(’ ’,/,/,&
             ’  **  OUTPUT  **’,/,/,&
             ’      IERR = ’, I4)
 1300 FORMAT(’ ’,/,1X, 4(5X, A11, 2X))
 1400 FORMAT(3X, 4(2X, 1PD14.7, 2X))
 1500 FORMAT(2X, 4(F14.8, 4X))
      END
```

(d) 出力結果

```
 ***  DCSTSS  ***

  **  INPUT  **

      N = 10

      M =  2

      INPUT MATRIX A

      DIAGONAL
         5.0    2.0    2.0    2.0    2.0    2.0    2.0    2.0    2.0    5.0

      SUBDIAGONAL
         3.0    3.0    3.0    3.0    3.0    3.0    3.0    3.0    3.0

  **  OUTPUT  **

      IERR =    0

      EIGENVALUE        EIGENVALUE
      8.0000000D+00     7.7063391D+00

      EIGENVECTOR       EIGENVECTOR
      0.31622777       -0.44170765
      0.31622777       -0.39847023
      0.31622777       -0.31622777
      0.31622777       -0.20303072
      0.31622777       -0.06995962
      0.31622777        0.06995962
      0.31622777        0.20303072
      0.31622777        0.31622777
      0.31622777        0.39847023
      0.31622777        0.44170765
```

4.10.4 DCSTSN, RCSTSN
実対称 3 重対角行列の固有値

(1) **機 能**

実対称 3 重対角行列 A(ベクトル型) の固有値を無平方根 QR 法またはバイセクション法により, 大きい方から m 個, または小さい方から m 個求める.

(2) **使用法**

倍精度サブルーチン:
CALL DCSTSN (D, N, SD, EPS, E, M, ISW, W1, IERR)
単精度サブルーチン:
CALL RCSTSN (D, N, SD, EPS, E, M, ISW, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: $\left\{\begin{array}{l}\text{32 ビット整数版では INTEGER(4)}\\ \text{64 ビット整数版では INTEGER(8)}\end{array}\right\}$

項番	引数名	型	大きさ	入出力	内　　容
1	D	$\begin{Bmatrix}\text{D}\\ \text{R}\end{Bmatrix}$	N	入 力	実対称 3 重対角行列 A の対角成分
2	N	I	1	入 力	行列 A の次数
3	SD	$\begin{Bmatrix}\text{D}\\ \text{R}\end{Bmatrix}$	N	入 力	実対称 3 重対角行列 A の副対角成分
4	EPS	$\begin{Bmatrix}\text{D}\\ \text{R}\end{Bmatrix}$	1	入 力	固有値の収束判定のための絶対誤差の上限を与えるパラメータ (注意事項 (d) 参照)
5	E	$\begin{Bmatrix}\text{D}\\ \text{R}\end{Bmatrix}$	M	出 力	固有値
6	M	I	1	入 力	求めたい固有値の個数 m
7	ISW	I	1	入 力	処理スイッチ ISW $\geq$ 0: 大きい方から固有値を求める. ISW $<$ 0: 小さい方から固有値を求める.
8	W1	$\begin{Bmatrix}\text{D}\\ \text{R}\end{Bmatrix}$	3×N	ワーク	作業領域
9	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \text{M} \leq \text{N}$

(5) エラーインディケータ

IERR の値	意　　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	E(1) ← D(1) とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.

(6) 注意事項

(a) 1 次元配列 D, SD に, 実対称 3 重対角行列のそれぞれ対角成分, 副対角成分を格納する. SD (N) は任意でよい (付録 B を参照).

(b) 固有値は ISW≥0 のときには大きい順に, ISW<0 のときには小さい順に格納される.

(c) 固有値は無平方根 QR 法とバイセクション法を内部で適切に切り分けて計算している.

(d) EPS $\leq$ 0 のときは内部で自動的に最適値を設定する. 通常は自動設定されるように負の値を設定されたい. なお, EPS はバイセクション法で固有値を求めるときに使用される.

4.10.5 DCSTEE, RCSTEE
実対称 3 重対角行列の固有値・固有ベクトル (区間指定)

(1) **機 能**

実対称 3 重対角行列 A(ベクトル型) の指定した区間の固有値をバイセクション法により, 昇順で m 個, もしくは降順で m 個求め, それに対応する固有ベクトルを逆反復法により求める.

(2) **使用法**

倍精度サブルーチン:

CALL DCSTEE (D, N, SD, EPS, E, M, E1, E2, VE, LNV, IW1, W1, IERR)

単精度サブルーチン:

CALL RCSTEE (D, N, SD, EPS, E, M, E1, E2, VE, LNV, IW1, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: { 32 ビット整数版では INTEGER(4)
　 64 ビット整数版では INTEGER(8) }

項番	引数名	型	大きさ	入出力	内　容
1	D	D R	N	入 力	実対称 3 重対角行列 A の対角成分
2	N	I	1	入 力	行列 A の次数
3	SD	D R	N	入 力	実対称 3 重対角行列 A の副対角成分
4	EPS	D R	1	入 力	固有値の収束判定のための絶対誤差の上限を与えるパラメータ (注意事項 (b) 参照)
5	E	D R	M	出 力	固有値
6	M	I	1	入 力	求めたい固有値の個数 m
				出 力	求めた固有値の個数
7	E1	D R	1	入 力	E1 < E2: E1 から昇順で固有値を求める. (E2 は, 上限) E1 > E2: E1 から降順で固有値を求める. (E2 は, 下限) (注意事項 (c)(d) 参照)
8	E2	D R	1	入 力	
9	VE	D R	LNV, M	出 力	各固有値に対応する固有ベクトル (列ベクトル)
10	LNV	I	1	入 力	配列 VE の整合寸法
11	IW1	I	M	出 力	固有ベクトルフラグ (注意事項 (e) 参照)
12	W1	D R	6 × N	ワーク	作業領域
13	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNV}$

(b) $0 < \mathrm{M} \leq \mathrm{N}$

(5) **エラーインディケータ**

IERR の値	意　　味	処 理 内 容
0	正常終了.	
1000	$\mathrm{N} = 1$ であった.	$\mathrm{E}(1) \leftarrow \mathrm{D}(1)$, $\mathrm{VE}(1,1) \leftarrow 1.0$ とする.
1500	区間 [E1, E2] に存在する固有値は M 個未満であった.	区間 [E1, E2] 内にある全ての固有値, 固有ベクトルを求める. 引数 M には求めた固有値の個数を出力する.
2000	固有ベクトルを求める逆反復で最大反復回数をこえた.	固有ベクトルに一部精度の低いものがあるが, 処理は続行する (注意事項 (e) 参照).
3000	制限条件 (a) または (b) を満足しなかった.	処理を打ち切る.

(6) **注意事項**

(a) 1 次元配列 D, SD に, 実対称 3 重対角行列のそれぞれ対角成分, 副対角成分を格納する. SD (N) は任意でよい (付録 B を参照).

(b) $\mathrm{EPS} \leq 0$ のときは内部で自動的に最適値を設定する. 通常は自動設定されるように負の値を設定されたい. なお, EPS はバイセクション法で固有値を求めるときに使用される.

(c) 固有値は E1<E2 のときには小さい順, E1>E2 のときには大きい順に格納される.

(d) E1=E2 のとき, 区間 [E1-EPS, E1+EPS] にある固有値が求まる. しかし, 通常は E1≠E2 となるように設定されたい.

(e) 逆反復法で最大反復回数をこえた場合 (IERR=2000 出力時) について

- $\mathrm{IW1}(i) = 0$ の場合：
  i 番目の固有ベクトル計算は正常終了している.
- $\mathrm{IW1}(i) \neq 0$ の場合：
  i 番目の固有ベクトル計算は収束条件が満たされず, 固有ベクトルの精度は低い.
  この場合 $\mathrm{IW1}(i)$ は, 反復回数が設定される.

なお正常終了時 (IERR=0 出力時) は, $\mathrm{IW1}(i) \leftarrow 0$ が設定される.

(f) 固有ベクトルは正規直交系である.

(g) 固有ベクトルを必要としないときは, 4.10.6 $\begin{Bmatrix} \mathrm{DCSTEN} \\ \mathrm{RCSTEN} \end{Bmatrix}$ を使用する.

(7) **使用例**

(a) 問 題

$$A = \begin{bmatrix} 5 & 3 & & & & & & & & \\ 3 & 2 & 3 & & & & & & & \\ & 3 & 2 & 3 & & & & 0 & & \\ & & 3 & 2 & 3 & & & & & \\ & & & 3 & 2 & 3 & & & & \\ & & & & 3 & 2 & 3 & & & \\ & & & & & 3 & 2 & 3 & & \\ & & 0 & & & & 3 & 2 & 3 & \\ & & & & & & & 3 & 2 & 3 \\ & & & & & & & & 3 & 5 \end{bmatrix}$$

のとき, エルミート行列 A の固有値を区間 [0, 7.9] 内で 7.9 に近いものから 4 個を降順で求め, それに対応する固有ベクトルを求める.

(b) 入力データ

行列 A の対角成分 D, N=10, 行列 A の副対角成分 SD, EPS=−1.0 M=4, E1=7.9, E2=0, LNV=10

(c) 主プログラム

```
      PROGRAM BCSTEE
! *** EXAMPLE OF DCSTEE ***
      IMPLICIT REAL(8) (A-H,O-Z)
      PARAMETER ( LNV = 10 )
      DIMENSION D(LNV), SD(LNV), E(LNV), VE(LNV,LNV),&
                IW1(LNV), W1(6*LNV)
!
      READ(5,*) N, M, E1, E2
      READ(5,*) (D(I), I=1, N)
      READ(5,*) (SD(I), I=1, N-1)
!
      WRITE(6,1000) N, M, E1, E2
      WRITE(6,1100) 'DIAGONAL   ', (D(I), I=1, N)
      WRITE(6,1100) 'SUBDIAGONAL', (SD(I), I=1, N-1)
!
      EPS = -1.0D0
!
      CALL DCSTEE(D,N,SD,EPS,E,M,E1,E2,VE,LNV,IW1,W1,IERR)
!
      WRITE(6,1200) IERR
!
      DO 20 K=1, M-3, 4
         WRITE(6,1300) ('EIGENVALUE ', I=1, 4)
         WRITE(6,1400) (E(I), I=K, K+3)
         WRITE(6,1300) ('EIGENVECTOR', I=1, 4)
         DO 10 J=1, N
            WRITE(6,1500) (VE(J,I), I=K, K+3)
   10    CONTINUE
   20 CONTINUE
      IF(MOD(M,4).NE.0) THEN
         WRITE(6,1300) ('EIGENVALUE ', I=M/4*4+1, M)
         WRITE(6,1400) (E(I), I=M/4*4+1, M)
         WRITE(6,1300) ('EIGENVECTOR', I=M/4*4+1, M)
         DO 30 J=1, N
            WRITE(6,1500) (VE(J,I), I=M/4*4+1, M)
   30    CONTINUE
      ENDIF
      STOP
!
 1000 FORMAT(1X,/,/,&
             1X,'***  DCSTEE  ***',/,/,&
             1X,' **  INPUT  **',/,/,&
             1X,'N = ', I4,'  M = ', I4,/,/,&
             1X,'E1= ', 1PD14.7,'  E2= ', 1PD14.7,/,/,&
             1X,'     INPUT MATRIX A')
 1100 FORMAT(1X,/,6X, A11,/,&
             1X, 4X, 11(F7.1),/)
 1200 FORMAT(1X,/,/,&
             1X,' **  OUTPUT  **',/,/,&
             1X,'     IERR = ', I4)
 1300 FORMAT(1X,/,1X, 4(5X, A11, 2X))
 1400 FORMAT(1X, 2X, 4(2X, 1PD14.7, 2X))
 1500 FORMAT(1X, 1X, 4(F14.8, 4X))
      END
```

(d) 出力結果

```
***  DCSTEE  ***

 **  INPUT  **

N =   10  M =    4

E1=  7.9000000D+00  E2=  0.0000000D+00

     INPUT MATRIX A

     DIAGONAL
        5.0    2.0    2.0    2.0    2.0    2.0    2.0    2.0    2.0    5.0

     SUBDIAGONAL
        3.0    3.0    3.0    3.0    3.0    3.0    3.0    3.0    3.0


 **  OUTPUT  **

     IERR =    0

     EIGENVALUE        EIGENVALUE        EIGENVALUE        EIGENVALUE
     7.7063391D+00     6.8541020D+00     5.5267115D+00     3.8541020D+00

     EIGENVECTOR       EIGENVECTOR       EIGENVECTOR       EIGENVECTOR
    -0.44170765        0.42532540        0.39847023       -0.36180340
    -0.39847023        0.26286556        0.06995962        0.13819660
    -0.31622777        0.00000000       -0.31622777        0.44721360
    -0.20303072       -0.26286556       -0.44170765        0.13819660
    -0.06995962       -0.42532540       -0.20303072       -0.36180340
     0.06995962       -0.42532540        0.20303072       -0.36180340
     0.20303072       -0.26286556        0.44170765        0.13819660
     0.31622777        0.00000000        0.31622777        0.44721360
     0.39847023        0.26286556       -0.06995962        0.13819660
     0.44170765        0.42532540       -0.39847023       -0.36180340
```

4.10.6 DCSTEN, RCSTEN
実対称 3 重対角行列の固有値 (区間指定)

(1) **機 能**

実対称 3 重対角行列 A(ベクトル型) の指定した区間の固有値をバイセクション法により, 昇順で m 個, もしくは降順で m 個求める.

(2) **使用法**

倍精度サブルーチン:

```
  CALL DCSTEN (D, N, SD, EPS, E, M, E1, E2, W1, IERR)
```

単精度サブルーチン:

```
  CALL RCSTEN (D, N, SD, EPS, E, M, E1, E2, W1, IERR)
```

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

$$\text{I:}\ \begin{Bmatrix}\text{32 ビット整数版では INTEGER(4)}\\ \text{64 ビット整数版では INTEGER(8)}\end{Bmatrix}$$

項番	引数名	型	大きさ	入出力	内　　容
1	D	$\begin{Bmatrix}D\\R\end{Bmatrix}$	N	入 力	実対称 3 重対角行列 A の対角成分
2	N	I	1	入 力	行列 A の次数
3	SD	$\begin{Bmatrix}D\\R\end{Bmatrix}$	N	入 力	実対称 3 重対角行列 A の副対角成分
4	EPS	$\begin{Bmatrix}D\\R\end{Bmatrix}$	1	入 力	固有値の収束判定のための絶対誤差の上限を与えるパラメータ (注意事項 (b) 参照)
5	E	$\begin{Bmatrix}D\\R\end{Bmatrix}$	M	出 力	固有値
6	M	I	1	入 力	求めたい固有値の個数 m
				出 力	求めた固有値の個数
7	E1	$\begin{Bmatrix}D\\R\end{Bmatrix}$	1	入 力	E1 $<$ E2: E1 から昇順で固有値を求める. (E2 は, 上限) E1 $>$ E2: E1 から降順で固有値を求める. (E2 は, 下限) (注意事項 (c)(d) 参照)
8	E2	$\begin{Bmatrix}D\\R\end{Bmatrix}$	1	入 力	
9	W1	$\begin{Bmatrix}D\\R\end{Bmatrix}$	3 × N	ワーク	作業領域
10	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{M} \leq \mathrm{N}$

(5) **エラーインディケータ**

IERR の値	意　　　味	処　理　内　容
0	正常終了.	
1000	$N = 1$ であった.	$E(1) \leftarrow D(1)$ とする.
1500	区間 [E1, E2] に存在する固有値は M 個未満であった.	区間 [E1, E2] 内にある全ての固有値, 固有ベクトルを求める. 引数 M には求めた固有値の個数を出力する.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.

(6) **注意事項**

(a) 1 次元配列 D, SD に, 実対称 3 重対角行列のそれぞれ対角成分, 副対角成分を格納する. SD (N) は任意でよい (付録 B を参照).

(b) $\mathrm{EPS} \leq 0$ のときは内部で自動的に最適値を設定する. 通常は自動設定されるように負の値を設定されたい. なお, EPS はバイセクション法で固有値を求めるときに使用される.

(c) 固有値は E1<E2 のときには小さい順, E1>E2 のときには大きい順に格納される.

(d) E1=E2 のとき, 区間 [E1-EPS, E1+EPS] にある固有値が求まる. しかし, 通常は E1≠E2 となるように設定されたい.

4.11 実対称不規則スパース行列

4.11.1 DCSRSS, RCSRSS

実対称スパース行列 (実対称１次元行方向リスト型) (上三角型) の固有値・固有ベクトル

(1) 機 能

実対称スパース行列 A(実対称１次元行方向リスト型)(上三角型) の固有値を Jacobi-Davidson 法により, 大きい方から m 個, または小さい方から m 個求め, それに対応する固有ベクトルを求める.

(2) 使用法

倍精度サブルーチン:

CALL DCSRSS (A, NA, JA, IA, N, X, LDA, E, M, TR, IX, IS, ITM, IPREC, NDIA, ITJD,
ITQMR, IW, WK, IERR)

単精度サブルーチン:

CALL RCSRSS (A, NA, JA, IA, N, X, LDA, E, M, TR, IX, IS, ITM, IPREC, NDIA, ITJD,
ITQMR, IW, WK, IERR)

(3) 引 数

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\left\{\begin{array}{l}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{array}\right\}$

項番	引数名	型	大きさ	入出力	内　　　容
1	A	$\left\{\begin{array}{c}\mathrm{D}\\\mathrm{R}\end{array}\right\}$	NA	入 力	実対称行列 A(実対称１次元行方向リスト型) (上三角型) (注意事項 (a) 参照)
2	NA	I	1	入 力	配列 A の寸法: 行列 A の対角成分およびゼロでない上三角成分の個数
3	JA	I	NA	入 力	JA(i): 配列 A の第 i 要素に対応する行列 A の成分の列番号
4	IA	I	N + 1	入 力	IA(i): 行列 A の第 i 行の対角成分の, 配列 A における要素番号 (IA(N + 1) = IA(N) + 1)
5	N	I	1	入 力	行列 A の次数
6	X	$\left\{\begin{array}{c}\mathrm{D}\\\mathrm{R}\end{array}\right\}$	LDA, M	入 力	IX = 1 の時 : 反復ベクトル初期値
				出 力	各固有値に対応する固有ベクトル (列ベクトル)
7	LDA	I	1	入 力	配列 X の整合寸法
8	E	$\left\{\begin{array}{c}\mathrm{D}\\\mathrm{R}\end{array}\right\}$	M	出 力	固有値
9	M	I	1	入 力	求めたい固有値の個数 m (注意事項 (b) 参照)
10	TR	$\left\{\begin{array}{c}\mathrm{D}\\\mathrm{R}\end{array}\right\}$	M	入 力	固有値計算の収束判定のための相対残差ノルムの上限を与えるパラメータ (注意事項 (c) 参照)
				出 力	TR(i)　($i = 1, \cdots, \mathrm{M}$) 最終の相対残差ノルム

項番	引数名	型	大きさ	入出力	内　　　容
11	IX	I	1	入出力	反復ベクトル初期値の指定スイッチ (注意事項 (d) 参照) IX = −1: 反復ベクトル初期値をユーザは指定しない; 行列の対角成分を利用して固有値と固有ベクトルの初期値を生成する. IX = 0: 反復ベクトル初期値をユーザは指定しない; 乱数によってベクトルが初期発生される. IX = 1: 反復ベクトル初期値をユーザが指定する. それ以外の場合: 既定値 **0** が使われる.
12	IS	I	1	入 力	処理スイッチ (注意事項 (b) 参照) IS $\geq$ 0: 大きい方から順に M 個の固有値を求める. IS < 0 : 小さい方から順に M 個の固有値を求める.
13	ITM	I	1	入出力	近似部分空間の次元数 (注意事項 (e) 参照)
14	IPREC	I	1	入出力	前処理法選択スイッチ IPREC = 0: 対角スケーリング前処理 IPREC = 1: 反復 QMR 前処理, NDIA の先行する対角スケーリング前処理ステップつき それ以外の場合: IPREC を既定値 1 に変更して処理を続ける.
15	NDIA	I	1	入出力	先行する対角スケーリング前処理ステップの実行回数 (注意事項 (f) 参照)
16	ITJD	I	1	入出力	外側 JD 反復の上限回数 (既定値:1000) (注意事項 (g) 参照)
17	ITQMR	I	1	入出力	QMR 反復の上限回数 (既定値:1000) (注意事項 (h) 参照)
18	IW	I	2×M	ワーク	作業領域
19	WK	$\begin{Bmatrix} \mathrm{D} \\ \mathrm{R} \end{Bmatrix}$	内容参照	ワーク	作業領域 **大きさ**: $\mathrm{N} \times (2 \times \mathrm{ITM} + 3 \times \mathrm{M} + 9) + \mathrm{ITM} \times (3 \times \mathrm{ITM} + 2) + 4 \times \mathrm{M}$
20	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $\mathrm{N} > 0$

(b) $\mathrm{N} \leq \mathrm{NA}$

(c) $\mathrm{IA(N+1)} - 1 \leq \mathrm{NA}$

(d) $\mathrm{N} \leq \mathrm{LDA}$

(e) $0 < \mathrm{M} \leq \mathrm{N}$

(f) IX = 1 の時 : (ユーザが指定した M 個の反復ベクトル初期値すべて) $\neq 0$

(g) M < N の時 : M < ITM

(5) エラーインディケータ

IERR の値	意味	処理内容
0	正常終了.	
1000	N = 1 であった.	$E(1) \leftarrow A(1)$, $X(1,1) \leftarrow 1.0$ とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	
3030	制限条件 (d) を満足しなかった.	
3040	制限条件 (e) を満足しなかった.	
3070	制限条件 (f) を満足しなかった.	
3100	制限条件 (g) を満足しなかった.	
5000	部分固有ベクトルを求める処理中にエラーが発生した.	
6000	ITJD 回の反復を終了した時点で, 要求された精度まで収束しなかった. つまり, $\|Ax_i - \lambda_i x_i\|_2 / \|Ax_0 - \lambda_0 x_0\|_2$ がユーザが指定した収束判定基準値より大きかった.	

(6) 注意事項

(a) 1 次元配列 A に, 実対称行列の対角成分と上三角部分の非ゼロ成分を行方向に詰めて格納する (実対称 1 次元行方向リスト型格納 (上三角型); 付録 B 参照).

(b) 固有値は, IS≥0 のときには最大のものから大きい順に M 個, IS<0 のときには最小のものから小さい順に M 個が配列 E に格納される.

(c) 収束判定条件は, TR (1) の入力値に依存して以下のように決められる.

TR (1) > 0 の場合: 入力値を収束判定基準値として使う. すなわち, 収束判定は次の条件で行う.

$$\|Ax_i - \lambda_i x_i\|_2 / \|Ax_0 - \lambda_0 x_0\|_2 \leq \mathrm{TR}(1)$$

TR (1) ≤ 0 の場合: 既定値 10^{-8} (単精度サブルーチンの場合は 10^{-5}) を収束判定基準値とする. すなわち, 収束判定は次の条件で行う.

$$\|Ax_i - \lambda_i x_i\|_2 / \|Ax_0 - \lambda_0 x_0\|_2 \leq 10^{-8} \quad (10^{-5})$$

(d) IX = 1 のとき, 反復ベクトル初期値はユーザが指定する. 出発ベクトルとして好ましいのは, 求めるべき固有ベクトルを近似したものになっていることである. ユーザが指定したベクトルはこのサブルーチンの中で正規直交化される. もし, この処理がうまくいかなかった場合, 反復ベクトル初期値としてかわりに乱数発生された出発ベクトルが用いられる.

(e) 部分空間のサイズ ITM は JD アルゴリズムの収束性にとって非常に重要である. M < N のときは ITM > M でなくてはならない. ITM の最大値は全空間のサイズ N である. 大きい方あるいは小さい方から何個かの固有値と, 対応する固有ベクトルを求めるには, 部分空間の次元数 ITM を ITM ≥ 2 × M となるようにすることが推奨される.

部分空間のサイズを大きく選ぶほど JD アルゴリズムの収束性はよくなる. しかし, 部分空間が大きくなるほど多くのメモリを必要とする. 大規模スパース行列に対しては通常の場合, 2×M 以上 4×M 以下の部

分空間の次元数があれば十分である．

ITM の入力値が N 以上であったときは, ITM = N と変更して処理を続ける．

(f) 引数 NDIA の値は IPREC = 1 のときのみ参照される．IPREC = 1 かつ, NDIA の入力値が NDIA < 0 となったときは, NDIA = 10 と変更して処理を続ける．IPREC = 0 の場合, NDIA は 0 に変更され参照されない．

(g) 引数 ITJD の入力値が ITJD ≤ 0 となったときは, ITJD = 1000 と変更して処理を続ける．

(h) 引数 ITQMR の入力値が ITQMR ≤ 0 となったときは, ITQMR = 1000 と変更して処理を続ける．

(i) 固有ベクトルは正規直交系である．

(j) JD 反復が終了するのは, その時点までに求められた M 個の固有値および対応する固有ベクトルの残差ノルムを初期残差ノルムで割ったもの (= 相対残差ノルム) がすべて, ユーザが指定した収束判定基準値 TR(1) 以下になったときである．収束判定基準値はユーザの要求次第である．大抵の場合, 既定値 10^{-8} (単精度サブルーチンの場合は 10^{-5}) で十分な精度が得られる．

(7) **使用例**

(a) 問 題

次の行列 A の固有値を小さい順に 3 個とそれに対応する固有ベクトルを求める．

$$A = \begin{bmatrix} 5 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 6 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 3 & 6 & 3 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 6 & 3 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 3 & 6 & 3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 3 & 6 & 3 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 3 & 6 & 3 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 3 & 6 & 3 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 3 & 6 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 5 \end{bmatrix}$$

(b) 入力データ

行列 A を定義するための配列 A, JA, IA.

NA=27, N=10, LDA=11, M=3, TR(1)=10^{-10}, IX=0, IS=-1, ITM=5, IPREC=1, NDIA=1, ITJD=1000, ITQMR=1000.

(c) 主プログラム

```
      PROGRAM BCSRSS
! *** EXAMPLE OF DCSRSS ***
      IMPLICIT REAL(8) (A-H,O-Z)
      INTEGER LDA, ITMMAX
      PARAMETER ( LDA = 11, ITMMAX = 11 )
      DIMENSION A(LDA*LDA)
      DIMENSION JA(LDA*LDA),IA(LDA+1)
      DIMENSION X(LDA,LDA),E(LDA),TR(LDA)
      DIMENSION IW(2*ITMMAX)
      DIMENSION WK(LDA*(5*ITMMAX+9)+ITMMAX*(3*ITMMAX+6))
      INTEGER ITM, IPREC, NDIA, ITJD, ITQMR
!
      CHARACTER*80 FMT
!
      READ(5,*)  N ,NA ,M
      WRITE(6,1000) N ,NA ,M
      READ(5,*) (A(I),I=1,NA)
      READ(5,*) (JA(I),I=1,NA)
      READ(5,*) (IA(I),I=1,N+1)
      DO 40 I=1,N
        WRITE(FMT,1100) I*5, IA(I+1)-IA(I)
        WRITE(6,FMT)  ( A(J), J=IA(I), IA(I+1)-1 )
   40 CONTINUE
      WRITE(6,1150)
      WRITE(6,1200) (J,JA(J),J=1,NA)
      WRITE(6,1250)
```

```
      WRITE(6,1300) (J,IA(J),J=1,N+1)
!
      ITM = 5
      IX=0
      IS=-1
      IPREC = 1
      NDIA = 1
      ITJD = 1000
      ITQMR = 1000
!
      EPS = 1.0D-10
      TR(1) = EPS
!
      CALL DCSRSS(A, NA, JA, IA, N, X, LDA, E, M, TR, IX, IS,&
                  ITM, IPREC, NDIA, ITJD, ITQMR, IW, WK, IERR)
!
      WRITE(6,1400)  IERR
!
      DO 140 K=1, M-3, 4
         WRITE(6,2300) ('EIGENVALUE ', J=1,4)
         WRITE(6,2400) (E(J), J=K, K+3)
         WRITE(6,*)
         WRITE(6,2300) ('EIGENVECTOR', J=1,4)
         DO 130 I=1,N
            WRITE(6,2500) (X(I,J), J=K, K+3)
  130    CONTINUE
         WRITE(6,*)
         WRITE(6,2300) ('RESIDUAL', J=1,4)
         WRITE(6,2400) (TR(J), J=K, K+3)
  140 CONTINUE
!
      IF(MOD(M,4).NE.0) THEN
         MREM=M/4*4+1
         WRITE(6,2300) ('EIGENVALUE ', J=MREM, M)
         WRITE(6,2400) (E(J), J=M/4*4+1, M)
         WRITE(6,*)
         WRITE(6,2300) ('EIGENVECTOR', J=MREM, M)
         DO 150 I=1,N
            WRITE(6,2500) (X(I,J), J=MREM, M)
  150    CONTINUE
         WRITE(6,*)
         WRITE(6,2300) ('RESIDUAL', I=MREM, M)
         WRITE(6,2400) (TR(J), J=MREM, M)
      ENDIF
      STOP
!
 1000 FORMAT(/,1X,' ***  DCSRSS  ***',/,/,&
      1X,'   **  INPUT PARAMETER **',/,&
      1X,'     N = ',I5,/,&
      1X,'     NA = ',I4,/,&
      1X,'     M = ',I5,/,/,&
      1X,'   ** INPUT MATRIX A **')
 1100 FORMAT('(1X,',I3,'X,',I2,'(1X,F4.1))')
 1150 FORMAT(/,1X,'   ** INPUT INDEX JA **')
 1200 FORMAT((2X,5(' JA(',I2,') = ',I3,  2X,:)))
 1250 FORMAT(/,1X,'   ** INPUT INDEX IA **')
 1300 FORMAT((2X,5(' IA(',I2,') = ',I3,  2X,:)))
 1400 FORMAT(/,/,1X,'   ** OUTPUT **',/,/,5X,'IERR = ',I8,/)
 2300 FORMAT(/,1X,4(5X, A11, 2X),/)
 2400 FORMAT(1X,'  ',4(2X, 1PD14.7, 2X))
 2500 FORMAT(1X,'  ',4(F14.8, 4X))
      END
```

(d) 出力結果

```
 ***  DCSRSS  ***

   **  INPUT PARAMETER **
     N =    10
     NA =   27
     M =     3

   ** INPUT MATRIX A **
       5.0  2.0  1.0
            6.0  3.0  1.0
                 6.0  3.0  1.0
                      6.0  3.0  1.0
                           6.0  3.0  1.0
                                6.0  3.0  1.0
                                     6.0  3.0  1.0
                                          6.0  3.0  1.0
                                               6.0  2.0
                                                    5.0

   ** INPUT INDEX JA **
  JA( 1) =   1   JA( 2) =   2   JA( 3) =   3   JA( 4) =   2   JA( 5) =   3
  JA( 6) =   4   JA( 7) =   3   JA( 8) =   4   JA( 9) =   5   JA(10) =   4
  JA(11) =   5   JA(12) =   6   JA(13) =   5   JA(14) =   6   JA(15) =   7
  JA(16) =   6   JA(17) =   7   JA(18) =   8   JA(19) =   7   JA(20) =   8
  JA(21) =   9   JA(22) =   8   JA(23) =   9   JA(24) =  10   JA(25) =   9
  JA(26) =  10   JA(27) =  10

   ** INPUT INDEX IA **
  IA( 1) =   1   IA( 2) =   4   IA( 3) =   7   IA( 4) =  10   IA( 5) =  13
  IA( 6) =  16   IA( 7) =  19   IA( 8) =  22   IA( 9) =  25   IA(10) =  27
  IA(11) =  28


   ** OUTPUT **
```

```
IERR =        0

 EIGENVALUE        EIGENVALUE        EIGENVALUE
 1.8799058D+00     1.8926451D+00     2.2578112D+00

 EIGENVECTOR       EIGENVECTOR       EIGENVECTOR
  0.01172823        0.05600768       -0.12429215
  0.20382326       -0.01048668        0.41411041
 -0.44423970       -0.15306238       -0.48738830
  0.46949161        0.39024431        0.16106987
 -0.20136345       -0.56659903        0.22265032
 -0.20136345        0.56659903       -0.22265032
  0.46949161       -0.39024431       -0.16106987
 -0.44423970        0.15306238        0.48738830
  0.20382326        0.01048668       -0.41411041
  0.01172823       -0.05600768        0.12429215

    RESIDUAL          RESIDUAL          RESIDUAL
 9.7859321D-12     3.9447236D-16     5.4323145D-13
```

4.11.2 DCSJSS, RCSJSS
実対称スパース行列 (JAD 格納型) の固有値・固有ベクトル

(1) **機 能**

実対称スパース行列 A(JAD 格納型) の固有値を Jacobi-Davidson 法により, 大きい方から m 個, または小さい方から m 個求め, それに対応する固有ベクトルを求める.

(2) **使用法**

倍精度サブルーチン:

CALL DCSJSS (MJAD, AJAD, NA, IAJAD, JAJAD, JADORD, N, X, LDA, E, M, TR,
IX, IS, ITM, IPREC, NDIA, ITJD, ITQMR, IW, WK, IERR)

単精度サブルーチン:

CALL RCSJSS (MJAD, AJAD, NA, IAJAD, JAJAD, JADORD, N, X, LDA, E, M, TR,
IX, IS, ITM, IPREC, NDIA, ITJD, ITQMR, IW, WK, IERR)

(3) **引 数**

D:倍精度実数型 Z:倍精度複素数型
R:単精度実数型 C:単精度複素数型
I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\ \text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$

項番	引数名	型	大きさ	入出力	内　　容
1	MJAD	I	1	入 力	行列 A の JAD 格納型における, jagged diagonal の本数 (注意事項 (a) 参照)
2	AJAD	$\left\{\begin{matrix}D\\R\end{matrix}\right\}$	NA	入 力	行列 A(JAD 格納型) (注意事項 (a) 参照)
3	NA	I	1	入 力	配列 AJAD の大きさ. (行列 A のゼロでない要素として格納するものの個数)
4	IAJAD	I	MJAD+1	入 力	IAJAD(i): 行列 A の, 配列 AJAD および配列 JAJAD における, 第 i 番目の jagged diagonal の先頭インデックス (注意事項 (a) 参照)
5	JAJAD	I	NA	入 力	JAJAD(i): 行列 A の, 配列 AJAD における, 第 i 要素の列番号 (注意事項 (a) 参照)
6	JADORD	I	N	入 力	実対称 1 次元行方向リスト型から JAD 格納型へ変換を行う際に, 左側ベクトル (行列ベクトル積 $y = Ax$ の ベクトル y) に施される写像
7	N	I	1	入 力	行列 A の次数
8	X	$\left\{\begin{matrix}D\\R\end{matrix}\right\}$	LDA, M	入 力	IX = 1 の時 : 反復ベクトル初期値
				出 力	各列ベクトル: 各固有値に対応する固有ベクトル
9	LDA	I	1	入 力	配列 X の整合寸法
10	E	$\left\{\begin{matrix}D\\R\end{matrix}\right\}$	M	出 力	固有値

項番	引数名	型	大きさ	入出力	内　　容
11	M	I	1	入 力	求めたい固有値の個数 m (注意事項 (b) 参照)
12	TR	$\left\{\begin{matrix}\mathrm{D}\\ \mathrm{R}\end{matrix}\right\}$	M	入 力	(現残差ノルム)/(初期残差ノルム) の収束判定閾値 (注意事項 (c) 参照)
				出 力	TR(i)　($i = 1, \cdots, \mathrm{M}$) : (最終残差ノルム)/(初期残差ノルム)
13	IX	I	1	入出力	反復ベクトル初期値の指定スイッチ (注意事項 (d) 参照) IX $= -1$: 反復ベクトル初期値をユーザは指定しない; 行列の対角成分を利用して固有値と固有ベクトルの初期値を生成する. IX $= 0$: 反復ベクトル初期値をユーザは指定しない; 乱数によってベクトルが初期発生される. IX $= 1$: 反復ベクトル初期値をユーザが指定する. それ以外の場合:既定値 **0** が使われる.
14	IS	I	1	入 力	処理スイッチ (注意事項 (b) 参照) IS ≥ 0: 大きい方から順に m 個の固有値を求める. IS < 0 : 小さい方から順に m 個の固有値を求める.
15	ITM	I	1	入出力	近似部分空間の次元数 (注意事項 (e) 参照)
16	IPREC	I	1	入出力	前処理法選択スイッチ IPREC $= 0$: 対角スケーリング前処理 IPREC $= 1$: 反復 QMR 前処理, NDIA の先行する対角スケーリング前処理ステップつき それ以外の場合: IPREC を既定値 1 に変更して処理を続ける.
17	NDIA	I	1	入出力	先行する対角スケーリング前処理ステップの実行回数 (注意事項 (f) 参照)
18	ITJD	I	1	入出力	外側 JD 反復の上限回数 (既定値:1000) (注意事項 (g) 参照)
19	ITQMR	I	1	入出力	QMR 反復の上限回数 (既定値:1000) (注意事項 (h) 参照)
20	IW	I	2×M	ワーク	作業領域
21	WK	$\left\{\begin{matrix}\mathrm{D}\\ \mathrm{R}\end{matrix}\right\}$	内容参照	ワーク	作業領域 **大きさ**: $\mathrm{N} \times (2 \times \mathrm{ITM} + 3 \times \mathrm{M} + 9) + \mathrm{ITM} \times (3 \times \mathrm{ITM} + 2) + 4 \times \mathrm{M}$
22	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a) N > 0

(b) MJAD ≤ N

(c) MJAD > 0

(d) N ≤ NA

(e) IA(N + 1) − 1 ≤ NA

(f) N ≤ LDA

(g) 0 < M ≤ N

(h) IX = 1 の時：(ユーザが指定した M 個の反復ベクトル初期値すべて) ≠ 0

(i) M < N の時：M < ITM

(5) エラーインディケータ

IERR の値	意　　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	E(1) ← A(1), X(1, 1) ← 1.0 とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3005	制限条件 (b) を満足しなかった.	
3007	制限条件 (c) を満足しなかった.	
3010	制限条件 (d) を満足しなかった.	
3020	制限条件 (e) を満足しなかった.	
3030	制限条件 (f) を満足しなかった.	
3040	制限条件 (g) を満足しなかった.	
3070	制限条件 (h) を満足しなかった.	
3100	制限条件 (i) を満足しなかった.	
5000	部分固有ベクトルを求める処理中にエラーが発生した.	
6000	ITJD 回の反復を終了した時点で, 要求された精度まで収束しなかった. つまり, $\|Ax_i - \lambda_i x_i\|_2/\|Ax_0 - \lambda_0 x_0\|_2$ がユーザが指定した収束判定基準値より大きかった.	

(6) **注意事項**

(a) 行列 A のゼロでない要素を jagged diagonal の集まりとして配列し, 1 次元配列 AJAD に格納する (JAD 格納型; 付録 B 参照).

(b) 固有値は, IS≥0 のときには最大のものから大きい順に M 個, IS<0 のときには最小のものから小さい順に M 個が配列 E に格納される.

(c) 収束判定条件は, TR (1) の入力値に依存して以下のように決められる.

TR (1) > 0 の場合: 入力値を収束判定基準値として使う. すなわち, 収束判定は次の条件で行う.

$$\|Ax_i - \lambda_i x_i\|_2 / \|Ax_0 - \lambda_0 x_0\|_2 \leq \mathrm{TR}(1)$$

TR (1) ≤ 0 の場合: 既定値 10^{-8} (単精度サブルーチンの場合は 10^{-5}) を収束判定基準値とする. すなわち, 収束判定は次の条件で行う.

$$\|Ax_i - \lambda_i x_i\|_2 / \|Ax_0 - \lambda_0 x_0\|_2 \leq 10^{-8} \quad (10^{-5})$$

(d) IX $=$ 1 のとき, 反復ベクトル初期値はユーザが指定する. 出発ベクトルとして好ましいのは, 求めるべき固有ベクトルを近似したものになっていることである. ユーザが指定したベクトルはこのサブルーチンの中で正規直交化される. もし, この処理がうまくいかなかった場合, 反復ベクトル初期値としてかわりに乱数発生された出発ベクトルが用いられる.

(e) 部分空間のサイズ ITM は JD アルゴリズムの収束性にとって非常に重要である. M $<$ N のときは ITM $>$ M でなくてはならない. ITM の最大値は全空間のサイズ N である. 大きい方あるいは小さい方から何個かの固有値と, 対応する固有ベクトルを求めるには, 部分空間の次元数 ITM を ITM≥ 2×M となるようにすることが推奨される.

部分空間のサイズを大きく選ぶほど JD アルゴリズムの収束性はよくなる. しかし, 部分空間が大きくなるほど多くのメモリを必要とする. 大規模スパース行列に対しては通常の場合, 2×M 以上 4×M 以下の部分空間の次元数があれば十分である.

ITM の入力値が N 以上であったときは, ITM $=$ N と変更して処理を続ける.

(f) 引数 NDIA の値は IPREC $=$ 1 のときのみ参照される. IPREC $=$ 1 かつ, NDIA の入力値が NDIA $<$ 0 となったときは, NDIA $=$ 10 と変更して処理を続ける. IPREC $=$ 0 の場合, NDIA は 0 に変更され参照されない.

(g) 引数 ITJD の入力値が ITJD $\leq$ 0 となったときは, ITJD $=$ 1000 と変更して処理を続ける.

(h) 引数 ITQMR の入力値が ITQMR $\leq$ 0 となったときは, ITQMR $=$ 1000 と変更して処理を続ける.

(i) 固有ベクトルは正規直交系である.

(j) JD 反復が終了するのは, その時点までに求められた M 個の固有値および対応する固有ベクトルの残差ノルムを初期残差ノルムで割ったもの (= 相対残差ノルム) がすべて, ユーザが指定した収束判定基準値 TR(1) 以下になったときである. 収束判定基準値はユーザの要求次第である. 大抵の場合, 既定値 10^{-8} (単精度サブルーチンの場合は 10^{-5}) で十分な精度が得られる.

(7) **使用例**

(a) 問 題

次の行列 A の固有値を小さい順に 3 個とそれに対応する固有ベクトルを求める.

$$A = \begin{bmatrix} 5 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 6 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 3 & 6 & 3 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 6 & 3 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 3 & 6 & 3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 3 & 6 & 3 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 3 & 6 & 3 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 3 & 6 & 3 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 3 & 6 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 5 \end{bmatrix}$$

ここでは, (実対称 1 次元行方向リスト型)(上三角型) で行列データを入力し, 行列の格納モード変換サブルーチンを用いて JAD 格納型に変換後, 本サブルーチンを用いる.

(b) 入力データ

行列 A を定義するための配列 A, JA, IA.

NA=27, N=10, LDA=11, M=3, TR(1)=10^{-10}, IX=0, IS=-1, ITM=5, IPREC=1, NDIA=1, ITJD=1000, ITQMR=1000.

(c) 主プログラム

```
      PROGRAM BCSJSS
! *** EXAMPLE OF DCSJSS ***
      IMPLICIT NONE
      INTEGER N, NA, NAJAD, M, I, J, K, IERR
      INTEGER LDA, ITMMAX, LXA, LXIA
      PARAMETER ( LDA = 11, ITMMAX = 11 )
      PARAMETER ( LXA = LDA*LDA, LXIA = LDA+1 )
      REAL(8) A(LDA*LDA)
      INTEGER JA(LDA*LDA), IA(LDA+1)
!
      REAL(8) AJAD(LXA)
      INTEGER IAJAD(LXIA), JAJAD(LXA), JADORD(LDA), MJAD
      INTEGER IWJ(LDA*3+1)
!
      REAL(8) X(LDA,LDA), E(LDA), TR(LDA), EPS
      INTEGER IW(2*ITMMAX)
      REAL(8) WK(LDA*(5*ITMMAX+11)+ITMMAX*(3*ITMMAX+6))
      INTEGER ITM, IPREC, NDIA, ITJD, ITQMR, IX, IS, MREM
!
      CHARACTER*80 FMT
!
      READ(5,*)  N, NA, M
      WRITE(6,1000) N ,NA ,M
      READ(5,*) (A(I),I=1,NA)
      READ(5,*) (JA(I),I=1,NA)
      READ(5,*) (IA(I),I=1,N+1)
      DO 40 I=1,N
        WRITE(FMT,1100) I*5, IA(I+1)-IA(I)
        WRITE(6,FMT)  ( A(J), J=IA(I), IA(I+1)-1 )
   40 CONTINUE
      WRITE(6,1150)
      WRITE(6,1200) (J,JA(J),J=1,NA)
      WRITE(6,1250)
      WRITE(6,1300) (J,IA(J),J=1,N+1)
!
      CALL DARSJD(N, A, IA, JA, LXA, LXIA,&
           MJAD, AJAD, IAJAD, JAJAD,&
           JADORD, IWJ, IERR)
!
      WRITE(6,1400) IERR
!
      ITM = 5
      IX = 0
      IS = -1
      IPREC = 1
      NDIA = 1
      ITJD = 1000
      ITQMR = 1000
!
      EPS = 1.0D-10
      TR(1) = EPS
```

```
      NAJAD = IAJAD(MJAD+1) - IAJAD(1)
!
      CALL DCSJSS(MJAD, AJAD, NAJAD, IAJAD, JAJAD, JADORD,&
                  N, X, LDA, E, M, TR, IX, IS,&
                  ITM, IPREC, NDIA, ITJD, ITQMR, IW, WK, IERR)
!
      WRITE(6,1500)  IERR
!
      DO 140 K=1, M-3, 4
         WRITE(6,2300) ('EIGENVALUE ', J=1,4)
         WRITE(6,2400) (E(J), J=K, K+3)
         WRITE(6,2300) ('EIGENVECTOR', J=1,4)
         DO 130 I=1,N
            WRITE(6,2500) (X(I,J), J=K, K+3)
  130    CONTINUE
         WRITE(6,2300) ('RESIDUAL   ', J=1,4)
         WRITE(6,2400) (TR(J), J=K, K+3)
  140 CONTINUE
!
      IF(MOD(M,4).NE.0) THEN
         MREM=M/4*4+1
         WRITE(6,2300) ('EIGENVALUE ', J=MREM, M)
         WRITE(6,2400) (E(J), J=M/4*4+1, M)
         WRITE(6,2300) ('EIGENVECTOR', J=MREM, M)
         DO 150 I=1,N
            WRITE(6,2500) (X(I,J), J=MREM, M)
  150    CONTINUE
         WRITE(6,2300) ('RESIDUAL   ', I=MREM, M)
         WRITE(6,2400) (TR(J), J=MREM, M)
      ENDIF
      STOP
!
 1000 FORMAT(/,1X,' ***  DCSJSS  ***',/,/,&
      1X,'   ** INPUT PARAMETER **',/,&
      1X,'     N = ',I5,/,&
      1X,'     NA = ',I4,/,&
      1X,'     M = ',I5,/,/,&
      1X,'   ** INPUT MATRIX A **')
 1100 FORMAT('(1X,',I3,'X,',I2,'(1X,F4.1))')
 1150 FORMAT(/,1X,'   ** INPUT INDEX JA **')
 1200 FORMAT((2X,5(' JA(',I2,') = ',I3,  2X,:)))
 1250 FORMAT(/,1X,'   ** INPUT INDEX IA **')
 1300 FORMAT((2X,5(' IA(',I2,') = ',I3,  2X,:)))
 1400 FORMAT(/,/,&
      1X,'    IERR AT DARSJD (STORAGE TRANSFORM CSR->JAD) = ',I6)
 1500 FORMAT(/,/,1X,'   ** OUTPUT **',/,/,5X,'IERR = ',I8)
 2300 FORMAT(/,1X,4(5X, A11, 2X),/)
 2400 FORMAT(1X,'  ',4(2X, 1PD14.7, 2X))
 2500 FORMAT(1X,'  ',4(F14.8, 4X))
      END
```

(d) 出力結果

```
 ***  DCSJSS  ***

   ** INPUT PARAMETER **
     N =    10
     NA =   27
     M =     3

   ** INPUT MATRIX A **
       5.0  2.0  1.0
            6.0  3.0  1.0
                 6.0  3.0  1.0
                      6.0  3.0  1.0
                           6.0  3.0  1.0
                                6.0  3.0  1.0
                                     6.0  3.0  1.0
                                          6.0  3.0  1.0
                                               6.0  2.0
                                                    5.0

   ** INPUT INDEX JA **
  JA( 1) =   1   JA( 2) =   2   JA( 3) =   3   JA( 4) =   2   JA( 5) =   3
  JA( 6) =   4   JA( 7) =   3   JA( 8) =   4   JA( 9) =   5   JA(10) =   4
  JA(11) =   5   JA(12) =   6   JA(13) =   5   JA(14) =   6   JA(15) =   7
  JA(16) =   6   JA(17) =   7   JA(18) =   8   JA(19) =   7   JA(20) =   8
  JA(21) =   9   JA(22) =   8   JA(23) =   9   JA(24) =  10   JA(25) =   9
  JA(26) =  10   JA(27) =  10

   ** INPUT INDEX IA **
  IA( 1) =   1   IA( 2) =   4   IA( 3) =   7   IA( 4) =  10   IA( 5) =  13
  IA( 6) =  16   IA( 7) =  19   IA( 8) =  22   IA( 9) =  25   IA(10) =  27
  IA(11) =  28

    IERR AT DARSJD (STORAGE TRANSFORM CSR->JAD) =      0


   ** OUTPUT **

    IERR =        0

     EIGENVALUE        EIGENVALUE        EIGENVALUE
     1.8799058D+00     1.8926451D+00     2.2578112D+00

     EIGENVECTOR       EIGENVECTOR       EIGENVECTOR
     -0.01172823        0.05600768       -0.12429215
     -0.20382326       -0.01048668        0.41411041
      0.44423970       -0.15306238       -0.48738830
```

```
-0.46949161         0.39024431         0.16106987
 0.20136345        -0.56659903         0.22265032
 0.20136345         0.56659903        -0.22265032
-0.46949161        -0.39024431        -0.16106987
 0.44423970         0.15306238         0.48738830
-0.20382326         0.01048668        -0.41411041
-0.01172823        -0.05600768         0.12429215

RESIDUAL           RESIDUAL           RESIDUAL
9.7858582D-12      6.8865735D-16      5.4312008D-13
```

4.12 エルミート不規則スパース行列

4.12.1 ZCHJSS, CCHJSS
エルミートスパース行列 (JAD 格納型) の固有値・固有ベクトル

(1) **機 能**

エルミートスパース行列 A(JAD 格納型) の固有値を Jacobi-Davidson 法により, 大きい方から m 個, または小さい方から m 個求め, それに対応する固有ベクトルを求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZCHJSS (MJAD, AJAD, NA, IAJAD, JAJAD, JADORD, N, X, LDA, E, M, TR,
IX, IS, ITM, IPREC, NDIA, ITJD, ITQMR, IW, WK, IERR)

単精度サブルーチン:

CALL CCHJSS (MJAD, AJAD, NA, IAJAD, JAJAD, JADORD, N, X, LDA, E, M, TR,
IX, IS, ITM, IPREC, NDIA, ITJD, ITQMR, IW, WK, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$

項番	引数名	型	大きさ	入出力	内　容
1	MJAD	I	1	入 力	行列 A の JAD 格納型における, jagged diagonal の本数 (注意事項 (a) 参照)
2	AJAD	$\left\{\begin{matrix}\text{Z}\\\text{C}\end{matrix}\right\}$	NA	入 力	行列 A(JAD 格納型) (注意事項 (a) 参照)
3	NA	I	1	入 力	配列 AJAD の大きさ (行列 A のゼロでない要素として格納するものの個数)
4	IAJAD	I	MJAD+1	入 力	IAJAD(i): 行列 A の, 配列 AJAD および 配列 JAJAD における, 第 i 番目の jagged diagonal の先頭インデックス (注意事項 (a) 参照)
5	JAJAD	I	NA	入 力	JAJAD(i): 行列 A の, 配列 AJAD における, 第 i 要素の列番号 (注意事項 (a) 参照)
6	JADORD	I	N	入 力	実対称 1 次元行方向リスト型から JAD 格納型へ変換を行う際に, 左側ベクトル (行列ベクトル積 $y = Ax$ の ベクトル y) に施される写像
7	N	I	1	入 力	行列 A の次数
8	X	$\left\{\begin{matrix}\text{Z}\\\text{C}\end{matrix}\right\}$	LDA, M	入 力	IX $=1$ の時 : 反復ベクトル初期値
				出 力	各列ベクトル: 各固有値に対応する固有ベクトル
9	LDA	I	1	入 力	配列 X の整合寸法

項番	引数名	型	大きさ	入出力	内　　　　容
10	E	$\begin{Bmatrix}\mathrm{Z}\\\mathrm{C}\end{Bmatrix}$	M	出 力	固有値
11	M	I	1	入 力	求めたい固有値の個数 m (注意事項 (b) 参照)
12	TR	$\begin{Bmatrix}\mathrm{Z}\\\mathrm{C}\end{Bmatrix}$	M	入 力	(現残差ノルム)/(初期残差ノルム) の収束判定閾値 (注意事項 (c) 参照)
				出 力	TR(i)　($i = 1, \cdots,$ M): (最終残差ノルム)/(初期残差ノルム)
13	IX	I	1	入出力	反復ベクトル初期値の指定スイッチ (注意事項 (d) 参照) IX $= -1$: 反復ベクトル初期値をユーザは指定しない; 行列の対角成分を利用して固有値と固有ベクトルの初期値を生成する. IX $= 0$: 反復ベクトル初期値をユーザは指定しない; 乱数によってベクトルが初期発生される. IX $= 1$: 反復ベクトル初期値をユーザが指定する. それ以外の場合:既定値 **0** が使われる.
14	IS	I	1	入 力	処理スイッチ (注意事項 (b) 参照) IS ≥ 0 : 大きい方から順に m 個の固有値を求める. IS < 0 : 小さい方から順に m 個の固有値を求める.
15	ITM	I	1	入出力	近似部分空間の次元数 (注意事項 (e) 参照)
16	IPREC	I	1	入出力	前処理法選択スイッチ IPREC $= 0$: 対角スケーリング前処理 IPREC $= 1$: 反復 QMR 前処理, NDIA の先行する対角スケーリング前処理ステップつき それ以外の場合: IPREC を既定値 1 に変更して処理を続ける.
17	NDIA	I	1	入出力	先行する対角スケーリング前処理ステップの実行回数 (注意事項 (f) 参照)
18	ITJD	I	1	入出力	外側 JD 反復の上限回数 (既定値:1000) (注意事項 (g) 参照)
19	ITQMR	I	1	入出力	QMR 反復の上限回数 (既定値:1000) (注意事項 (h) 参照)
20	IW	I	2×M	ワーク	作業領域
21	WK	$\begin{Bmatrix}\mathrm{Z}\\\mathrm{C}\end{Bmatrix}$	内容参照	ワーク	作業領域 **大きさ**: $\mathrm{N} \times (2 \times \mathrm{ITM} + 3 \times \mathrm{M} + 9) + \mathrm{ITM} \times (3 \times \mathrm{ITM} + 2) + 4 \times \mathrm{M}$
22	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $\mathrm{N} > 0$

(b) $\mathrm{MJAD} \leq \mathrm{N}$

(c) $\mathrm{MJAD} > 0$

(d) $\mathrm{N} \leq \mathrm{NA}$

(e) $\mathrm{IAJAD(MJAD} + 1) - 1 \leq \mathrm{NA}$

(f) $\mathrm{N} \leq \mathrm{LDA}$

(g) $0 < \mathrm{M} \leq \mathrm{N}$

(h) $\mathrm{IX} = 1$ の時：(ユーザが指定した M 個の反復ベクトル初期値すべて) $\neq 0$

(i) $\mathrm{M} < \mathrm{N}$ の時：$\mathrm{M} < \mathrm{ITM}$

(5) **エラーインディケータ**

IERR の値	意　　味	処　理　内　容
0	正常終了.	
1000	$\mathrm{N} = 1$ であった.	$\mathrm{E}(1) \leftarrow \mathrm{A}(1)$, $\mathrm{X}(1,1) \leftarrow 1.0$ とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3005	制限条件 (b) を満足しなかった.	
3007	制限条件 (c) を満足しなかった.	
3010	制限条件 (d) を満足しなかった.	
3020	制限条件 (e) を満足しなかった.	
3030	制限条件 (f) を満足しなかった.	
3040	制限条件 (g) を満足しなかった.	
3070	制限条件 (h) を満足しなかった.	
3100	制限条件 (i) を満足しなかった.	
5000	部分固有ベクトルを求める処理中にエラーが発生した.	
6000	ITJD 回の反復を終了した時点で, 要求された精度まで収束しなかった. つまり, $\|Ax_i - \lambda_i x_i\|_2 / \|Ax_0 - \lambda_0 x_0\|_2$ がユーザが指定した収束判定基準値より大きかった.	

(6) **注意事項**

(a) 行列 A のゼロでない要素を jagged diagonal の集まりとして配列し, 1 次元配列 AJAD に格納する (JAD 格納型; 付録 B 参照).

(b) 固有値は, $\mathrm{IS} \geq 0$ のときには最大のものから大きい順に M 個, $\mathrm{IS} < 0$ のときには最小のものから小さい順に M 個が配列 E に格納される.

(c) 収束判定条件は, TR(1) の入力値に依存して以下のように決められる.

TR(1) > 0 の場合: 入力値を収束判定基準値として使う. すなわち, 収束判定は次の条件で行う.

$$\|Ax_i - \lambda_i x_i\|_2 / \|Ax_0 - \lambda_0 x_0\|_2 \le \mathrm{TR}(1)$$

TR(1) ≤ 0 の場合: 既定値 10^{-8} (単精度サブルーチンの場合は 10^{-5}) を収束判定基準値とする. すなわち, 収束判定は次の条件で行う.

$$\|Ax_i - \lambda_i x_i\|_2 / \|Ax_0 - \lambda_0 x_0\|_2 \le 10^{-8} \quad (10^{-5})$$

(d) IX $= 1$ のとき, 反復ベクトル初期値はユーザが指定する. 出発ベクトルとして好ましいのは, 求めるべき固有ベクトルを近似したものになっていることである. ユーザが指定したベクトルはこのサブルーチンの中で正規直交化される. もし, この処理がうまくいかなかった場合, 反復ベクトル初期値としてかわりに乱数発生された出発ベクトルが用いられる.

(e) 部分空間のサイズITM はJD アルゴリズムの収束性にとって非常に重要である. M $<$ N のときはITM $>$ M でなくてはならない. ITM の最大値は全空間のサイズN である. 大きい方あるいは小さい方から何個かの固有値と, 対応する固有ベクトルを求めるには, 部分空間の次元数 ITM を ITM$\ge$ 2×M となるようにすることが推奨される.

部分空間のサイズを大きく選ぶほど JD アルゴリズムの収束性はよくなる. しかし, 部分空間が大きくなるほど多くのメモリを必要とする. 大規模スパース行列に対しては通常の場合, 2×M 以上 4×M 以下の部分空間の次元数があれば十分である.

ITM の入力値が N 以上であったときは, ITM $=$ N と変更して処理を続ける.

(f) 引数 NDIA の値は IPREC $= 1$ のときのみ参照される. IPREC $= 1$ かつ, NDIA の入力値が NDIA < 0 となったときは, NDIA $= 10$ と変更して処理を続ける. IPREC $= 0$ の場合, NDIA は 0 に変更され参照されない.

(g) 引数 ITJD の入力値が ITJD ≤ 0 となったときは, ITJD $= 1000$ と変更して処理を続ける.

(h) 引数 ITQMR の入力値が ITQMR ≤ 0 となったときは, ITQMR $= 1000$ と変更して処理を続ける.

(i) 固有ベクトルは正規直交系である.

(j) JD 反復が終了するのは, その時点までに求められた M 個の固有値および対応する固有ベクトルの残差ノルムを初期残差ノルムで割ったもの (= 相対残差ノルム) がすべて, ユーザが指定した収束判定基準値 TR(1) 以下になったときである. 収束判定基準値はユーザの要求次第である. 大抵の場合, 既定値 10^{-8} (単精度サブルーチンの場合は 10^{-5}) で十分な精度が得られる.

(7) **使用例**

(a) 問 題

次の行列 A の固有値を小さい順に 2 個とそれに対応する固有ベクトルを求める.

$$A = \begin{bmatrix} -2.28 & 1.78-2.03i & 2.26+0.10i & -0.12+2.53i \\ 1.78+2.03i & -1.12 & 0.01+0.43i & -1.07+0.86i \\ 2.26-0.10i & 0.01-0.43i & -0.37 & 2.31-0.92i \\ -0.12-2.53i & -1.07-0.86i & 2.31+0.92i & -0.73 \end{bmatrix}$$

ここでは, (エルミート 1 次元行方向リスト型)(上三角型) で行列データを入力し, 行列の格納モード変換サブルーチンを用いて JAD 格納型に変換後, 本サブルーチンを用いる.

(b) 入力データ

行列 A を定義するための配列 A, JA, IA.

NA=121, N=4, LDA=11, M=2, TR(1)=10^{-10}, IX=0, IS=-1, ITM=3, IPREC=1, NDIA=1, ITJD=1000, ITQMR=1000.

(c) 主プログラム

```
      PROGRAM ACHJSS
! *** EXAMPLE OF ZCHJSS ***
      IMPLICIT NONE
      INTEGER N, NA, NAJAD, M, I, J, K, IERR
      INTEGER LDA, ITMMAX, LXA, LXIA
      PARAMETER ( LDA = 11, ITMMAX = 11 )
      PARAMETER ( LXA = LDA*LDA, LXIA = LDA+1 )
      COMPLEX(8) A(LDA*LDA)
      INTEGER JA(LDA*LDA),IA(LDA+1)
!
      COMPLEX(8) AJAD(LXA)
      INTEGER IAJAD(LXIA), JAJAD(LXA), JADORD(LDA), MJAD
      INTEGER IWJ(LDA*3+1)
!
      COMPLEX(8) X(LDA,LDA)
      REAL(8) E(LDA),TR(LDA), EPS
      INTEGER IW(2*ITMMAX)
      COMPLEX(8) WK(LDA*(5*ITMMAX+11)+ITMMAX*(3*ITMMAX+6))
      INTEGER ITM, IPREC, NDIA, ITJD, ITQMR, IX, IS, MREM
!
      CHARACTER*80 FMT
!
      READ(5,*)  N, NA, M
      WRITE(6,1000) N ,NA ,M
      READ(5,*) (A(I),I=1,NA)
      READ(5,*) (JA(I),I=1,NA)
      READ(5,*) (IA(I),I=1,N+1)
      DO 40 I=1,N
        WRITE(FMT,1100) (I-1)*18+1, IA(I+1)-IA(I)
        WRITE(6,FMT)  ( A(J), J=IA(I), IA(I+1)-1 )
   40 CONTINUE
      WRITE(6,1150)
      WRITE(6,1200) (J,JA(J),J=1,NA)
      WRITE(6,1250)
      WRITE(6,1300) (J,IA(J),J=1,N+1)
!
      CALL ZARSJD(N, A, IA, JA, LXA, LXIA,&
           MJAD, AJAD, IAJAD, JAJAD,&
           JADORD, IWJ, IERR)
!
      WRITE(6,1400) IERR
!
      ITM = 5
      IX = 0
      IS = -1
      IPREC = 1
      NDIA = 1
      ITJD = 1000
      ITQMR = 1000
!
      EPS = 1.0D-10
      TR(1) = EPS

      NAJAD = IAJAD(MJAD+1) - IAJAD(1)
!
      CALL ZCHJSS(MJAD, AJAD, NAJAD, IAJAD, JAJAD, JADORD,&
                  N, X, LDA, E, M, TR, IX, IS,&
                  ITM, IPREC, NDIA, ITJD, ITQMR, IW, WK, IERR)
!
      WRITE(6,1500)  IERR
!
      DO 140 K=1, M-1, 2
         WRITE(6,2300) ('EIGENVALUE ', J=1,2)
         WRITE(6,2400) (E(J), J=K, K+1)
         WRITE(6,2300) ('EIGENVECTOR', J=1,2)
         DO 130 I=1,N
            WRITE(6,2500) (X(I,J), J=K, K+1)
  130    CONTINUE
         WRITE(6,2300) ('RESIDUAL   ', J=1,2)
         WRITE(6,2400) (TR(J), J=K, K+1)
  140 CONTINUE
!
      IF(MOD(M,2).NE.0) THEN
         MREM=M/2*2+1
         WRITE(6,2300) ('EIGENVALUE ', J=MREM, M)
         WRITE(6,2400) (E(J), J=M/2*2+1, M)
         WRITE(6,2300) ('EIGENVECTOR', J=MREM, M)
         DO 150 I=1,N
            WRITE(6,2500) (X(I,J), J=MREM, M)
  150    CONTINUE
         WRITE(6,2300) ('RESIDUAL   ', I=MREM, M)
         WRITE(6,2400) (TR(J), J=MREM, M)
      ENDIF
      STOP
!
 1000 FORMAT(/,1X,' ***  ZCHJSS  ***',/,/,&
      1X,'   ** INPUT PARAMETER **',/,&
      1X,'     N = ',I5,/,&
      1X,'     NA = ',I4,/,&
      1X,'     M = ',I5,/,/,&
      1X,'   ** INPUT MATRIX A **')
 1100 FORMAT('(1X,',I3,'X,',I2,&
       '(4X,"(",F5.2,",",1X,F5.2,")"))')
 1150 FORMAT(/,1X,'   ** INPUT INDEX JA **')
 1200 FORMAT((2X,5(' JA(',I2,') = ',I3,  2X,:)))
 1250 FORMAT(/,1X,'   ** INPUT INDEX IA **')
 1300 FORMAT((2X,5(' IA(',I2,') = ',I3,  2X,:)))
 1400 FORMAT(/,/,&
```

```
     1X,'    IERR AT ZARSJD (STORAGE TRANSFORM CSR->JAD) = ',I6)
1500 FORMAT(/,/,1X,'   ** OUTPUT **',/,/,5X,'IERR = ',I8)
2300 FORMAT(/,1X,2(5X, A11, 20X),/)
2400 FORMAT(1X,'  ',2(2X, 1PD14.7, 18X))
2500 FORMAT(1X,'  ',2('(', F14.8, ',', F14.8, ')', 3X))
     END
```

(d) 出力結果

```
***  ZCHJSS  ***

  ** INPUT PARAMETER **
    N =     4
    NA =   10
    M =     2

  ** INPUT MATRIX A **
    (-2.28,  0.00)    ( 1.78, -2.03)    ( 2.26,  0.10)    (-0.12,  2.53)
                      (-1.12,  0.00)    ( 0.01,  0.43)    (-1.07,  0.86)
                                        (-0.37,  0.00)    ( 2.31, -0.92)
                                                          (-0.73,  0.00)

  ** INPUT INDEX JA **
 JA( 1) =   1   JA( 2) =   2   JA( 3) =   3   JA( 4) =   4   JA( 5) =   2
 JA( 6) =   3   JA( 7) =   4   JA( 8) =   3   JA( 9) =   4   JA(10) =   4

  ** INPUT INDEX IA **
 IA( 1) =   1   IA( 2) =   5   IA( 3) =   8   IA( 4) =  10   IA( 5) =  11


   IERR AT ZARSJD (STORAGE TRANSFORM CSR->JAD) =      0


  ** OUTPUT **

   IERR =        0

    EIGENVALUE                          EIGENVALUE

   -6.0001855D+00                     -3.0030337D+00

    EIGENVECTOR                         EIGENVECTOR

 (   -0.69071106,   -0.23593286)   (   -0.22312724,    0.13258141)
 (    0.09074333,    0.24877543)   (    0.72843164,    0.05737392)
 (    0.34833130,    0.26867067)   (   -0.33188966,    0.02190003)
 (   -0.03041543,   -0.45020731)   (    0.50799388,    0.17333161)

    RESIDUAL                            RESIDUAL

    2.5845464D-15                      7.5446512D-16
```

4.13　実行列 (2 次元配列型) の一般化固有値問題

4.13.1　DCGGAA, RCGGAA
実行列 (一般化固有値問題) の全固有値・全固有ベクトル

(1) **機 能**

実行列 (2 次元配列型) の一般化固有値問題 $A\boldsymbol{x} = \lambda B\boldsymbol{x}$ (A, B:実行列) の全固有値とそれに対応する全固有ベクトルをハウスホルダー法, コンビネーションシフト QZ 法により求める.

(2) **使用法**

倍精度サブルーチン:

CALL DCGGAA (A, LNA, N, B, LNB, ALFR, ALFI, BETA, VE, LNV, IERR)

単精度サブルーチン:

CALL RCGGAA (A, LNA, N, B, LNB, ALFR, ALFI, BETA, VE, LNV, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\left\{\begin{array}{l}\text{32 ビット整数版では INTEGER(4)}\\ \text{64 ビット整数版では INTEGER(8)}\end{array}\right\}$

項番	引数名	型	大きさ	入出力	内　容
1	A	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	LNA, N	入 力	実行列 A(2 次元配列型)
				出 力	入力時の内容は保存されない
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A, B の次数
4	B	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	LNB, N	入 力	実行列 B (2 次元配列型)
				出 力	入力時の内容は保存されない
5	LNB	I	1	入 力	配列 B の整合寸法
6	ALFR	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	N	出 力	変換後の行列 A の対角成分の実部 (注意事項 (a) 参照)
7	ALFI	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	N	出 力	変換後の行列 A の対角成分の虚部 (注意事項 (a) 参照)
8	BETA	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	N	出 力	変換後の行列 B の対角成分 (注意事項 (a) 参照)
9	VE	$\begin{Bmatrix}\text{D}\\\text{R}\end{Bmatrix}$	LNV, N	出 力	固有ベクトル (注意事項 (b), (c) 参照)
10	LNV	I	1	入 力	配列 VE の整合寸法
11	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNA}, \mathrm{LNB}, \mathrm{LNV}$

(5) **エラーインディケータ**

IERR の値	意味	処理内容
0	正常終了.	
1000	$\mathrm{N} = 1$ であった.	$\mathrm{ALFR}(1) \leftarrow \mathrm{sign}(\mathrm{B}(1,1)) \times \mathrm{A}(1,1)$, $\mathrm{ALFI}(1) \leftarrow 0.0$, $\mathrm{BETA}(1) \leftarrow \lvert\mathrm{B}(1,1)\rvert$, $\mathrm{VE}(1,1) \leftarrow 1.0$ とする.
2000	配列 BETA の中に 0.0 がある.	処理を続ける (注意事項 (b) 参照).
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
$5000 + i$	i 番目の固有値を求める段階で収束しなかった. $(1 \le i \le \mathrm{N})$	ALFR, ALFI, BETA の $i+1$ 番目より N 番目の要素は求められている. 固有ベクトルは求められない.

(6) **注意事項**

(a) 固有値は添字の大きい方から求まり, ALFR, ALFI, BETA の配列に格納される.
ALFR, ALFI, BETA の第 j 番目の要素をそれぞれ $\alpha_j, \alpha'_j, \beta_j$ とすると, 固有値は以下の式で表される.
(第 j 番目の固有値)=$\dfrac{\alpha_j + \alpha'_j i}{\beta_j}$　　　(i:虚数単位)
このとき, 第 j 番目の固有値が実数であれば, α'_j には 0.0 が格納される. また, 複素数の場合は, それと共役な複素固有値が第 (j+1) 番目の要素に格納される. ただし, $\alpha'_j > 0$, $\alpha'_{j+1} < 0$ であり, β_j は常に 0 以上である (図 4−2 参照).

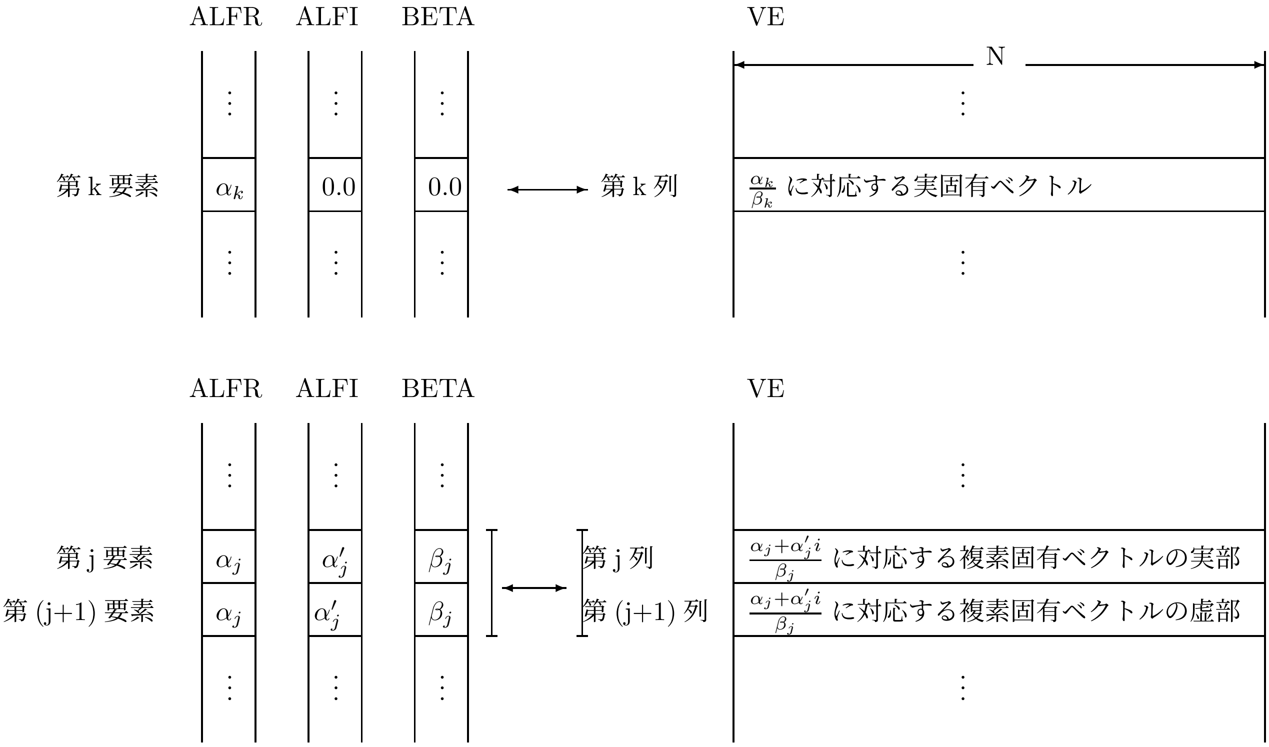
(b) IERR=2000 のとき, $\beta_j = 0$ に対応する固有値は, 非常に大きな値であることを意味している. このとき固有値を $(\alpha_j + \alpha'_{ji})/\beta_j$ で求めるとゼロ除算エラーが発生するので注意を要す.

(c) 求められた固有値に対応する固有ベクトルは, 配列 VE の各列に, 図 4−2 で示されるように格納される. すなわち, 第 j 番目の固有値が実数であれば, それに対応する固有値が配列 VE の第 j 列に格納される. また, 第 j 番目および (j+1) 番目の固有値が 1 組の共役な複素数であれば, 第 j 番目の固有値に対応する複素固有ベクトルの実部, 虚部が配列 VE の第 j 列第 (j+1) 列にそれぞれ格納される. この複素固有ベクトルに共役なベクトルが, 第 (j+1) 番目の固有値に対応する固有ベクトルとなる.

(d) 固有ベクトルは, 各要素の絶対値の最大値が 1.0 となるように正規化されている.

(e) 固有ベクトルを必要としない場合は 4.13.2 $\begin{Bmatrix} \mathrm{DCGGAN} \\ \mathrm{RCGGAN} \end{Bmatrix}$ を使用する.

図 4−2　固有値と固有ベクトルの格納形式



備　考

a.　$\alpha'_j > 0, \beta_j > 0$

(7) **使用例**

(a) 問 題

$$A = \begin{bmatrix} 50 & -60 & 50 & -27 & 6 & 6 \\ 38 & -28 & 27 & -17 & 5 & 5 \\ 27 & -17 & 27 & -17 & 5 & 5 \\ 27 & -28 & 38 & -17 & 5 & 5 \\ 27 & -28 & 27 & -17 & 16 & 5 \\ 27 & -28 & 27 & -17 & 5 & 16 \end{bmatrix},$$

$$B = \begin{bmatrix} 16 & 5 & 4 & 3 & -2 & 1 \\ 5 & 16 & 5 & 4 & -6 & 2 \\ 4 & 5 & 16 & 5 & -6 & 3 \\ 3 & 4 & 5 & 16 & -6 & 4 \\ 2 & 3 & 4 & 5 & -6 & 16 \\ 1 & 6 & 6 & 6 & -5 & 6 \end{bmatrix}$$

のとき, $A\boldsymbol{x} = \lambda B\boldsymbol{x}$ の全固有値とそれに対応する固有ベクトルを求める.

(b) 入力データ

行列 A, LNA=11, N=6, 行列 B, LNB=11, LNV=11

(c) 主プログラム

```
      PROGRAM BCGGAA
! *** EXAMPLE OF DCGGAA ***
      IMPLICIT REAL(8)(A-H,O-Z)
      PARAMETER ( ZERO = 0.0D0 )
      PARAMETER (LNA=11, LNB=11, LNV=11)
      DIMENSION A(LNA,LNA), B(LNB,LNB), VE(LNV,LNV),&
                ALFR(LNA), ALFI(LNA), BETA(LNA)
!
      READ(5,*) N
      DO 10 J=1, N
         READ(5,*) (A(J,I), I=1, N)
   10 CONTINUE
      DO 20 J=1, N
         READ(5,*) (B(J,I), I=1, N)
   20 CONTINUE
!
      WRITE(6,1000) N
      WRITE(6,1100) 'A'
      DO 30 J=1, N
         WRITE(6,1200) (A(J,I), I=1, N)
   30 CONTINUE
      WRITE(6,1100) 'B'
      DO 40 J=1, N
         WRITE(6,1200) (B(J,I), I=1, N)
   40 CONTINUE
!
      CALL DCGGAA(A,LNA,N,B,LNB,ALFR,ALFI,BETA,VE,LNV,IERR)
!
      WRITE(6,1300) IERR
!
      DO 100 J=1, N-1, 2
         WRITE(6,1400) 'EIGENVALUE ', 'EIGENVALUE '
         WRITE(6,1500) 'ALFR', ALFR(J), 'ALFR', ALFR(J+1)
         WRITE(6,1500) 'ALFI', ALFI(J), 'ALFI', ALFI(J+1)
         WRITE(6,1500) 'BETA', BETA(J), 'BETA', BETA(J+1)
         WRITE(6,1400) 'EIGENVECTOR', 'EIGENVECTOR'
         IF(ALFI(J).EQ.ZERO) THEN
            IF(ALFI(J+1).EQ.ZERO) THEN
               DO 50 I=1, N
                  WRITE(6,1600) VE(I,J), ZERO, VE(I,J+1), ZERO
   50          CONTINUE
            ELSE
               DO 60 I=1, N
                  WRITE(6,1600) VE(I,J), ZERO, VE(I,J+1), VE(I,J+2)
   60          CONTINUE
            ENDIF
         ELSE
            IF(ALFI(J+1).EQ.ZERO) THEN
               DO 70 I=1, N
                  WRITE(6,1600) VE(I,J-1),-VE(I,J), VE(I,J+1), ZERO
   70          CONTINUE
            ELSE
               IF(ALFI(J).LT.ZERO) THEN
               DO 80 I=1, N
                  WRITE(6,1600) VE(I,J-1),-VE(I,J), VE(I,J+1), VE(I,J+2)
   80          CONTINUE
               ELSE
               DO 90 I=1, N
                  WRITE(6,1600) VE(I,J), VE(I,J+1), VE(I,J),-VE(I,J+1)
   90          CONTINUE
               ENDIF
            ENDIF
         ENDIF
  100 CONTINUE
      IF(MOD(N,2).NE.0) THEN
         WRITE(6,1400) 'EIGENVALUE '
         WRITE(6,1500) 'ALFR', ALFR(N)
         WRITE(6,1500) 'ALFI', ALFI(N)
         WRITE(6,1500) 'BETA', BETA(N)
         WRITE(6,1400) 'EIGENVECTOR'
         IF(ALFI(N).EQ.ZERO) THEN
            DO 110 I=1, N
               WRITE(6,1600) VE(I,N), ZERO
  110       CONTINUE
         ELSE
            DO 120 I=1, N
               WRITE(6,1600) VE(I,N-1),-VE(I,N)
  120       CONTINUE
         ENDIF
      ENDIF
      STOP
!
 1000 FORMAT(' ',/,/,&
             ' ***  DCGGAA  ***',/,/,&
             '  **  INPUT  **',/,/,&
             '      N = ', I2)
 1100 FORMAT(' ',/,&
             '      INPUT MATRIX ',A1,/)
 1200 FORMAT(7X, 10(F7.1))
 1300 FORMAT(' ',/,/,&
             '  **  OUTPUT **',/,/,&
             '      IERR = ', I4)
 1400 FORMAT(' ',/,14X,2(A11,22X))
 1500 FORMAT(' ',2(7X,A4,' = ',1PD14.7,5X))
 1600 FORMAT(' ',2(5X,F12.8,' ,',F12.8,2X))
      END
```

(d) 出力結果

```
***  DCGGAA  ***

 **  INPUT  **

     N =  6

     INPUT MATRIX A

         50.0  -60.0   50.0  -27.0    6.0    6.0
         38.0  -28.0   27.0  -17.0    5.0    5.0
         27.0  -17.0   27.0  -17.0    5.0    5.0
         27.0  -28.0   38.0  -17.0    5.0    5.0
         27.0  -28.0   27.0  -17.0   16.0    5.0
         27.0  -28.0   27.0  -17.0    5.0   16.0

     INPUT MATRIX B

         16.0    5.0    4.0    3.0   -2.0    1.0
          5.0   16.0    5.0    4.0   -6.0    2.0
          4.0    5.0   16.0    5.0   -6.0    3.0
          3.0    4.0    5.0   16.0   -6.0    4.0
          2.0    3.0    4.0    5.0   -6.0   16.0
          1.0    6.0    6.0    6.0   -5.0    6.0


 **  OUTPUT **

     IERR =    0

             EIGENVALUE                       EIGENVALUE
       ALFR = -1.1634546D+01            ALFR =  1.1837890D+01
       ALFI =  0.0000000D+00            ALFI =  0.0000000D+00
       BETA =  9.3588312D-01            BETA =  3.9038739D+00

             EIGENVECTOR                      EIGENVECTOR
      -0.02489667 ,  0.00000000         0.19987257 ,  0.00000000
       0.25251218 ,  0.00000000        -0.19189298 ,  0.00000000
       0.19443080 ,  0.00000000        -0.24240591 ,  0.00000000
       0.20492111 ,  0.00000000        -0.21760871 ,  0.00000000
       1.00000000 ,  0.00000000        -1.00000000 ,  0.00000000
       0.16361487 ,  0.00000000        -0.44802812 ,  0.00000000

             EIGENVALUE                       EIGENVALUE
       ALFR =  6.6131567D+00            ALFR =  9.3466830D+00
       ALFI =  1.5653458D+01            ALFI = -2.2123763D+01
       BETA =  1.4195630D+01            BETA =  2.0063347D+01

             EIGENVECTOR                      EIGENVECTOR
      -0.90275293 , -0.16929219        -0.90275293 ,  0.16929219
      -0.61708834 ,  0.78689388        -0.61708834 , -0.78689388
       0.12677136 ,  0.78675507         0.12677136 , -0.78675507
       0.25033712 ,  0.21412357         0.25033712 , -0.21412357
      -0.41100766 ,  0.54117624        -0.41100766 , -0.54117624
      -0.22010716 ,  0.61947141        -0.22010716 , -0.61947141

             EIGENVALUE                       EIGENVALUE
       ALFR =  4.8318024D+00            ALFR =  5.7494668D+00
       ALFI =  7.8056571D+00            ALFI = -9.2881212D+00
       BETA =  1.0797432D+01            BETA =  1.2848100D+01

             EIGENVECTOR                      EIGENVECTOR
       0.46429690 , -0.29190301         0.46429690 ,  0.29190301
       0.06840004 , -0.86309624         0.06840004 ,  0.86309624
      -0.60524575 , -0.79603869        -0.60524575 ,  0.79603869
      -0.88694870 , -0.14801432        -0.88694870 ,  0.14801432
      -0.21908532 , -0.37959993        -0.21908532 ,  0.37959993
      -0.21818206 , -0.41595648        -0.21818206 ,  0.41595648
```

4.13.2 DCGGAN, RCGGAN
実行列 (一般化固有値問題) の全固有値

(1) **機 能**

実行列 (2次元配列型) の一般化固有値問題 $A\boldsymbol{x} = \lambda B\boldsymbol{x}$ (A, B:実行列) の全固有値をハウスホルダー法, コンビネーションシフトQZ法により求める.

(2) **使用法**

倍精度サブルーチン:

CALL DCGGAN (A, LNA, N, B, LNB, ALFR, ALFI, BETA, IERR)

単精度サブルーチン:

CALL RCGGAN (A, LNA, N, B, LNB, ALFR, ALFI, BETA, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: {32ビット整数版ではINTEGER(4)
64ビット整数版ではINTEGER(8)}

項番	引数名	型	大きさ	入出力	内　　　容
1	A	D R	LNA, N	入 力	実行列 A(2次元配列型)
				出 力	入力時の内容は保存されない
2	LNA	I	1	入 力	配列Aの整合寸法
3	N	I	1	入 力	行列 A, B の次数
4	B	D R	LNB, N	入 力	実行列 B (2次元配列型)
				出 力	入力時の内容は保存されない
5	LNB	I	1	入 力	配列Bの整合寸法
6	ALFR	D R	N	出 力	変換後の行列 A の対角成分の実部 (注意事項 (a) 参照)
7	ALFI	D R	N	出 力	変換後の行列 A の対角成分の虚部 (注意事項 (a) 参照)
8	BETA	D R	N	出 力	変換後の行列 B の対角成分 (注意事項 (a) 参照)
9	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNA}, \mathrm{LNB}$

(5) **エラーインディケータ**

IERR の値	意　　　　　味	処　理　内　容
0	正常終了.	
1000	$\mathrm{N}=1$ であった.	$\mathrm{ALFR}(1) \leftarrow \mathrm{sign}(\mathrm{B}(1,1)) \times \mathrm{A}(1,1)$, $\mathrm{ALFI}(1) \leftarrow 0.0$, $\mathrm{BETA}(1) \leftarrow \lvert\mathrm{B}(1,1)\rvert$ とする.
2000	配列 BETA の中に 0.0 がある.	処理を続ける (注意事項 (b) 参照).
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
$5000+i$	i 番目の固有値を求める段階で収束しなかった. $(1 \leq i \leq \mathrm{N})$	ALFR, ALFI, BETA の $i+1$ 番目より N 番目の要素は求められている.

(6) **注意事項**

(a) 固有値は添字の大きい方から求まり, ALFR, ALFI, BETA の配列に格納される.
ALFR, ALFI, BETA の第 i 番目の要素をそれぞれ $\alpha_j, \alpha'_j, \beta_j$ とすると, 固有値は以下の式で表される.
(第 j 番目の固有値)=$\dfrac{\alpha_j+\alpha'_j i}{\beta_j}$　　　　　(i:虚数単位)
このとき, 第 j 番目の固有値が実数であれば, α'_j には 0.0 が格納される. また, 複素数の場合は, それと共役な複素固有値が第 (j+1) 番目の要素に格納される. ただし, $\alpha'_j > 0$, $\alpha'_{j+1} < 0$ であり, β_j は常に正である (図 4−2 参照).

(b) IERR=2000 のとき, $\beta_j = 0$ に対応する固有値は非常に大きな値であることを意味している. このとき固有値を $(\alpha_j + \alpha'_{ji})/\beta_j$ で求めるとゼロ除算エラーが発生するので注意を要す.

4.14 実対称行列 (2 次元配列型) (上三角型) の一般化固有値問題 $(Ax = \lambda Bx)$

4.14.1 DCGSAA, RCGSAA
実対称行列 (一般化固有値問題 $Ax = \lambda Bx$, B: 正定値) の全固有値・全固有ベクトル

(1) **機 能**

実対称行列 (2 次元配列型) (上三角型) の一般化固有値問題

$$A\boldsymbol{x} = \lambda B\boldsymbol{x} \quad (A : \text{実対称行列}, B : \text{正値実対称行列})$$

をコレスキー法を用いて標準の固有値問題に変換し, ハウスホルダー法, QR 法により全固有値とそれに対応する全固有ベクトルを求める.

(2) **使用法**

倍精度サブルーチン:
CALL DCGSAA (A, LNA, N, B, LNB, E, W1, IERR)
単精度サブルーチン:
CALL RCGSAA (A, LNA, N, B, LNB, E, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: $\left\{\begin{array}{l}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{array}\right\}$

項番	引数名	型	大きさ	入出力	内　　容
1	A	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	LNA, N	入 力	実対称行列 A (2 次元配列型)(上三角型)
				出 力	各固有値に対応する固有ベクトル (列ベクトル)
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A, B の次数
4	B	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	LNB, N	入 力	正値対称行列 B (2 次元配列型) (上三角型)
				出 力	入力時の内容は保存されない
5	LNB	I	1	入 力	配列 B の整合寸法
6	E	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	N	出 力	固有値
7	W1	$\left\{\begin{matrix}\mathrm{D}\\\mathrm{R}\end{matrix}\right\}$	2 × N	ワーク	作業領域
8	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNA}, \mathrm{LNB}$

(5) エラーインディケータ

IERR の値	意　　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	$\mathrm{E(1)} \leftarrow \dfrac{\mathrm{A(1,1)}}{\mathrm{B(1,1)}}$, $\mathrm{A(1,1)} \leftarrow \dfrac{1.0}{\sqrt{\mathrm{B(1,1)}}}$ とする.
2100	B の対角要素に非常に小さい絶対値をもつものがあった.	固有ベクトルの精度が低い可能性があるが, 処理を続ける.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
4000	B が正定値でなかった.	
$5000 + i$	固有値を求める段階で収束しなかった. $(1 \leq i \leq \mathrm{N})$	$\mathrm{E}(1), \cdots, \mathrm{E}(i-1)$ にそれまでに求まった固有値が入る (ただし順不同). このとき固有ベクトルは求まらない.

(6) 注意事項

(a) 配列 A, B には, 上三角部分のみにデータが格納されていればよい.

(b) 固有値は小さい順に格納される.

(c) 各固有ベクトル $\boldsymbol{v}_i$ は $\boldsymbol{v}_j^T B \boldsymbol{v}_k = \delta_{j,k}$ となるように正規直交化される.

(d) 固有ベクトルを必要としないときは, 4.14.2 $\left\{\begin{matrix}\text{DCGSAN}\\\text{RCGSAN}\end{matrix}\right\}$ を使用する.

(7) 使用例

(a) 問 題

$$A = \begin{bmatrix} 2 & 1 & 1 & 2 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 2 \\ 2 & 1 & 2 & 4 \end{bmatrix}$$

$$B = \begin{bmatrix} 153 & 31 & 58 & -58 \\ 31 & 153 & -53 & 58 \\ 58 & -58 & 153 & 31 \\ -58 & 58 & 31 & 153 \end{bmatrix}$$

のとき, $A\boldsymbol{x} = \lambda B\boldsymbol{x}$ の全固有値とそれに対応する固有ベクトルを求める.

(b) 入力データ

行列 A, LNA=11, N=4, 行列 B, LNB=11

(c) 主プログラム

```
      PROGRAM BCGSAA
! *** EXAMPLE OF DCGSAA ***
      IMPLICIT REAL(8) (A-H,O-Z)
      PARAMETER ( LNA = 11, LNB = 11 )
      DIMENSION A(LNA,LNA), B(LNB,LNB), E(LNA), W1(2*LNA)
!
      READ(5,*) N
      DO 10 J=1, N
```

```
         READ(5,*) (A(J,I), I=J, N)
   10 CONTINUE
      DO 20 J=1, N
         READ(5,*) (B(J,I), I=J, N)
   20 CONTINUE
!
      WRITE(6,1000) N
      WRITE(6,1100) 'A'
      DO 30 J=1, N
         WRITE(6,1200) (A(I,J), I=1, J-1), (A(J,I), I=J, N)
   30 CONTINUE
      WRITE(6,1100) 'B'
      DO 40 J=1, N
         WRITE(6,1200) (B(I,J), I=1, J-1), (B(J,I), I=J, N)
   40 CONTINUE
!
      CALL DCGSAA(A,LNA,N,B,LNB,E,W1,IERR)
!
      WRITE(6,1300) IERR
!
      DO 60 K=1, N-3, 4
         WRITE(6,1400) ('EIGENVALUE ', I=1, 4)
         WRITE(6,1500) (E(I), I=K, K+3)
         WRITE(6,1400) ('EIGENVECTOR', I=1, 4)
         DO 50 J=1, N
            WRITE(6,1500) (A(J,I), I=K, K+3)
   50    CONTINUE
   60 CONTINUE
      IF(MOD(N,4).NE.0) THEN
         WRITE(6,1400) ('EIGENVALUE ', I=N/4*4+1, N)
         WRITE(6,1500) (E(I), I=N/4*4+1, N)
         WRITE(6,1400) ('EIGENVECTOR', I=N/4*4+1, N)
         DO 70 J=1, N
            WRITE(6,1500) (A(J,I), I=N/4*4+1, N)
   70    CONTINUE
      ENDIF
      STOP
!
 1000 FORMAT(' ',/,/,&
             ' ***  DCGSAA  ***',/,/,&
             '  **  INPUT  **',/,/,&
             '       N = ', I2)
 1100 FORMAT(' ',/,&
             '       INPUT MATRIX ',A1,/)
 1200 FORMAT(7X, 11(F8.1))
 1300 FORMAT(' ',/,/,&
             '  **  OUTPUT  **',/,/,&
             '       IERR = ', I4)
 1400 FORMAT(' ',/,1X, 4(5X, A11, 2X))
 1500 FORMAT(' ', 4(2X, 1PD14.7, 2X))
      END
```

(d) 出力結果

```
***  DCGSAA  ***

 **  INPUT  **

     N =  4

     INPUT MATRIX A

           2.0     1.0     1.0     2.0
           1.0     1.0     1.0     1.0
           1.0     1.0     2.0     2.0
           2.0     1.0     2.0     4.0

     INPUT MATRIX B

         153.0    31.0    58.0   -58.0
          31.0   153.0   -58.0    58.0
          58.0   -58.0   153.0    31.0
         -58.0    58.0    31.0   153.0


 **  OUTPUT  **

     IERR =    0

     EIGENVALUE         EIGENVALUE         EIGENVALUE         EIGENVALUE
   6.4779811D-04      5.3688291D-03      2.7447086D-02      2.1668091D-01

     EIGENVECTOR        EIGENVECTOR        EIGENVECTOR        EIGENVECTOR
   2.9405960D-02      4.9839235D-02     -1.6115522D-02      2.0451432D-01
  -4.6877602D-02      3.7709049D-02      6.8634504D-02     -1.9262477D-01
   3.1083549D-02     -1.9357438D-02      8.5979167D-02     -1.9157548D-01
  -1.9570768D-02     -3.3245027D-02      1.4513123D-03      2.0962854D-01
```

4.14.2 DCGSAN, RCGSAN
実対称行列 (一般化固有値問題 $Ax = \lambda Bx$, B: 正定値) の全固有値

(1) 機 能

実対称行列 (2 次元配列型) (上三角型) の一般化固有値問題

$$A\boldsymbol{x} = \lambda B\boldsymbol{x} \quad (A : \text{実対称行列}, B : \text{正値実対称行列})$$

をコレスキー法を用いて標準の固有値問題に変換し, ハウスホルダー法, QR 法により全固有値を求める.

(2) 使用法

倍精度サブルーチン:

CALL DCGSAN (A, LNA, N, B, LNB, E, W1, IERR)

単精度サブルーチン:

CALL RCGSAN (A, LNA, N, B, LNB, E, W1, IERR)

(3) 引 数

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: { 32 ビット整数版では INTEGER(4)
　　64 ビット整数版では INTEGER(8) }

項番	引数名	型	大きさ	入出力	内　　容
1	A	D R	LNA, N	入 力	実対称行列 A (2 次元配列型)(上三角型)
				出 力	入力時の内容は保存されない
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A, B の次数
4	B	D R	LNB, N	入 力	正値対称行列 B (2 次元配列型) (上三角型)
				出 力	入力時の内容は保存されない
5	LNB	I	1	入 力	配列 B の整合寸法
6	E	D R	N	出 力	固有値
7	W1	D R	N	ワーク	作業領域
8	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a) $0 < \mathrm{N} \leq \mathrm{LNA}, \mathrm{LNB}$

(5) エラーインディケータ

IERR の値	意　　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	$\mathrm{E}(1) \leftarrow \dfrac{\mathrm{A}(1,1)}{\mathrm{B}(1,1)}$ とする.
2100	B の対角要素に非常に小さい絶対値をもつものがあった.	処理を続ける.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
4000	B が正定値でなかった.	
$5000 + i$	固有値を求める段階で収束しなかった. $(1 \leq i \leq \mathrm{N})$	$\mathrm{E}(1), \cdots, \mathrm{E}(i-1)$ にそれまでに求まった固有値が入る (ただし順不同).

(6) 注意事項

(a) 配列 A, B には, 上三角部分のみにデータが格納されていればよい.

(b) 固有値は小さい順に格納される.

4.14.3 DCGSSS, RCGSSS
実対称行列 (一般化固有値問題 $Ax = \lambda Bx$, B: 正定値) の固有値・固有ベクトル

(1) **機 能**

実対称行列 (2 次元配列型) (上三角型) の一般化固有値問題

$$A\boldsymbol{x} = \lambda B\boldsymbol{x} \quad (A: \text{実対称行列}, B: \text{正値実対称行列})$$

をコレスキー法を用いて標準の固有値問題に変換し, ハウスホルダー法, 無平方根 QR 法またはバイセクション法により, 大きい方から m 個, または小さい方から m 個の固有値を求め, それに対応する固有ベクトルを逆反復法により求める.

(2) **使用法**

倍精度サブルーチン:

CALL DCGSSS (A, LNA, N, B, LNB, EPS, E, M, VE, LNV, ISW, IW1, W1, IERR)

単精度サブルーチン:

CALL RCGSSS (A, LNA, N, B, LNB, EPS, E, M, VE, LNV, ISW, IW1, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\left\{\begin{array}{l}\text{32 ビット整数版では INTEGER(4)}\\ \text{64 ビット整数版では INTEGER(8)}\end{array}\right\}$

項番	引数名	型	大きさ	入出力	内　容
1	A	$\left\{\begin{matrix}\text{D}\\\text{R}\end{matrix}\right\}$	LNA, N	入 力	実対称行列 A (2 次元配列型) (上三角型)
				出 力	入力時の内容は保存されない
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A, B の次数
4	B	$\left\{\begin{matrix}\text{D}\\\text{R}\end{matrix}\right\}$	LNB, N	入 力	正値対称行列 B (2 次元配列型) (上三角型)
				出 力	狭義の上三角部分は保存されない
5	LNB	I	1	入 力	配列 B の整合寸法
6	EPS	$\left\{\begin{matrix}\text{D}\\\text{R}\end{matrix}\right\}$	1	入 力	固有値の収束判定のための絶対誤差の上限を与えるパラメータ (注意事項 (d) 参照)
7	E	$\left\{\begin{matrix}\text{D}\\\text{R}\end{matrix}\right\}$	M	出 力	固有値
8	M	I	1	入 力	求めたい固有値の個数 m
9	VE	$\left\{\begin{matrix}\text{D}\\\text{R}\end{matrix}\right\}$	LNV, M	出 力	各固有値に対応する固有ベクトル (列ベクトル)
10	LNV	I	1	入 力	配列 VE の整合寸法
11	ISW	I	1	入 力	処理スイッチ ISW $\geq$ 0: 大きい方から固有値を求める. ISW $<$ 0: 小さい方から固有値を求める.
12	IW1	I	M	出 力	固有ベクトルフラグ (注意事項 (e) 参照)
13	W1	$\left\{\begin{matrix}\text{D}\\\text{R}\end{matrix}\right\}$	9×N	ワーク	作業領域
14	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a) $0 < \mathrm{N} \le \mathrm{LNA}, \mathrm{LNB}, \mathrm{LNV}$

(b) $0 < \mathrm{M} \le \mathrm{N}$

(5) エラーインディケータ

IERR の値	意　　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	$\mathrm{E}(1) \leftarrow \dfrac{\mathrm{A}(1,1)}{\mathrm{B}(1,1)}$, $\mathrm{VE}(1,1) \leftarrow \dfrac{1.0}{\sqrt{\mathrm{B}(1,1)}}$ とする.
2000	固有ベクトルを求める逆反復で最大反復回数をこえた.	固有ベクトルに一部精度の低いものがあるが, 処理は続ける (注意事項 (e) 参照).
2100	B の対角要素に非常に小さい絶対値をもつものがあった.	固有ベクトルの精度が低い可能性があるが, 処理を続ける.
3000	制限条件 (a) または (b) を満足しなかった.	処理を打ち切る.
4000	B が正定値でなかった.	

(6) 注意事項

(a) 配列 A, B には, 上三角部分のみにデータが格納されていればよい.

(b) 固有値は ISW≥0 のときには大きい順に, ISW<0 のときには小さい順に格納される.

(c) 固有値は無平方根 QR 法とバイセクション法を内部で適切に切り分けて計算している.

(d) EPS ≤ 0 のときは内部で自動的に最適値を設定する. 通常は自動設定されるように負の値を設定されたい. なお, EPS はバイセクション法で固有値を求めるときに使用される.

(e) 逆反復法で最大反復回数をこえた場合 (IERR=2000 出力時) について

- $\mathrm{IW1}(i) = 0$ の場合 :
  i 番目の固有ベクトル計算は正常終了している.
- $\mathrm{IW1}(i) \ne 0$ の場合 :
  i 番目の固有ベクトル計算は収束条件が満たされず, 固有ベクトルの精度は低い.
  この場合 $\mathrm{IW1}(i)$ は, 反復回数が設定される.

なお正常終了時 (IERR=0 出力時) は, $\mathrm{IW1}(i) \leftarrow 0$ が設定される.

(f) 各固有ベクトル $\boldsymbol{v}_i$ は $\boldsymbol{v}_j^T B \boldsymbol{v}_k = \delta_{j,k}$ となるように正規直交化される.

(g) 固有ベクトルを必要としないときは, 4.14.4 $\left\{\begin{matrix}\mathrm{DCGSSN}\\ \mathrm{RCGSSN}\end{matrix}\right\}$ を使用する.

(7) **使用例**

(a) 問 題

$$A = \begin{bmatrix} 611 & 196 & -192 & 407 & -8 & -52 & -49 & 29 \\ 196 & 899 & 113 & -192 & -71 & -43 & -8 & -44 \\ -192 & 113 & 899 & 196 & 61 & 49 & 8 & 52 \\ 407 & -192 & 196 & 611 & 8 & 44 & 59 & -23 \\ -8 & -71 & 61 & 8 & 411 & -599 & 208 & 208 \\ -52 & -43 & 49 & 44 & -599 & 411 & 208 & 208 \\ -49 & -8 & 8 & 59 & 208 & 208 & 99 & -911 \\ 29 & -44 & 52 & -23 & 208 & 208 & -911 & 99 \end{bmatrix}$$

$$B = \begin{bmatrix} 170 & 18 & 33 & -21 & -17 & 13 & 25 & -36 \\ 18 & 171 & -21 & 22 & 13 & -17 & -36 & 25 \\ 33 & -21 & 171 & 18 & 25 & -36 & -17 & 13 \\ -21 & 22 & 18 & 171 & -36 & 25 & 13 & -17 \\ -17 & 13 & 25 & -36 & 171 & 18 & 33 & -21 \\ 13 & -17 & -36 & 25 & 18 & 171 & -21 & -3 \\ 25 & -36 & -17 & 13 & 33 & -21 & 171 & 18 \\ -36 & 25 & 13 & -17 & -21 & -3 & 18 & 171 \end{bmatrix}$$

のとき, $A\boldsymbol{x} = \lambda B\boldsymbol{x}$ の固有値を小さい順に 2 個とそれに対応する固有ベクトルを求める.

(b) 入力データ

行列 A, LNA=11, N=8, 行列 B, LNB=11, EPS=−1.0, M=2, LNV=10, ISW=−1

(c) 主プログラム

```
      PROGRAM BCGSSS
! *** EXAMPLE OF DCGSSS ***
      IMPLICIT REAL(8) (A-H,O-Z)
      PARAMETER ( LNA = 11, LNB = 11, LNV = 10 )
      DIMENSION A(LNA,LNA), B(LNB,LNB), E(LNA), VE(LNV,LNV),&
                IW1(LNA), W1(9*LNA)
!
      READ(5,*) N, M
      DO 10 J=1, N
         READ(5,*) (A(J,I), I=J, N)
   10 CONTINUE
      DO 20 J=1, N
         READ(5,*) (B(J,I), I=J, N)
   20 CONTINUE
!
      WRITE(6,1000) N, M
      WRITE(6,1100) 'A'
      DO 30 J=1, N
         WRITE(6,1200) (A(I,J), I=1, J-1), (A(J,I), I=J, N)
   30 CONTINUE
      WRITE(6,1100) 'B'
      DO 40 J=1, N
         WRITE(6,1200) (B(I,J), I=1, J-1), (B(J,I), I=J, N)
   40 CONTINUE
!
      ISW = -1
      EPS = -1.0D0
!
      CALL DCGSSS(A,LNA,N,B,LNB,EPS,E,M,VE,LNV,ISW,IW1,W1,IERR)
!
      WRITE(6,1300) IERR
!
      DO 60 K=1, M-3, 4
         WRITE(6,1400) ('EIGENVALUE ', I=1, 4)
         WRITE(6,1500) (E(I), I=K, K+3)
         WRITE(6,1400) ('EIGENVECTOR', I=1, 4)
         DO 50 J=1, N
            WRITE(6,1500) (VE(J,I), I=K, K+3)
   50    CONTINUE
   60 CONTINUE
      IF(MOD(M,4).NE.0) THEN
         WRITE(6,1400) ('EIGENVALUE ', I=M/4*4+1, M)
         WRITE(6,1500) (E(I), I=M/4*4+1, M)
         WRITE(6,1400) ('EIGENVECTOR', I=M/4*4+1, M)
         DO 70 J=1, N
            WRITE(6,1500) (VE(J,I), I=M/4*4+1, M)
```

```
   70    CONTINUE
      ENDIF
      STOP
!
 1000 FORMAT(' ',/,/,&
             ' ***  DCGSSS  ***',/,/,&
             '  **  INPUT  **',/,/,&
             '      N = ', I2,/,/,&
             '      M = ', I2)
 1100 FORMAT(' ',/,&
             '      INPUT MATRIX ',A1,/)
 1200 FORMAT(7X, 11(F8.1))
 1300 FORMAT(' ',/,/,&
             '  **  OUTPUT  **',/,/,&
             '      IERR = ', I4)
 1400 FORMAT(' ',/,1X, 4(5X, A11, 2X))
 1500 FORMAT(' ', 4(2X, 1PD14.7, 2X))
      END
```

(d) 出力結果

```
*** DCGSSS ***

 **  INPUT  **

     N =  8

     M =  2

     INPUT MATRIX A

          611.0   196.0  -192.0   407.0    -8.0   -52.0   -49.0    29.0
          196.0   899.0   113.0  -192.0   -71.0   -43.0    -8.0   -44.0
         -192.0   113.0   899.0   196.0    61.0    49.0     8.0    52.0
          407.0  -192.0   196.0   611.0     8.0    44.0    59.0   -23.0
           -8.0   -71.0    61.0     8.0   411.0  -599.0   208.0   208.0
          -52.0   -43.0    49.0    44.0  -599.0   411.0   208.0   208.0
          -49.0    -8.0     8.0    59.0   208.0   208.0    99.0  -911.0
           29.0   -44.0    52.0   -23.0   208.0   208.0  -911.0    99.0

     INPUT MATRIX B

          170.0    18.0    33.0   -21.0   -17.0    13.0    25.0   -36.0
           18.0   171.0   -21.0    22.0    13.0   -17.0   -36.0    25.0
           33.0   -21.0   171.0    18.0    25.0   -36.0   -17.0    13.0
          -21.0    22.0    18.0   171.0   -36.0    25.0    13.0   -17.0
          -17.0    13.0    25.0   -36.0   171.0    18.0    33.0   -21.0
           13.0   -17.0   -36.0    25.0    18.0   171.0   -21.0    -3.0
           25.0   -36.0   -17.0    13.0    33.0   -21.0   171.0    18.0
          -36.0    25.0    13.0   -17.0   -21.0    -3.0    18.0   171.0


 **  OUTPUT  **

     IERR =    0

     EIGENVALUE        EIGENVALUE
  -5.3020806D+00    -1.0369304D-15

     EIGENVECTOR       EIGENVECTOR
   7.8865043D-04    -3.2898475D-03
   1.4571715D-03    -6.5796950D-03
   6.2438782D-04     6.5796950D-03
  -1.6786293D-03     3.2898475D-03
  -2.4707363D-02    -4.6057865D-02
  -1.9017760D-02    -4.6057865D-02
   4.7852016D-02    -2.3028932D-02
   4.4548878D-02    -2.3028932D-02
```

4.14.4 DCGSSN, RCGSSN
実対称行列 (一般化固有値問題 $Ax = \lambda Bx$, B: 正定値) の固有値

(1) **機 能**

実対称行列 (2 次元配列型) (上三角型) の一般化固有値問題

$$A\boldsymbol{x} = \lambda B\boldsymbol{x} \quad (A : \text{実対称行列}, B : \text{正値実対称行列})$$

をコレスキー法を用いて標準の固有値問題に変換し, ハウスホルダー法, 無平方根 QR 法またはバイセクション法により, 大きい方から m 個, または小さい方から m 個の固有値を求める.

(2) **使用法**

倍精度サブルーチン:
CALL DCGSSN (A, LNA, N, B, LNB, EPS, E, M, ISW, W1, IERR)
単精度サブルーチン:
CALL RCGSSN (A, LNA, N, B, LNB, EPS, E, M, ISW, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)} \\ \text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$

項番	引数名	型	大きさ	入出力	内　　容
1	A	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	LNA, N	入 力	実対称行列 A (2 次元配列型) (上三角型)
				出 力	入力時の内容は保存されない
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A, B の次数
4	B	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	LNB, N	入 力	正値対称行列 B (2 次元配列型) (上三角型)
				出 力	入力時の内容は保存されない
5	LNB	I	1	入 力	配列 B の整合寸法
6	EPS	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	1	入 力	固有値の収束判定のための絶対誤差の上限を与えるパラメータ (注意事項 (d) 参照)
7	E	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	M	出 力	固有値
8	M	I	1	入 力	求めたい固有値の個数 m
9	ISW	I	1	入 力	処理スイッチ ISW $\geq$ 0: 大きい方から固有値を求める. ISW $<$ 0: 小さい方から固有値を求める.
10	W1	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	$5 \times \mathrm{N}$	ワーク	作業領域
11	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \leq \mathrm{LNA}, \mathrm{LNB}$

(b) $0 < \mathrm{M} \leq \mathrm{N}$

(5) **エラーインディケータ**

IERR の値	意　　　　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	$\mathrm{E(1)} \leftarrow \dfrac{\mathrm{A(1,1)}}{\mathrm{B(1,1)}}$ とする.
2100	B の対角要素に非常に小さい絶対値をもつものがあった.	処理を続ける.
3000	制限条件 (a) または (b) を満足しなかった.	処理を打ち切る.
4000	B が正定値でなかった.	

(6) **注意事項**

(a) 配列 A, B には, 上三角部分のみにデータが格納されていればよい.

(b) 固有値は ISW≥0 のときには大きい順に, ISW<0 のときには小さい順に格納される.

(c) 固有値は無平方根 QR 法とバイセクション法を内部で適切に切り分けて計算している.

(d) EPS $\leq$ 0 のときは内部で自動的に最適値を設定する. 通常は自動設定されるように負の値を設定されたい. なお, EPS はバイセクション法で固有値を求めるときに使用される.

4.14.5 DCGSEE, RCGSEE
実対称行列 (一般化固有値問題 $Ax = \lambda Bx$, B: 正定値) の固有値・固有ベクトル (区間指定)

(1) **機 能**

実対称行列 (2 次元配列型) (上三角型) の一般化固有値問題

$$A\boldsymbol{x} = \lambda B\boldsymbol{x} \quad (A : \text{実対称行列}, B : \text{正値実対称行列})$$

をコレスキー法を用いて標準の固有値問題に変換し, ハウスホルダー法, バイセクション法により, 指定した区間の固有値を昇順で m 個, もしくは降順で m 個求め, それに対応する固有ベクトルを逆反復法により求める.

(2) **使用法**

倍精度サブルーチン:

CALL DCGSEE (A, LNA, N, B, LNB, EPS, E, M, E1, E2, VE, LNV, IW1, W1, IERR)

単精度サブルーチン:

CALL RCGSEE (A, LNA, N, B, LNB, EPS, E, M, E1, E2, VE, LNV, IW1, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: $\left\{\begin{array}{l}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{array}\right\}$

項番	引数名	型	大きさ	入出力	内　容
1	A	D R	LNA, N	入 力	実対称行列 A (2 次元配列型) (上三角型)
				出 力	入力時の内容は保存されない
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A, B の次数
4	B	D R	LNB, N	入 力	正値対称行列 B (2 次元配列型) (上三角型)
				出 力	狭義の上三角部分は保存されない
5	LNB	I	1	入 力	配列 B の整合寸法
6	EPS	D R	1	入 力	固有値の収束判定のための絶対誤差の上限を与えるパラメータ (注意事項 (b) 参照)
7	E	D R	M	出 力	固有値
8	M	I	1	入 力	求めたい固有値の個数 m
				出 力	求めた固有値の個数
9	E1	D R	1	入 力	E1 < E2: E1 から昇順で固有値を求める. (E2 は, 上限) E1 > E2: E1 から降順で固有値を求める. (E2 は, 下限) (注意事項 (c)(d) 参照)
10	E2	D R	1	入 力	

項番	引数名	型	大きさ	入出力	内　　容
11	VE	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	LNV, M	出 力	各固有値に対応する固有ベクトル (列ベクトル)
12	LNV	I	1	入 力	配列 VE の整合寸法
13	IW1	I	M	出 力	固有ベクトルフラグ (注意事項 (e) 参照)
14	W1	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	9×N	ワーク	作業領域
15	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $0 < \mathrm{N} \le \mathrm{LNA, LNB, LNV}$

(b) $0 < \mathrm{M} \le \mathrm{N}$

(5) **エラーインディケータ**

IERR の値	意　　味	処 理 内 容
0	正常終了.	
1000	N = 1 であった.	$\mathrm{E(1)} \leftarrow \dfrac{\mathrm{A(1,1)}}{\mathrm{B(1,1)}}$, $\mathrm{VE(1,1)} \leftarrow \dfrac{1.0}{\sqrt{\mathrm{B(1,1)}}}$ とする.
1500	区間 [E1, E2] に存在する固有値は M 個未満であった.	区間 [E1, E2] 内にある全ての固有値, 固有ベクトルを求める. 引数 M には求めた固有値の個数を出力する.
2000	固有ベクトルを求める逆反復で最大反復回数をこえた.	固有ベクトルに一部精度の低いものがあるが, 処理は続ける (注意事項 (e) 参照).
2100	B の対角要素に非常に小さい絶対値をもつものがあった.	固有ベクトルの精度が低い可能性があるが, 処理を続ける.
3000	制限条件 (a) または (b) を満足しなかった.	処理を打ち切る.
4000	B が正定値でなかった.	

(6) **注意事項**

(a) 配列 A, B には, 上三角部分のみにデータが格納されていればよい.

(b) EPS $\le$ 0 のときは内部で自動的に最適値を設定する. 通常は自動設定されるように負の値を設定されたい. なお, EPS はバイセクション法で固有値を求めるときに使用される.

(c) 固有値は E1<E2 のときには小さい順, E1>E2 のときには大きい順に格納される.

(d) E1=E2 のとき, 区間 [E1-EPS, E1+EPS] にある固有値が求まる. しかし, 通常は E1≠E2 となるように設定されたい.

(e) 逆反復法で最大反復回数をこえた場合 (IERR=2000 出力時) について

- $\mathrm{IW1}(i) = 0$ の場合 :
  i 番目の固有ベクトル計算は正常終了している.

- IW1(i) $\neq 0$ の場合 :
  i 番目の固有ベクトル計算は収束条件が満たされず, 固有ベクトルの精度は低い.
  この場合 IW1(i) は, 反復回数が設定される.

なお正常終了時 (IERR=0 出力時) は, IW1(i) $\leftarrow 0$ が設定される.

(f) 各固有ベクトル $\boldsymbol{v}_i$ は $\boldsymbol{v}_j^T B \boldsymbol{v}_k = \delta_{j,k}$ となるように正規直交化される.

(g) 固有ベクトルを必要としないときは, 4.14.6 $\left\{\begin{matrix}\text{DCGSEN}\\ \text{RCGSEN}\end{matrix}\right\}$ を使用する.

(7) **使用例**

(a) 問 題

$$
A = \begin{bmatrix}
611 & 196 & -192 & 407 & -8 & -52 & -49 & 29 \\
196 & 899 & 113 & -192 & -71 & -43 & -8 & -44 \\
-192 & 113 & 899 & 196 & 61 & 49 & 8 & 52 \\
407 & -192 & 196 & 611 & 8 & 44 & 59 & -23 \\
-8 & -71 & 61 & 8 & 411 & -599 & 208 & 208 \\
-52 & -43 & 49 & 44 & -599 & 411 & 208 & 208 \\
-49 & -8 & 8 & 59 & 208 & 208 & 99 & -911 \\
29 & -44 & 52 & -23 & 208 & 208 & -911 & 99
\end{bmatrix}
$$

$$
B = \begin{bmatrix}
170 & 18 & 33 & -21 & -17 & 13 & 25 & -36 \\
18 & 171 & -21 & 22 & 13 & -17 & -36 & 25 \\
33 & -21 & 171 & 18 & 25 & -36 & -17 & 13 \\
-21 & 22 & 18 & 171 & -36 & 25 & 13 & -17 \\
-17 & 13 & 25 & -36 & 171 & 18 & 33 & -21 \\
13 & -17 & -36 & 25 & 18 & 171 & -21 & -3 \\
25 & -36 & -17 & 13 & 33 & -21 & 171 & 18 \\
-36 & 25 & 13 & -17 & -21 & -3 & 18 & 171
\end{bmatrix}
$$

のとき, $A\boldsymbol{x} = \lambda B\boldsymbol{x}$ の固有値を区間 [0.001, 0.1] で 2 個求め (昇順), それに対応する固有ベクトルを求める.

(b) 入力データ

行列 A, LNA=11, N=8, 行列 B, LNB=11, EPS=−1.0, M=2, E1=0.001, E2=0.1, LNV=10

(c) 主プログラム

```
      PROGRAM BCGSEE
! *** EXAMPLE OF DCGSEE ***
      IMPLICIT REAL(8) (A-H,O-Z)
      PARAMETER ( LNA = 11, LNB = 11, LNV = 10 )
      DIMENSION A(LNA,LNA), B(LNB,LNB), E(LNA), VE(LNV,LNV),&
                IW1(LNA), W1(9*LNA)
!
      READ(5,*) N, M, E1, E2
      DO 10 J=1, N
         READ(5,*) (A(J,I), I=J, N)
   10 CONTINUE
      DO 20 J=1, N
         READ(5,*) (B(J,I), I=J, N)
   20 CONTINUE
!
      WRITE(6,1000) N, M, E1, E2
      WRITE(6,1100) 'A'
      DO 30 J=1, N
         WRITE(6,1200) (A(I,J), I=1, J-1), (A(J,I), I=J, N)
   30 CONTINUE
      WRITE(6,1100) 'B'
      DO 40 J=1, N
         WRITE(6,1200) (B(I,J), I=1, J-1), (B(J,I), I=J, N)
   40 CONTINUE
!
      EPS = -1.0D0
!
      CALL DCGSEE(A,LNA,N,B,LNB,EPS,E,M,E1,E2,VE,LNV,IW1,W1,IERR)
```

```
!
      WRITE(6,1300) IERR
!
      DO 60 K=1, M-3, 4
         WRITE(6,1400) (’EIGENVALUE ’, I=1, 4)
         WRITE(6,1500) (E(I), I=K, K+3)
         WRITE(6,1400) (’EIGENVECTOR’, I=1, 4)
         DO 50 J=1, N
            WRITE(6,1500) (VE(J,I), I=K, K+3)
   50    CONTINUE
   60 CONTINUE
      IF(MOD(M,4).NE.0) THEN
         WRITE(6,1400) (’EIGENVALUE ’, I=M/4*4+1, M)
         WRITE(6,1500) (E(I), I=M/4*4+1, M)
         WRITE(6,1400) (’EIGENVECTOR’, I=M/4*4+1, M)
         DO 70 J=1, N
            WRITE(6,1500) (VE(J,I), I=M/4*4+1, M)
   70    CONTINUE
      ENDIF
      STOP
!
 1000 FORMAT(1X,/,/,&
             1X,’***  DCGSEE  ***’,/,/,&
             1X,’ **  INPUT  **’,/,/,&
             1X,’N = ’, I4,’  M = ’, I4,/,/,&
             1X,’E1= ’, 1PD14.7,’  E2= ’, 1PD14.7)
 1100 FORMAT(1X,/,&
             1X,’     INPUT MATRIX ’,A1,/)
 1200 FORMAT(1X, 6X, 11(F8.1))
 1300 FORMAT(1X,/,/,&
             1X,’ **  OUTPUT  **’,/,/,&
             1X,’     IERR = ’, I4)
 1400 FORMAT(1X,/,1X, 4(5X, A11, 2X))
 1500 FORMAT(1X, 4(2X, 1PD14.7, 2X))
      END
```

(d) 出力結果

```
***  DCGSEE  ***

 **  INPUT  **

N =    4  M =    2

E1=  1.0000000D-03  E2=  1.0000000D-01

     INPUT MATRIX A

           2.0     1.0     1.0     2.0
           1.0     1.0     1.0     1.0
           1.0     1.0     2.0     2.0
           2.0     1.0     2.0     4.0

     INPUT MATRIX B

         153.0    31.0    58.0   -58.0
          31.0   153.0   -58.0    58.0
          58.0   -58.0   153.0    31.0
         -58.0    58.0    31.0   153.0


 **  OUTPUT  **

     IERR =    0

     EIGENVALUE         EIGENVALUE
   5.3688291D-03     2.7447086D-02

     EIGENVECTOR        EIGENVECTOR
  -4.9839235D-02     1.6115522D-02
  -3.7709049D-02    -6.8634504D-02
   1.9357438D-02    -8.5979167D-02
   3.3245027D-02    -1.4513123D-03
```

4.14.6 DCGSEN, RCGSEN
実対称行列 (一般化固有値問題 $Ax = \lambda Bx$, B: 正定値) の固有値 (区間指定)

(1) **機 能**

実対称行列 (2 次元配列型) (上三角型) の一般化固有値問題

$$A\boldsymbol{x} = \lambda B\boldsymbol{x} \quad (A : \text{実対称行列}, B : \text{正値実対称行列})$$

をコレスキー法を用いて標準の固有値問題に変換し, ハウスホルダー法, バイセクション法により, 指定した区間の固有値を昇順で m 個, もしくは降順で m 個求める.

(2) **使用法**

倍精度サブルーチン:

CALL DCGSEN (A, LNA, N, B, LNB, EPS, E, M, E1, E2, W1, IERR)

単精度サブルーチン:

CALL RCGSEN (A, LNA, N, B, LNB, EPS, E, M, E1, E2, W1, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: $\left\{\begin{array}{l}\text{32 ビット整数版では INTEGER(4)}\\ \text{64 ビット整数版では INTEGER(8)}\end{array}\right\}$

項番	引数名	型	大きさ	入出力	内　容
1	A	D R	LNA, N	入 力	実対称行列 A (2 次元配列型) (上三角型)
				出 力	入力時の内容は保存されない
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A, B の次数
4	B	D R	LNB, N	入 力	正値対称行列 B (2 次元配列型) (上三角型)
				出 力	入力時の内容は保存されない
5	LNB	I	1	入 力	配列 B の整合寸法
6	EPS	D R	1	入 力	固有値の収束判定のための絶対誤差の上限を与えるパラメータ (注意事項 (b) 参照)
7	E	D R	M	出 力	固有値
8	M	I	1	入 力	求めたい固有値の個数 m
				出 力	求めた固有値の個数
9	E1	D R	1	入 力	E1 < E2 の場合 : E1 から昇順で固有値を求める. (E2 は, 上限)
10	E2	D R	1	入 力	E1 > E2 の場合 : E1 から降順で固有値を求める. (E2 は, 下限) (注意事項 (c)(d) 参照)
11	W1	D R	5 × N	ワーク	作業領域
12	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a) $0 < \mathrm{N} \le \mathrm{LNA}, \mathrm{LNB}$

(b) $0 < \mathrm{M} \le \mathrm{N}$

(5) エラーインディケータ

IERR の値	意　　味	処　理　内　容
0	正常終了.	
1000	$\mathrm{N} = 1$ であった.	$\mathrm{E}(1) \leftarrow \dfrac{\mathrm{A}(1,1)}{\mathrm{B}(1,1)}$ とする.
1500	区間 [E1, E2] に存在する固有値は M 個未満であった.	区間 [E1, E2] 内にある全ての固有値, 固有ベクトルを求める. 引数 M には求めた固有値の個数を出力する.
2100	B の対角要素に非常に小さい絶対値をもつものがあった.	処理を続ける.
3000	制限条件 (a) または (b) を満足しなかった.	処理を打ち切る.
4000	B が正定値でなかった.	

(6) 注意事項

(a) 配列 A, B には, 上三角部分のみにデータが格納されていればよい.

(b) $\mathrm{EPS} \le 0$ のときは内部で自動的に最適値を設定する. 通常は自動設定されるように負の値を設定されたい. なお, EPS はバイセクション法で固有値を求めるときに使用される.

(c) 固有値は E1<E2 のときには小さい順, E1>E2 のときには大きい順に格納される.

(d) E1=E2 のとき, 区間 [E1-EPS, E1+EPS] にある固有値が求まる. しかし, 通常は E1≠E2 となるように設定されたい.

4.15 実対称行列 (2 次元配列型) (上三角型) の一般化固有値問題 $(ABx = \lambda x)$

4.15.1 DCGJAA, RCGJAA

実対称行列 (一般化固有値問題 $ABx = \lambda x$, B: 正定値) の全固有値・全固有ベクトル

(1) **機 能**

実対称行列 (2 次元配列型) (上三角型) (実引数型) の一般化固有値問題

$ABx = \lambda x$ (A: 実対称行列 , B: 正定値実対称行列)

をコレスキー分解, ハウスホルダー法, QR 法で解いて, 全固有値 λ とそれに対応する全固有ベクトル $\boldsymbol{x}$ を求める.

(2) **使用法**

倍精度サブルーチン:

CALL DCGJAA (A, LNA, N, B, LNB, E, WORK, IERR)

単精度サブルーチン:

CALL RCGJAA (A, LNA, N, B, LNB, E, WORK, IERR)

(3) **引 数**

D:倍精度実数型 Z:倍精度複素数型
R:単精度実数型 C:単精度複素数型
I: $\left\{ \begin{array}{l} \text{32 ビット整数版では INTEGER(4)} \\ \text{64 ビット整数版では INTEGER(8)} \end{array} \right\}$

項番	引数名	型	大きさ	入出力	内 容
1	A	$\begin{Bmatrix} \text{D} \\ \text{R} \end{Bmatrix}$	LNA, N	入 力	実対称行列 A
				出 力	固有ベクトル $\boldsymbol{x}$
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A,B の次数
4	B	$\begin{Bmatrix} \text{D} \\ \text{R} \end{Bmatrix}$	LNB, N	入 力	実対称行列 B
				出 力	入力時の内容は保存されない
5	LNB	I	1	入 力	配列 B の整合寸法
6	E	$\begin{Bmatrix} \text{D} \\ \text{R} \end{Bmatrix}$	N	出 力	固有値 λ
7	WORK	$\begin{Bmatrix} \text{D} \\ \text{R} \end{Bmatrix}$	2 × N	ワーク	作業領域
8	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $1 \leq \text{N} \leq \text{LNA}, \text{LNB}$

(5) **エラーインディケータ**

IERR の値	意　　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	$\mathrm{E}(1) \leftarrow \mathrm{A}(1,1) \times \mathrm{B}(1,1)$, $\mathrm{A}(1,1) \leftarrow \dfrac{1.0}{\sqrt{\mathrm{B}(1,1)}}$ とする.
2100	B の対角要素に非常に小さい絶対値をもつものがあった.	結果の精度が低い可能性があるが, 処理を続ける.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
4000	B が正定値でなかった.	
$5000 + i$	固有値を求める段階で収束しなかった. $(1 \leq i \leq \mathrm{N})$	$\mathrm{E}(1), \cdots, \mathrm{E}(i-1)$ にそれまでに求まった固有値が入る (ただし順不同). このとき固有ベクトルは求まらない.

(6) **注意事項**

(a) 配列 A, B には, 上三角部分のみにデータが格納されていればよい.

(b) 固有値は小さい順に格納される.

(c) 各固有ベクトル $\boldsymbol{v}_i$ は $\boldsymbol{v}_j^T B \boldsymbol{v}_k = \delta_{j,k}$ となるように正規直交化される.

(d) 固有ベクトルを必要としないときは, 4.15.2 $\left\{\begin{matrix}\mathrm{DCGJAN}\\ \mathrm{RCGJAN}\end{matrix}\right\}$ を使用する.

(e) 行列 A のみが正定値行列であるときは, 4.16.1 $\left\{\begin{matrix}\mathrm{DCGKAA}\\ \mathrm{RCGKAA}\end{matrix}\right\}$ を使用する.

(7) **使用例**

(a) 問 題

4 次実正定値対称行列 A, B

$$A = \begin{bmatrix} 1.07692 & 0.28571 & 0.09733 & 0.04887 \\ 0.28571 & 1.02041 & 0.26316 & 0.08610 \\ 0.09733 & 0.26316 & 1.00917 & 0.25676 \\ 0.04887 & 0.08610 & 0.25676 & 1.00518 \end{bmatrix}$$

$$B = \begin{bmatrix} 1.04762 & 0.18841 & 0.05996 & 0.02968 \\ 0.18841 & 1.01235 & 0.17460 & 0.05314 \\ 0.05996 & 0.17460 & 1.00552 & 0.17073 \\ 0.02968 & 0.05314 & 0.17073 & 1.00312 \end{bmatrix}$$

に対して, 非対称行列 AB の固有値と固有ベクトルを求める.

注 ここで, A, B は, 以下で定義される実正定値対称行列である.

$$A = P(3.0, 4), \quad B = P(5.0, 4)$$

ただし,

$$P(a^2, \mathrm{N})_{i,j} = 2\int_0^\infty \cos(ait)\cos(ajt)e^{-t}dt \quad (1 \le i \le \mathrm{N}; 1 \le j \le \mathrm{N})$$

また各々の行列の固有値は, 全て, 次数 N に無関係に定まる有界区間内に存在する.

(b) 入力データ

N=4, LNA=LNB=4, 実正定値対称行列 A, B

(c) 主プログラム

```
      PROGRAM BCGJAA
! *** EXAMPLE OF DCGJAA ***
      IMPLICIT NONE
!
      INTEGER N,LNA,LNB
      PARAMETER( N = 4, LNA = 4, LNB = 4 )
      INTEGER IERR,I,J,L
      REAL(8) A(LNA,N),B(LNB,N)
      REAL(8) E(N),WORK(2*N)
      REAL(8) ONE,TRE,FIV
      PARAMETER( ONE = 1.D0, TRE = 3.D0, FIV = 5.D0 )
!
      WRITE(6,6000) N, LNA, LNB
      DO 100 I=1,N
      DO 110 J=1,N
         A(I,J)= ONE/(ONE+TRE*DBLE(I+J)**2)+ONE/(ONE+TRE*DBLE(I-J)**2)
         B(I,J)= ONE/(ONE+FIV*DBLE(I+J)**2)+ONE/(ONE+FIV*DBLE(I-J)**2)
  110 CONTINUE
  100 CONTINUE
      WRITE(6,6010)
      DO 120 I=1,N
         WRITE(6,6020) A(I,1),A(I,2),A(I,3),A(I,4)
  120 CONTINUE
      WRITE(6,6030)
      DO 130 I=1,N
         WRITE(6,6020) B(I,1),B(I,2),B(I,3),B(I,4)
  130 CONTINUE
!
      CALL DCGJAA(A, LNA, N, B, LNB, E, WORK, IERR)
!
      WRITE(6,6040) IERR
      DO 140 I=1,N,2
         WRITE(6,6050) (' EIGENVALUE',L=1,2)
         WRITE(6,6060) E(I),E(I+1)
         WRITE(6,6050) ('EIGENVECTOR',L=1,2)
         WRITE(6,6060) A(1,I),A(1,I+1)
         WRITE(6,6060) A(2,I),A(2,I+1)
         WRITE(6,6060) A(3,I),A(3,I+1)
         WRITE(6,6060) A(4,I),A(4,I+1)
  140 CONTINUE
!
      STOP
 6000 FORMAT(/,&
             1X,'***  DCGJAA  ***',/,/,&
             1X,' **  INPUT  **',/,/,&
             1X,'     N = ',I4, '    LNA = ',I4,'  LNB = ',I4,/)
 6010 FORMAT(/,&
             1X,'     INPUT MATRIX A',/)
```

```
6020 FORMAT(1X,3X,4(2X,F9.5))
6030 FORMAT(/,&
            1X,'     INPUT MATRIX B',/)
6040 FORMAT(/,&
            1X,' **  OUTPUT  **',/,/,&
            1X,'     IERR =',I5,/)
6050 FORMAT(/,&
            1X,7X,A11,22X,A11)
6060 FORMAT(1X,5X,1PD14.7,19X,1PD14.7)
     END
```

(d) 出力結果

```
***  DCGJAA  ***

 **  INPUT  **

     N =    4    LNA =    4  LNB =    4


     INPUT MATRIX A

       1.07692    0.28571    0.09733    0.04887
       0.28571    1.02041    0.26316    0.08610
       0.09733    0.26316    1.00917    0.25676
       0.04887    0.08610    0.25676    1.00518

     INPUT MATRIX B

       1.04762    0.18841    0.05996    0.02968
       0.18841    1.01235    0.17460    0.05314
       0.05996    0.17460    1.00552    0.17073
       0.02968    0.05314    0.17073    1.00312

 **  OUTPUT  **

     IERR =    0


        EIGENVALUE                       EIGENVALUE
      5.0334859D-01                    7.0649951D-01

       EIGENVECTOR                      EIGENVECTOR
     -3.5988845D-01                   -5.7276742D-01
      7.0243936D-01                    4.9740733D-01
     -7.1756352D-01                    4.1982158D-01
      4.0575850D-01                   -6.2631989D-01

        EIGENVALUE                       EIGENVALUE
      1.1519432D+00                    2.1334368D+00

       EIGENVECTOR                      EIGENVECTOR
      5.9964094D-01                    4.1404961D-01
      2.5727788D-01                    4.9417207D-01
     -3.8926246D-01                    4.5974369D-01
     -6.0442734D-01                    3.2426391D-01
```

4.15.2 DCGJAN, RCGJAN
実対称行列 (一般化固有値問題 $ABx = \lambda x$, B: 正定値) の全固有値

(1) 機 能

実対称行列 (2 次元配列型) (上三角型) (実引数型) の一般化固有値問題

$$AB\boldsymbol{x} = \lambda\boldsymbol{x} \ (A: \text{実対称行列}, \ B: \text{正定値実対称行列})$$

をコレスキー分解, ハウスホルダー法, QR 法で解いて, 全固有値 λ を求める.

(2) 使用法

倍精度サブルーチン:
CALL DCGJAN (A, LNA, N, B, LNB, E, WORK, IERR)
単精度サブルーチン:
CALL RCGJAN (A, LNA, N, B, LNB, E, WORK, IERR)

(3) 引 数

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: $\left\{\begin{array}{l}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{array}\right\}$

項番	引数名	型	大きさ	入出力	内　容
1	A	$\left\{\begin{matrix}\text{D}\\\text{R}\end{matrix}\right\}$	LNA, N	入 力	実対称行列 A
				出 力	入力時の内容は保存されない
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A,B の次数
4	B	$\left\{\begin{matrix}\text{D}\\\text{R}\end{matrix}\right\}$	LNB, N	入 力	実対称行列 B
				出 力	入力時の内容は保存されない
5	LNB	I	1	入 力	配列 B の整合寸法
6	E	$\left\{\begin{matrix}\text{D}\\\text{R}\end{matrix}\right\}$	N	出 力	固有値 λ
7	WORK	$\left\{\begin{matrix}\text{D}\\\text{R}\end{matrix}\right\}$	2 × N	ワーク	作業領域
8	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a) $1 \leq \text{N} \leq \text{LNA}, \text{LNB}$

(5) エラーインディケータ

IERR の値	意　　　　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	E(1) ← A(1, 1) × B(1, 1) とする.
2100	B の対角要素に非常に小さい絶対値をもつものがあった.	結果の精度が低い可能性があるが, 処理を続ける.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
4000	B が正定値でなかった.	
$5000 + i$	固有値を求める段階で収束しなかった. ($1 \le i \le$ N)	E(1), ..., E($i - 1$) にそれまでに求まった固有値が入る (ただし順不同).

(6) 注意事項

(a) 配列 A, B には, 上三角部分のみにデータが格納されていればよい.

(b) 固有値は小さい順に格納される.

(c) 固有ベクトルを必要とするときは, 4.15.1 $\left\{\begin{matrix}\text{DCGJAA}\\ \text{RCGJAA}\end{matrix}\right\}$ を使用する.

(d) 行列 A のみが正定値行列であるときは, 4.16.2 $\left\{\begin{matrix}\text{DCGKAN}\\ \text{RCGKAN}\end{matrix}\right\}$ を使用する.

4.16 実対称行列 (2 次元配列型) (上三角型) の一般化固有値問題 $(BAx = \lambda x)$

4.16.1 DCGKAA, RCGKAA
実対称行列 (一般化固有値問題 $BAx = \lambda x$, B: 正定値) の全固有値・全固有ベクトル

(1) **機能**

実対称行列 (2 次元配列型) (上三角型) (実引数型) の一般化固有値問題

$$B A \boldsymbol{x} = \lambda \boldsymbol{x} \ (A: \text{実対称行列}, \ B: \text{正定値実対称行列})$$

をコレスキー分解, ハウスホルダー法, QR 法で解いて, 全固有値 λ とそれに対応する全固有ベクトル $\boldsymbol{x}$ を求める.

(2) **使用法**

倍精度サブルーチン:

CALL DCGKAA (A, LNA, N, B, LNB, E, WORK, IERR)

単精度サブルーチン:

CALL RCGKAA (A, LNA, N, B, LNB, E, WORK, IERR)

(3) **引数**

D:倍精度実数型 Z:倍精度複素数型
R:単精度実数型 C:単精度複素数型

I: {32 ビット整数版では INTEGER(4)
64 ビット整数版では INTEGER(8)}

項番	引数名	型	大きさ	入出力	内容
1	A	{D R}	LNA, N	入力	実対称行列 A
				出力	固有ベクトル $\boldsymbol{x}$
2	LNA	I	1	入力	配列 A の整合寸法
3	N	I	1	入力	行列 A, B の次数
4	B	{D R}	LNB, N	入力	実対称行列 B
				出力	入力時の内容は保存されない
5	LNB	I	1	入力	配列 B の整合寸法
6	E	{D R}	N	出力	固有値 λ
7	WORK	{D R}	2 × N	ワーク	作業領域
8	IERR	I	1	出力	エラーインディケータ

(4) **制限条件**

(a) $1 \leq \mathrm{N} \leq \mathrm{LNA}, \mathrm{LNB}$

(5) エラーインディケータ

IERR の値	意　　　　味	処　理　内　容
0	正常終了.	
1000	$\mathrm{N} = 1$ であった.	$\mathrm{E}(1) \leftarrow \mathrm{A}(1,1) \times \mathrm{B}(1,1)$, $\mathrm{A}(1,1) \leftarrow \sqrt{\mathrm{B}(1,1)}$ とする.
2100	B の対角要素に非常に小さい絶対値をもつものがあった.	結果の精度が低い可能性があるが, 処理を続ける.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
4000	B が正定値でなかった.	
$5000 + i$	固有値を求める段階で収束しなかった. $(1 \leq i \leq \mathrm{N})$	$\mathrm{E}(1), \ldots, \mathrm{E}(i-1)$ にそれまでに求まった固有値が入る (ただし順不同). このとき固有ベクトルは求まらない.

(6) 注意事項

(a) 配列 A, B には, 上三角部分のみにデータが格納されていればよい.

(b) 固有値は小さい順に格納される.

(c) 各固有ベクトル $\boldsymbol{v}_i$ は $\boldsymbol{v}_j^T B^{-1} \boldsymbol{v}_k = \delta_{j,k}$ となるように正規直交化される.

(d) 固有ベクトルを必要としないときは, 4.16.2 $\left\{\begin{matrix}\mathrm{DCGKAN}\\ \mathrm{RCGKAN}\end{matrix}\right\}$ を使用する.

(e) 行列 A のみが正定値行列であるときは, 4.15.1 $\left\{\begin{matrix}\mathrm{DCGJAA}\\ \mathrm{RCGJAA}\end{matrix}\right\}$ を使用する.

(7) 使用例

(a) 問 題

4 次実正定値対称行列 A, B

$$A = \begin{bmatrix} 1.07692 & 0.28571 & 0.09733 & 0.04887 \\ 0.28571 & 1.02041 & 0.26316 & 0.08610 \\ 0.09733 & 0.26316 & 1.00917 & 0.25676 \\ 0.04887 & 0.08610 & 0.25676 & 1.00518 \end{bmatrix}$$

$$B = \begin{bmatrix} 1.04762 & 0.18841 & 0.05996 & 0.02968 \\ 0.18841 & 1.01235 & 0.17460 & 0.05314 \\ 0.05996 & 0.17460 & 1.00552 & 0.17073 \\ 0.02968 & 0.05314 & 0.17073 & 1.00312 \end{bmatrix}$$

に対して, 非対称行列 AB の固有値と固有ベクトルを求める.

注 ここで, A, B は, 以下で定義される実正定値対称行列である.

$$A = P(3.0, 4), \quad B = P(5.0, 4)$$

ただし,

$$P(a^2, \mathrm{N})_{i,j} = 2\int_0^\infty \cos(ait)\cos(ajt)e^{-t}dt \quad (1 \leq i \leq \mathrm{N}; 1 \leq j \leq \mathrm{N})$$

また各々の行列の固有値は, 全て, 次数 N に無関係に定まる有界区間内に存在する.

(b) 入力データ

N=4, LNA=LNB=4, 実正定値対称行列 A, B

(c) 主プログラム

```
      PROGRAM BCGKAA
! *** EXAMPLE OF DCGKAA ***
      IMPLICIT NONE
!
      INTEGER N,LNA,LNB
      PARAMETER( N = 4, LNA = 4, LNB = 4 )
      INTEGER IERR,I,J,L
      REAL(8) A(LNA,N),B(LNB,N)
      REAL(8) E(N),WORK(2*N)
      REAL(8) ONE, TRE, FIV
      PARAMETER( ONE = 1.D0, TRE = 3.D0, FIV = 5.D0 )
!
      WRITE(6,6000) N, LNA, LNB
      DO 100 I=1,N
      DO 110 J=1,N
         A(I,J)= ONE/(ONE+TRE*DBLE(I+J)**2)+ONE/(ONE+TRE*DBLE(I-J)**2)
         B(I,J)= ONE/(ONE+FIV*DBLE(I+J)**2)+ONE/(ONE+FIV*DBLE(I-J)**2)
  110 CONTINUE
  100 CONTINUE
      WRITE(6,6010)
      DO 120 I=1,N
         WRITE(6,6020) A(I,1),A(I,2),A(I,3),A(I,4)
  120 CONTINUE
      WRITE(6,6030)
      DO 130 I=1,N
         WRITE(6,6020) B(I,1),B(I,2),B(I,3),B(I,4)
  130 CONTINUE
!
      CALL DCGKAA(A, LNA, N, B, LNB, E, WORK, IERR)
!
      WRITE(6,6040) IERR
      DO 140 I=1,N,2
         WRITE(6,6050) (’ EIGENVALUE’,L=1,2)
         WRITE(6,6060) E(I),E(I+1)
         WRITE(6,6050) (’EIGENVECTOR’,L=1,2)
         WRITE(6,6060) A(1,I),A(1,I+1)
         WRITE(6,6060) A(2,I),A(2,I+1)
         WRITE(6,6060) A(3,I),A(3,I+1)
         WRITE(6,6060) A(4,I),A(4,I+1)
  140 CONTINUE
!
      STOP
 6000 FORMAT(/,&
             1X,’***  DCGKAA  ***’,/,/,&
             1X,’ **  INPUT  **’,/,/,&
             1X,’     N = ’,I4, ’    LNA = ’,I4,’  LNB = ’,I4,/)
 6010 FORMAT(/,&
             1X,’     INPUT MATRIX A’,/)
 6020 FORMAT(1X,3X,4(2X,F9.5))
 6030 FORMAT(/,&
             1X,’     INPUT MATRIX B’,/)
 6040 FORMAT(/,&
             1X,’ **  OUTPUT  **’,/,/,&
             1X,’     IERR =’,I5,/)
 6050 FORMAT(/,&
             1X,7X,A11,22X,A11)
 6060 FORMAT(1X,5X,1PD14.7,19X,1PD14.7)
      END
```

(d) 出力結果

```
***  DCGKAA  ***

 **  INPUT  **

     N =    4    LNA =    4  LNB =    4


     INPUT MATRIX A

       1.07692    0.28571    0.09733    0.04887
       0.28571    1.02041    0.26316    0.08610
       0.09733    0.26316    1.00917    0.25676
       0.04887    0.08610    0.25676    1.00518

     INPUT MATRIX B

       1.04762    0.18841    0.05996    0.02968
       0.18841    1.01235    0.17460    0.05314
       0.05996    0.17460    1.00552    0.17073
       0.02968    0.05314    0.17073    1.00312

 **  OUTPUT  **

     IERR =    0


        EIGENVALUE                         EIGENVALUE
      5.0334859D-01                      7.0649951D-01

       EIGENVECTOR                        EIGENVECTOR
     -2.7566971D-01                     -4.9973959D-01
```

```
  5.3958111D-01                    4.3565256D-01
 -5.5118459D-01                    3.6771142D-01
  3.1116215D-01                   -5.4715726D-01

    EIGENVALUE                       EIGENVALUE
  1.1519432D+00                    2.1334368D+00

   EIGENVECTOR                      EIGENVECTOR
  6.3538913D-01                    5.6406228D-01
  2.7334189D-01                    6.7578767D-01
 -4.1372916D-01                    6.2875822D-01
 -6.4130226D-01                    4.4231632D-01
```

4.16.2 DCGKAN, RCGKAN
実対称行列 (一般化固有値問題 $BAx = \lambda x$, B: 正定値) の全固有値

(1) 機 能

実対称行列 (2 次元配列型) (上三角型) (実引数型) の一般化固有値問題

$$BA\boldsymbol{x} = \lambda\boldsymbol{x} \ (A: \text{実対称行列}\ ,\ B: \text{正定値実対称行列})$$

をコレスキー分解, ハウスホルダー法, QR 法で解いて, 全固有値 λ を求める.

(2) 使用法

倍精度サブルーチン:

CALL DCGKAN (A, LNA, N, B, LNB, E, WORK, IERR)

単精度サブルーチン:

CALL RCGKAN (A, LNA, N, B, LNB, E, WORK, IERR)

(3) 引 数

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: {32 ビット整数版では INTEGER(4)
64 ビット整数版では INTEGER(8)}

項番	引数名	型	大きさ	入出力	内　　容
1	A	{D R}	LNA, N	入 力	実対称行列 A
				出 力	入力時の内容は保存されない
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A,B の次数
4	B	{D R}	LNB, N	入 力	実対称行列 B
				出 力	入力時の内容は保存されない
5	LNB	I	1	入 力	配列 B の整合寸法
6	E	{D R}	N	出 力	固有値 λ
7	WORK	{D R}	$2 \times$ N	ワーク	作業領域
8	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a) $1 \leq \mathrm{N} \leq \mathrm{LNA}, \mathrm{LNB}$

(5) **エラーインディケータ**

IERR の値	意　　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	$\mathrm{E}(1) \leftarrow \mathrm{A}(1,1) \times \mathrm{B}(1,1)$ とする.
2100	B の対角要素に非常に小さい絶対値をもつものがあった.	結果の精度が低い可能性があるが, 処理を続ける.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
4000	B が正定値でなかった.	
$5000 + i$	固有値を求める段階で収束しなかった. $(1 \leq i \leq \mathrm{N})$	$\mathrm{E}(1), \ldots, \mathrm{E}(i-1)$ にそれまでに求まった固有値が入る (ただし順不同).

(6) **注意事項**

(a) 配列 A, B には, 上三角部分のみにデータが格納されていればよい.

(b) 固有値は小さい順に格納される.

(c) 固有ベクトルを必要とするときは, 4.16.1 $\left\{\begin{array}{l}\mathrm{DCGKAA}\\ \mathrm{RCGKAA}\end{array}\right\}$ を使用する.

(d) 行列 A のみが正定値行列であるときは, 4.15.2 $\left\{\begin{array}{l}\mathrm{DCGJAN}\\ \mathrm{RCGJAN}\end{array}\right\}$ を使用する.

4.17 エルミート行列 (2 次元配列型) (上三角型) (実数引数型) の一般化固有値問題 ($Az = \lambda Bz$)

4.17.1 ZCGRAA, CCGRAA

エルミート行列 (一般化固有値問題 $Az = \lambda Bz$, B: 正定値) の全固有値・全固有ベクトル

(1) **機 能**

エルミート行列 (2 次元配列型) (上三角型) (実数引数型) の一般化固有値問題

$$Az = \lambda Bz \ (A:\ \text{エルミート行列},\ B:\ \text{正定値エルミート行列})$$

をコレスキー分解, ハウスホルダー法, QR 法で解いて, 全固有値 λ とそれに対応する全固有ベクトル $\boldsymbol{z}$ を求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZCGRAA (AR, AI, LNA, N, BR, BI, LNB, E, WORK, IERR)

単精度サブルーチン:

CALL CCGRAA (AR, AI, LNA, N, BR, BI, LNB, E, WORK, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: 32 ビット整数版では INTEGER(4)
　 64 ビット整数版では INTEGER(8)

項番	引数名	型	大きさ	入出力	内　　容
1	AR	D R	LNA, N	入 力	エルミート行列 A の実部
				出 力	固有ベクトル $\boldsymbol{z}$ の実部が代入される
2	AI	D R	LNA, N	入 力	エルミート行列 A の虚部
				出 力	固有ベクトル $\boldsymbol{z}$ の虚部が代入される
3	LNA	I	1	入 力	配列 AR,AI の整合寸法
4	N	I	1	入 力	行列 A, B の次数
5	BR	D R	LNB, N	入 力	エルミート行列 B の実部
				出 力	入力時の内容は保存されない
6	BI	D R	LNB, N	入 力	エルミート行列 B の虚部
				出 力	入力時の内容は保存されない
7	LNB	I	1	入 力	配列 BR,BI の整合寸法
8	E	D R	N	出 力	固有値 λ
9	WORK	D R	4 × N	ワーク	作業領域
10	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a) $1 \le \mathrm{N} \le \mathrm{LNA}, \mathrm{LNB}$

(5) エラーインディケータ

IERR の値	意　　　味	処　理　内　容
0	正常終了.	
1000	$\mathrm{N} = 1$ であった.	$\mathrm{E}(1) \leftarrow \dfrac{\mathrm{AR}(1,1)}{\mathrm{BR}(1,1)}$, $\mathrm{AR}(1,1) \leftarrow \dfrac{1.0}{\sqrt{\mathrm{BR}(1,1)}}$, $\mathrm{AI}(1,1) \leftarrow 0.0$ とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
4000	B が正定値でなかった.	
5000	標準の固有値問題の解法過程において, 処理が続行不能となった.	

(6) 注意事項

(a) 配列 AR,AI,BR, BI には, 上三角部分のみにデータが格納されていればよい.

(b) 固有値は小さい順に格納される.

(c) 各固有ベクトル $\boldsymbol{v}_i$ は $\boldsymbol{v}_j^* B \boldsymbol{v}_k = \delta_{j,k}$ となるように正規直交化される.

(d) 固有ベクトルを必要としないときは, 4.17.2 $\begin{Bmatrix} \mathrm{ZCGRAN} \\ \mathrm{CCGRAN} \end{Bmatrix}$ を使用する.

(7) 使用例

(a) 問 題

4 次エルミート行列

$$A = \begin{bmatrix} 8 & 3 & 1-2i & -1-2i \\ 3 & 9 & 1+2i & -1+2i \\ 1+2i & 1-2i & 10 & -3 \\ -1+2i & -1-2i & -3 & 11 \end{bmatrix}$$

と, 各成分が共役である 4 次エルミート行列

$$B = \begin{bmatrix} 8 & 3 & 1+2i & -1+2i \\ 3 & 9 & 1-2i & -1-2i \\ 1-2i & 1+2i & 10 & -3 \\ -1-2i & -1+2i & -3 & 11 \end{bmatrix}$$

に対する一般化固有値問題を解いて, 固有ベクトルも求める.

(b) 入力データ

行列 A, $\mathrm{LNA} = 4$, $\mathrm{N} = 4$, 行列 B, $\mathrm{LNB} = 4$

(c) 主プログラム

```
      PROGRAM ACGRAA
      IMPLICIT REAL(8)(A-H,O-Z)
      PARAMETER ( N=4 , LN=4 )
      REAL(8)       E(N), WORK(4*N)
      COMPLEX(8)  KEEP(LN,N),VM
```

```
      REAL(8)      AR(LN,N),BR(LN,N),AI(LN,N),BI(LN,N),SCL(N)
      KEEP(1,1)=( 8.D0, 0.D0)
      KEEP(2,2)=( 9.D0, 0.D0)
      KEEP(3,3)=( 10.D0, 0.D0)
      KEEP(4,4)=( 11.D0, 0.D0)
      KEEP(1,2)=( 3.D0, 0.D0)
      KEEP(1,3)=( 1.D0, 2.D0)
      KEEP(1,4)=(-1.D0, 2.D0)
      KEEP(2,3)=( 1.D0,-2.D0)
      KEEP(2,4)=(-1.D0,-2.D0)
      KEEP(3,4)=(-3.D0, 0.D0)
      METHOD=0
      WRITE(6,60)
      IF(METHOD.EQ.0) WRITE(6,70) N
      DO 1000 I=1,N
      DO 1001 J=I,N
        BR(I,J)=DBLE (KEEP(I,J))
        BI(I,J)=DIMAG(KEEP(I,J))
        KEEP(J,I)=CONJG(KEEP(I,J))
        AR(I,J)=BR(I,J)
        AI(I,J)=-BI(I,J)
        AR(J,I)=AR(I,J)
        AI(J,I)=-AI(I,J)
        BR(J,I)=BR(I,J)
        BI(J,I)=-BI(I,J)
 1001 CONTINUE
 1000 CONTINUE
      IF(METHOD.EQ.0) THEN
         WRITE(6,80)
         DO 1150 I=1,N
         WRITE(6,5000) AR(I,1),AI(I,1),AR(I,2),AI(I,2),&
                       AR(I,3),AI(I,3),AR(I,4),AI(I,4)
 1150    CONTINUE
         WRITE(6,90)
         DO 1250 I=1,N
         WRITE(6,5000) BR(I,1),BI(I,1),BR(I,2),BI(I,2),&
                       BR(I,3),BI(I,3),BR(I,4),BI(I,4)
 1250    CONTINUE
      ENDIF
      IF(METHOD.GT.0) THEN
         CALL BWINTA(AR, AI, BR, BI, LN, N, SCL, IERRA)
         IF(IERRA.GT.0) STOP
         WRITE(6,95)
         DO 1100 I=1,N
         DO 1101 J=I,N
            WRITE(6,6000) AR(I,J),AI(I,J),BR(I,J),BI(I,J),I,J
 1101    CONTINUE
 1100    CONTINUE
      ENDIF
!
      CALL ZCGRAA(AR,AI, LN, N, BR,BI, LN, E, WORK, IERR)
!
      IF(METHOD.GT.0) THEN
         CALL BWINTB(AR, AI, LN, N, SCL, IERRB)
         IF(IERRB.GT.0) STOP
      ENDIF
      IF(METHOD.GT.0) THEN
         WRITE(6,96)
      ELSE
         WRITE(6,97)
      ENDIF
      WRITE(6,98) IERR
      DO 2000 I=1,N,2
         VM=(0.D0,0.D0)
         DO 2100 J=1,N
         DO 2101 K=1,N
            VM=VM + KEEP(J,K)*CMPLX(AR(K,I), AI(K,I), KIND=8)&
                             *CMPLX(AR(J,I),-AI(J,I), KIND=8)
 2101    CONTINUE
 2100    CONTINUE
         WRITE(6,6500) ('EIGENVALUE ',L=1,2)
         WRITE(6,7000) E(I),E(I+1)
         WRITE(6,6500) ('EIGENVECTOR',L=1,2)
         WRITE(6,8000) AR(1,I),AI(1,I),AR(1,I+1),AI(1,I+1)
         WRITE(6,8000) AR(2,I),AI(2,I),AR(2,I+1),AI(2,I+1)
         WRITE(6,8000) AR(3,I),AI(3,I),AR(3,I+1),AI(3,I+1)
         WRITE(6,8000) AR(4,I),AI(4,I),AR(4,I+1),AI(4,I+1)
 2000 CONTINUE
      STOP
   60 FORMAT(1X,  ' *** ZCGRAA *** ',/,/)
   70 FORMAT(1X, ' *** INPUT ***',/,/,'     N= ',I3,/,/)
   80 FORMAT(1X,'     INPUT MATRIX A ( REAL , IMAGINARY )',/)
   90 FORMAT(1X,/,/,'     INPUT MATRIX B ( REAL , IMAGINARY )',/)
   95 FORMAT(1X,' *** INPUT *** (SCALED) ')
   96 FORMAT(1X,' *** OUTPUT *** (SCALED) ')
   97 FORMAT(1X,/,/, ' *** OUTPUT ***',/,/)
   98 FORMAT(1X,'     IERR = ',I4)
 5000 FORMAT(1X,4('(',F5.1,',',F5.1,')'))
 6000 FORMAT(1X,' RE(A) = ',E10.2,' IM(A) = ',E10.2,&
                ' RE(B) = ',E10.2,' IM(B) = ',E10.2,&
                 ' I,J = ',I2,3X,I2)
 6500 FORMAT(1X ,/, 2(14X,A11,8X))
 7000 FORMAT(1X,2(12X,1PD14.7,7X))
 8000 FORMAT(1X, 2(5X,F12.8,' ,',F12.8,2X))
      END
      SUBROUTINE BWINTA(AR, AI, BR, BI, LN, N, SCL, IERR)
      INTEGER           LN, N
      REAL(8)  AR(LN, N), BR(LN, N), SCL(N),&
                        AI(LN, N), BI(LN, N)
!
      INTEGER I,J
```

```
      REAL(8) F,ZERO,ONE
      PARAMETER(ZERO=0.D0)
      PARAMETER(ONE =1.D0)
!
      IERR=0
      IF(N.GT.LN) IERR=3000
      IF(N.LT. 1) IERR=3000
      IF(IERR.GT.0) RETURN
      DO 1000 I=1,N
         IF(BR(I,I).LE.ZERO) THEN
              IERR=4000
              RETURN
         ENDIF
         SCL(I)=ONE/SQRT(BR(I,I))
 1000 CONTINUE
      DO 2000 J=1,N
      DO 2001 I=1,N
         F=SCL(I)*SCL(J)
         AR(I,J)=AR(I,J)*F
         AI(I,J)=AI(I,J)*F
         BR(I,J)=BR(I,J)*F
         BI(I,J)=BI(I,J)*F
 2001 CONTINUE
 2000 CONTINUE
!
      RETURN
      END
!
      SUBROUTINE BWINTB(AR, AI, LN, N, SCL, IERR)
      INTEGER           LN, N
      REAL(8)  AR(LN, N), AI(LN, N), SCL(N)
      INTEGER I,J
!
      IERR=0
      IF(N.GT.LN) IERR=3000
      IF(N.LT. 1) IERR=3000
      IF(IERR.GT.0) RETURN
!
      DO 1000 I=1,N
      DO 1001 J=1,N
         AR(I,J)=AR(I,J)*SCL(I)
         AI(I,J)=AI(I,J)*SCL(I)
 1001 CONTINUE
 1000 CONTINUE
!
      RETURN
      END
```

(d) 出力結果

```
 *** ZCGRAA ***

 *** INPUT ***

    N=   4

     INPUT MATRIX A ( REAL , IMAGINARY )

(  8.0,  0.0)(  3.0,  0.0)(  1.0, -2.0)( -1.0, -2.0)
(  3.0,  0.0)(  9.0,  0.0)(  1.0,  2.0)( -1.0,  2.0)
(  1.0,  2.0)(  1.0, -2.0)( 10.0,  0.0)( -3.0,  0.0)
( -1.0,  2.0)( -1.0, -2.0)( -3.0,  0.0)( 11.0,  0.0)

    INPUT MATRIX B ( REAL , IMAGINARY )

(  8.0,  0.0)(  3.0,  0.0)(  1.0,  2.0)( -1.0,  2.0)
(  3.0,  0.0)(  9.0,  0.0)(  1.0, -2.0)( -1.0, -2.0)
(  1.0, -2.0)(  1.0,  2.0)( 10.0,  0.0)( -3.0,  0.0)
( -1.0, -2.0)( -1.0,  2.0)( -3.0,  0.0)( 11.0,  0.0)

*** OUTPUT ***

     IERR =    0

             EIGENVALUE                       EIGENVALUE
             2.3087618D-01                    1.0000000D+00

             EIGENVECTOR                      EIGENVECTOR
       0.17507548 ,  0.00103858         0.20825807 ,  0.00000000
      -0.16011650 ,  0.00086549         0.20825807 ,  0.00000000
      -0.00110807 , -0.14886363         0.00144123 , -0.00000000
       0.00110807 , -0.13795850        -0.00144123 , -0.00000000

             EIGENVALUE                       EIGENVALUE
             1.0000000D+00                    4.3313260D+00

             EIGENVECTOR                      EIGENVECTOR
       0.00314584 , -0.03530977         0.36436426 , -0.00216148
       0.00314584 , -0.03530977        -0.33323187 , -0.00180124
      -0.01730214 ,  0.19420373        -0.00230610 ,  0.30981257
       0.01730214 , -0.19420373         0.00230610 ,  0.28711700
```

4.17.2 ZCGRAN, CCGRAN
エルミート行列 (一般化固有値問題 $Az = \lambda Bz$, B: 正定値) の全固有値

(1) 機 能

エルミート行列 (2 次元配列型) (上三角型) (実数引数型) の一般化固有値問題

$$Az = \lambda Bz \ (A: \text{ エルミート行列 }, \ B: \text{ 正定値エルミート行列})$$

をコレスキー分解, ハウスホルダー法, QR 法で解いて, 全固有値 λ を求める.

(2) 使用法

倍精度サブルーチン:

CALL ZCGRAN (AR, AI, LNA, N, BR, BI, LNB, E, WORK, IERR)

単精度サブルーチン:

CALL CCGRAN (AR, AI, LNA, N, BR, BI, LNB, E, WORK, IERR)

(3) 引 数

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: { 32 ビット整数版では INTEGER(4)
64 ビット整数版では INTEGER(8) }

項番	引数名	型	大きさ	入出力	内　容
1	AR	{D R}	LNA, N	入 力	エルミート行列 A の実部
				出 力	入力時の内容は保存されない
2	AI	{D R}	LNA, N	入 力	エルミート行列 A の虚部
				出 力	入力時の内容は保存されない
3	LNA	I	1	入 力	配列 AR,AI の整合寸法
4	N	I	1	入 力	行列 A, B の次数
5	BR	{D R}	LNB, N	入 力	エルミート行列 B の実部
				出 力	入力時の内容は保存されない
6	BI	{D R}	LNB, N	入 力	エルミート行列 B の虚部
				出 力	入力時の内容は保存されない
7	LNB	I	1	入 力	配列 BR,BI の整合寸法
8	E	{D R}	N	出 力	固有値 λ
9	WORK	{D R}	4 × N	ワーク	作業領域
10	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a) $1 \leq \mathrm{N} \leq \mathrm{LNA}, \mathrm{LNB}$

(5) **エラーインディケータ**

IERR の値	意　　　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	$\mathrm{E}(1) \leftarrow \dfrac{\mathrm{AR}(1,1)}{\mathrm{BR}(1,1)}$ とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
4000	B が正定値でなかった.	
5000	標準の固有値問題の解法過程において, 処理が続行不能となった.	

(6) **注意事項**

(a) 配列 AR,AI,BR, BI には, 上三角部分のみにデータが格納されていればよい.

(b) 固有値は小さい順に格納される.

(c) 固有ベクトルを必要とするときは, 4.17.1 $\left\{\begin{matrix}\text{ZCGRAA}\\\text{CCGRAA}\end{matrix}\right\}$ を使用する.

4.18 エルミート行列 (2 次元配列型) (上三角型) (複素引数型) の一般化固有値問題 ($Az = \lambda Bz$)

4.18.1 ZCGHAA, CCGHAA

エルミート行列 (一般化固有値問題 $Az = \lambda Bz$, B: 正定値) の全固有値・全固有ベクトル

(1) **機 能**

エルミート行列 (2 次元配列型) (上三角型) (複素引数型) の一般化固有値問題

$Az = \lambda Bz$ (A : エルミート行列 , B : 正定値エルミート行列)

をコレスキー分解, ハウスホルダー法, QR 法で解いて, 全固有値 λ とそれに対応する全固有ベクトル $\boldsymbol{z}$ を求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZCGHAA (A, LNA, N, B, LNB, E, WORK, ZZW, IERR)

単精度サブルーチン:

CALL CCGHAA (A, LNA, N, B, LNB, E, WORK, ZZW, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型

R:単精度実数型　C:単精度複素数型

I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$

項番	引数名	型	大きさ	入出力	内　容
1	A	Z C	LNA, N	入 力	エルミート行列 A
				出 力	固有ベクトル $\boldsymbol{z}$
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A, B の次数
4	B	Z C	LNB, N	入 力	エルミート行列 B
				出 力	入力時の内容は保存されない
5	LNB	I	1	入 力	配列 B の整合寸法
6	E	D R	N	出 力	固有値 λ
7	WORK	D R	$2 \times$ N	ワーク	作業領域
8	ZZW	Z C	N	ワーク	作業領域
9	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $1 \leq \text{N} \leq \text{LNA}, \text{LNB}$

(5) エラーインディケータ

IERR の値	意　　　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	$\mathrm{E(1)} \leftarrow \dfrac{\mathrm{A(1,1)}}{\mathrm{B(1,1)}}$, $\mathrm{A(1,1)} \leftarrow \dfrac{1.0}{\sqrt{\mathrm{B(1,1)}}}$ とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
4000	B が正定値でなかった.	
5000	標準の固有値問題の解法過程において, 処理が続行不能となった.	

(6) 注意事項

(a) 配列 A, B には, 上三角部分のみにデータが格納されていればよい.

(b) 固有値は小さい順に格納される.

(c) 各固有ベクトル $\boldsymbol{v}_i$ は $\boldsymbol{v}_j^* B \boldsymbol{v}_k = \delta_{j,k}$ となるように正規直交化される.

(d) 固有ベクトルを必要としないときは, 4.18.2 $\left\{\begin{matrix}\text{ZCGHAN}\\ \text{CCGHAN}\end{matrix}\right\}$ を使用する.

(7) 使用例

(a) 問 題

4 次エルミート行列

$$A = \begin{bmatrix} 8 & 3 & 1-2i & -1-2i \\ 3 & 9 & 1+2i & -1+2i \\ 1+2i & 1-2i & 10 & -3 \\ -1+2i & -1-2i & -3 & 11 \end{bmatrix}$$

と, 各成分が共役である 4 次エルミート行列

$$B = \begin{bmatrix} 8 & 3 & 1+2i & -1+2i \\ 3 & 9 & 1-2i & -1-2i \\ 1-2i & 1+2i & 10 & -3 \\ -1-2i & -1+2i & -3 & 11 \end{bmatrix}$$

に対する一般化固有値問題を解いて, 固有ベクトルも求める.

(b) 入力データ

行列 A, LNA=4, N=4, 行列 B, LNB=4

(c) 主プログラム

```
      PROGRAM ACGHAA
      IMPLICIT REAL(8)(A-H,O-Z)
      PARAMETER ( N=4 , LN=4 )
      REAL(8)       E(N), WORK(2*N)
      COMPLEX(8)  KEEP(LN,N),VM,A(LN,N),B(LN,N),ZZW(N)
      REAL(8)       AR(LN,N),BR(LN,N),AI(LN,N),BI(LN,N),SCL(N)
      KEEP(1,1)=( 8.D0, 0.D0)
      KEEP(2,2)=( 9.D0, 0.D0)
      KEEP(3,3)=( 10.D0, 0.D0)
      KEEP(4,4)=( 11.D0, 0.D0)
      KEEP(1,2)=( 3.D0, 0.D0)
      KEEP(1,3)=( 1.D0, 2.D0)
      KEEP(1,4)=(-1.D0, 2.D0)
```

```
      KEEP(2,3)=( 1.D0,-2.D0)
      KEEP(2,4)=(-1.D0,-2.D0)
      KEEP(3,4)=(-3.D0, 0.D0)
      METHOD=0
      WRITE(6,60)
      IF(METHOD.EQ.0) WRITE(6,70) N
      DO 1000 I=1,N
      DO 1001 J=I,N
        BR(I,J)=DBLE (KEEP(I,J))
        BI(I,J)=DIMAG(KEEP(I,J))
        KEEP(J,I)=CONJG(KEEP(I,J))
        AR(I,J)=BR(I,J)
        AI(I,J)=-BI(I,J)
        AR(J,I)=AR(I,J)
        AI(J,I)=-AI(I,J)
        BR(J,I)=BR(I,J)
        BI(J,I)=-BI(I,J)
 1001 CONTINUE
 1000 CONTINUE
      IF(METHOD.EQ.0) THEN
         WRITE(6,80)
         DO 1150 I=1,N
         WRITE(6,5000) AR(I,1),AI(I,1),AR(I,2),AI(I,2),&
                       AR(I,3),AI(I,3),AR(I,4),AI(I,4)
 1150    CONTINUE
         WRITE(6,90)
         DO 1250 I=1,N
         WRITE(6,5000) BR(I,1),BI(I,1),BR(I,2),BI(I,2),&
                       BR(I,3),BI(I,3),BR(I,4),BI(I,4)
 1250    CONTINUE
      ENDIF
      IF(METHOD.GT.0) THEN
         CALL BWINTA(AR, AI, BR, BI, LN, N, SCL, IERRA)
         IF(IERRA.GT.0) STOP
         WRITE(6,95)
         DO 1100 I=1,N
         DO 1101 J=I,N
            WRITE(6,6000) AR(I,J),AI(I,J),BR(I,J),BI(I,J),I,J
 1101    CONTINUE
 1100    CONTINUE
      ENDIF
      DO 1200  I=1,N
      DO 1201  J=I,N
         A(I,J)=CMPLX(AR(I,J),AI(I,J), KIND=8)
         B(I,J)=CMPLX(BR(I,J),BI(I,J), KIND=8)
 1201 CONTINUE
 1200 CONTINUE
!
      CALL ZCGHAA(A, LN, N, B, LN, E, WORK, ZZW, IERR)
!
      DO 1300  I=1,N
      DO 1301  J=1,N
         AR(I,J)=A(I,J)
         AI(I,J)=DIMAG(A(I,J))
 1301 CONTINUE
 1300 CONTINUE
      IF(METHOD.GT.0) THEN
         CALL BWINTB(AR, AI, LN, N, SCL, IERRB)
         IF(IERRB.GT.0) STOP
      ENDIF
      IF(METHOD.GT.0) THEN
         WRITE(6,96)
      ELSE
         WRITE(6,97)
      ENDIF
      WRITE(6,98) IERR
      DO 2000 I=1,N,2
         VM=(0.D0,0.D0)
         DO 2100 J=1,N
         DO 2101 K=1,N
            VM=VM + KEEP(J,K)*CMPLX(AR(K,I), AI(K,I), KIND=8)&
                             *CMPLX(AR(J,I),-AI(J,I), KIND=8)
 2101    CONTINUE
 2100    CONTINUE
         WRITE(6,6500) ('EIGENVALUE ',L=1,2)
         WRITE(6,7000) E(I),E(I+1)
         WRITE(6,6500) ('EIGENVECTOR',L=1,2)
         WRITE(6,8000) AR(1,I),AI(1,I),AR(1,I+1),AI(1,I+1)
         WRITE(6,8000) AR(2,I),AI(2,I),AR(2,I+1),AI(2,I+1)
         WRITE(6,8000) AR(3,I),AI(3,I),AR(3,I+1),AI(3,I+1)
         WRITE(6,8000) AR(4,I),AI(4,I),AR(4,I+1),AI(4,I+1)
 2000 CONTINUE
      STOP
   60 FORMAT(1X,  ' *** ZCGHAA *** ',/,/)
   70 FORMAT(1X, ' *** INPUT ***',/,/,'     N= ',I3,/,/)
   80 FORMAT(1X,'     INPUT MATRIX A ( REAL , IMAGINARY )',/)
   90 FORMAT(1X,/,/,'     INPUT MATRIX B ( REAL , IMAGINARY )',/)
   95 FORMAT(1X,' *** INPUT *** (SCALED) ')
   96 FORMAT(1X,' *** OUTPUT *** (SCALED) ')
   97 FORMAT(1X,/,/, ' *** OUTPUT ***',/,/)
   98 FORMAT(1X,'     IERR = ',I4)
 5000 FORMAT(1X,4('(',F5.1,',',F5.1,')'))
 6000 FORMAT(1X,' RE(A) = ',E10.2,' IM(A) = ',E10.2,&
                ' RE(B) = ',E10.2,' IM(B) = ',E10.2,&
                 ' I,J = ',I2,3X,I2)
 6500 FORMAT(1X ,/, 2(14X,A11,8X))
 7000 FORMAT(1X,2(12X,1PD14.7,7X))
 8000 FORMAT(1X, 2(5X,F12.8,' ,',F12.8,2X))
      END
      SUBROUTINE BWINTA(AR, AI, BR, BI, LN, N, SCL, IERR)
      INTEGER           LN, N
```

```
      REAL(8)  AR(LN, N), BR(LN, N), SCL(N),&
                        AI(LN, N), BI(LN, N)
!
      INTEGER I,J
      REAL(8) F,ZERO,ONE
      PARAMETER(ZERO=0.D0)
      PARAMETER(ONE =1.D0)
!
      IERR=0
      IF(N.GT.LN) IERR=3000
      IF(N.LT. 1) IERR=3000
      IF(IERR.GT.0) RETURN
      DO 1000 I=1,N
         IF(BR(I,I).LE.ZERO) THEN
              IERR=4000
              RETURN
         ENDIF
         SCL(I)=ONE/SQRT(BR(I,I))
 1000 CONTINUE
      DO 2000 J=1,N
      DO 2001 I=1,N
         F=SCL(I)*SCL(J)
         AR(I,J)=AR(I,J)*F
         AI(I,J)=AI(I,J)*F
         BR(I,J)=BR(I,J)*F
         BI(I,J)=BI(I,J)*F
 2001 CONTINUE
 2000 CONTINUE
!
      RETURN
      END
!
      SUBROUTINE BWINTB(AR, AI, LN, N, SCL, IERR)
      INTEGER           LN, N
      REAL(8)  AR(LN, N), AI(LN, N), SCL(N)
      INTEGER I,J
!
      IERR=0
      IF(N.GT.LN) IERR=3000
      IF(N.LT. 1) IERR=3000
      IF(IERR.GT.0) RETURN
!
      DO 1000 I=1,N
      DO 1001 J=1,N
         AR(I,J)=AR(I,J)*SCL(I)
         AI(I,J)=AI(I,J)*SCL(I)
 1001 CONTINUE
 1000 CONTINUE
!
      RETURN
      END
```

(d) 出力結果

```
 *** ZCGHAA ***

 *** INPUT ***

    N=   4

     INPUT MATRIX A ( REAL , IMAGINARY )

(  8.0,  0.0)(  3.0,  0.0)(  1.0, -2.0)( -1.0, -2.0)
(  3.0,  0.0)(  9.0,  0.0)(  1.0,  2.0)( -1.0,  2.0)
(  1.0,  2.0)(  1.0, -2.0)( 10.0,  0.0)( -3.0,  0.0)
( -1.0,  2.0)( -1.0, -2.0)( -3.0,  0.0)( 11.0,  0.0)

    INPUT MATRIX B ( REAL , IMAGINARY )

(  8.0,  0.0)(  3.0,  0.0)(  1.0,  2.0)( -1.0,  2.0)
(  3.0,  0.0)(  9.0,  0.0)(  1.0, -2.0)( -1.0, -2.0)
(  1.0, -2.0)(  1.0,  2.0)( 10.0,  0.0)( -3.0,  0.0)
( -1.0, -2.0)( -1.0,  2.0)( -3.0,  0.0)( 11.0,  0.0)

*** OUTPUT ***

     IERR =    0

             EIGENVALUE                       EIGENVALUE
            2.3087618D-01                     1.0000000D+00

             EIGENVECTOR                      EIGENVECTOR
       0.17507548 ,  0.00103858         0.20825807 ,  0.00000000
      -0.16011650 ,  0.00086549         0.20825807 ,  0.00000000
      -0.00110807 , -0.14886363         0.00144123 , -0.00000000
       0.00110807 , -0.13795850        -0.00144123 , -0.00000000
             EIGENVALUE                       EIGENVALUE
            1.0000000D+00                     4.3313260D+00

             EIGENVECTOR                      EIGENVECTOR
       0.00314584 , -0.03530977         0.36436426 , -0.00216148
       0.00314584 , -0.03530977        -0.33323187 , -0.00180124
      -0.01730214 ,  0.19420373        -0.00230610 ,  0.30981257
       0.01730214 , -0.19420373         0.00230610 ,  0.28711700
```

4.18.2 ZCGHAN, CCGHAN
エルミート行列 (一般化固有値問題 $Az = \lambda Bz$, B: 正定値) の全固有値

(1) **機 能**

エルミート行列 (2 次元配列型) (上三角型) (複素引数型) の一般化固有値問題

$$Az = \lambda Bz \ (A: \text{ エルミート行列 }, \ B: \text{ 正定値エルミート行列})$$

をコレスキー分解, ハウスホルダー法, QR 法で解いて, 全固有値 λ を求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZCGHAN (A, LNA, N, B, LNB, E, WORK, ZZW, IERR)

単精度サブルーチン:

CALL CCGHAN (A, LNA, N, B, LNB, E, WORK, ZZW, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\ \text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$

項番	引数名	型	大きさ	入出力	内　　　容
1	A	$\begin{Bmatrix}Z\\C\end{Bmatrix}$	LNA, N	入 力	エルミート行列 A
				出 力	入力時の内容は保存されない
2	LNA	I	1	入 力	配列 A の整合寸法
3	N	I	1	入 力	行列 A, B の次数
4	B	$\begin{Bmatrix}Z\\C\end{Bmatrix}$	LNB, N	入 力	エルミート行列 B
				出 力	入力時の内容は保存されない
5	LNB	I	1	入 力	配列 B の整合寸法
6	E	$\begin{Bmatrix}D\\R\end{Bmatrix}$	N	出 力	固有値 λ
7	WORK	$\begin{Bmatrix}D\\R\end{Bmatrix}$	2 × N	ワーク	作業領域
8	ZZW	$\begin{Bmatrix}Z\\C\end{Bmatrix}$	N	ワーク	作業領域
9	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $1 \leq \mathrm{N} \leq \mathrm{LNA}, \mathrm{LNB}$

(5) **エラーインディケータ**

IERR の値	意　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	$\mathrm{E}(1) \leftarrow \dfrac{\mathrm{A}(1,1)}{\mathrm{B}(1,1)}$ とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
4000	B が正定値でなかった.	
5000	標準の固有値問題の解法過程において, 処理が続行不能となった.	

(6) **注意事項**

(a) 配列 A, B には, 上三角部分のみにデータが格納されていればよい.

(b) 固有値は小さい順に格納される.

(c) 固有ベクトルを必要とするときは, 4.18.1 $\left\{\begin{array}{c}\mathrm{ZCGHAA}\\ \mathrm{CCGHAA}\end{array}\right\}$を使用する.

4.19 エルミート行列 (2 次元配列型) (上三角型) (実数引数型) の一般化固有値問題 ($ABz = \lambda z$)

4.19.1 ZCGJAA, CCGJAA

エルミート行列 (一般化固有値問題 $ABz = \lambda z$, B: 正定値) の全固有値・全固有ベクトル

(1) **機 能**

エルミート行列 (2 次元配列型) (上三角型) (実数引数型) の一般化固有値問題

$$AB\boldsymbol{z} = \lambda \boldsymbol{z} \ (A:\ \text{エルミート行列},\ B:\ \text{正定値エルミート行列})$$

をコレスキー分解, ハウスホルダー法, QR 法で解いて, 全固有値 λ とそれに対応する全固有ベクトル $\boldsymbol{z}$ を求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZCGJAA (AR, AI, LNA, N, BR, BI, LNB, E, WORK, IERR)

単精度サブルーチン:

CALL CCGJAA (AR, AI, LNA, N, BR, BI, LNB, E, WORK, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型
I: $\begin{Bmatrix}\text{32 ビット整数版では INTEGER(4)}\\ \text{64 ビット整数版では INTEGER(8)}\end{Bmatrix}$

項番	引数名	型	大きさ	入出力	内　　容
1	AR	$\begin{Bmatrix}\mathrm{D}\\ \mathrm{R}\end{Bmatrix}$	LNA, N	入 力	エルミート行列 A の実部
				出 力	固有ベクトル $\boldsymbol{z}$ の実部
2	AI	$\begin{Bmatrix}\mathrm{D}\\ \mathrm{R}\end{Bmatrix}$	LNA, N	入 力	エルミート行列 A の虚部
				出 力	固有ベクトル $\boldsymbol{z}$ の虚部
3	LNA	I	1	入 力	配列 AR, AI の整合寸法
4	N	I	1	入 力	行列 A, B の次数
5	BR	$\begin{Bmatrix}\mathrm{D}\\ \mathrm{R}\end{Bmatrix}$	LNB, N	入 力	エルミート行列 B の実部
				出 力	入力時の内容は保存されない
6	BI	$\begin{Bmatrix}\mathrm{D}\\ \mathrm{R}\end{Bmatrix}$	LNB, N	入 力	エルミート行列 B の虚部
				出 力	入力時の内容は保存されない
7	LNB	I	1	入 力	配列 BR, BI の整合寸法
8	E	$\begin{Bmatrix}\mathrm{D}\\ \mathrm{R}\end{Bmatrix}$	N	出 力	固有値 λ
9	WORK	$\begin{Bmatrix}\mathrm{D}\\ \mathrm{R}\end{Bmatrix}$	4 × N	ワーク	作業領域
10	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a) $1 \leq \mathrm{N} \leq \mathrm{LNA}, \mathrm{LNB}$

(5) エラーインディケータ

IERR の値	意　　　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	$\mathrm{E}(1) \leftarrow \mathrm{A}(1,1) \times \mathrm{B}(1,1)$, $\mathrm{A}(1,1) \leftarrow \dfrac{1.0}{\sqrt{\mathrm{B}(1,1)}}$ とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
4000	B が正定値でなかった.	
5000	標準の固有値問題の解法過程において, 処理が続行不能となった.	

(6) 注意事項

(a) 配列 AR, AI, BR, BI には, 上三角部分のみにデータが格納されていればよい.

(b) 固有値は小さい順に格納される.

(c) 各固有ベクトル $\boldsymbol{v}_i$ は $\boldsymbol{v}_j^* B \boldsymbol{v}_k = \delta_{j,k}$ となるように正規直交化される.

(d) 固有ベクトルを必要としないときは, 4.19.2 $\left\{\begin{matrix}\mathrm{ZCGJAN}\\ \mathrm{CCGJAN}\end{matrix}\right\}$ を使用する.

(e) 行列 A のみが正定値行列であるときは, 4.20.1 $\left\{\begin{matrix}\mathrm{ZCGKAA}\\ \mathrm{CCGKAA}\end{matrix}\right\}$ を使用する.

(7) 使用例

(a) 問 題

4 次エルミート行列

$$A = \begin{bmatrix} 8 & 3 & 1-2i & -1-2i \\ 3 & 9 & 1+2i & -1+2i \\ 1+2i & 1-2i & 10 & -3 \\ -1+2i & -1-2i & -3 & 11 \end{bmatrix}$$

と, 各成分が共役である 4 次エルミート行列

$$B = \begin{bmatrix} 8 & 3 & 1+2i & -1+2i \\ 3 & 9 & 1-2i & -1-2i \\ 1-2i & 1+2i & 10 & -3 \\ -1-2i & -1+2i & -3 & 11 \end{bmatrix}$$

A, B に対して, 非エルミート行列 AB の固有値と固有ベクトルを求める.

(b) 入力データ

N=4, LNA=LNB=4, エルミート行列 A, B

(c) 主プログラム

```
      PROGRAM ACGJAA
! *** EXAMPLE OF ZCGJAA ***
      IMPLICIT NONE
!
      INTEGER N,LNA,LNB
      PARAMETER( N = 4, LNA = 4, LNB = 4 )
      INTEGER IERR,I,J,L
      REAL(8) AR(LNA,N),BR(LNB,N),AI(LNA,N),BI(LNB,N)
      REAL(8) E(N),WORK(4*N)
      COMPLEX(8) KEEP(LNA,N)
!
      KEEP(1,1)=( 8.D0, 0.D0)
      KEEP(2,2)=( 9.D0, 0.D0)
      KEEP(3,3)=( 10.D0, 0.D0)
      KEEP(4,4)=( 11.D0, 0.D0)
      KEEP(1,2)=( 3.D0, 0.D0)
      KEEP(1,3)=( 1.D0, 2.D0)
      KEEP(1,4)=(-1.D0, 2.D0)
      KEEP(2,3)=( 1.D0,-2.D0)
      KEEP(2,4)=(-1.D0,-2.D0)
      KEEP(3,4)=(-3.D0, 0.D0)
      WRITE(6,6000) N, LNA, LNB
      DO 100 I=1,N
      DO 110 J=I,N
         BR(I,J)=DBLE (KEEP(I,J))
         BI(I,J)=DIMAG(KEEP(I,J))
         KEEP(J,I)=CONJG(KEEP(I,J))
         AR(I,J)=BR(I,J)
         AI(I,J)=-BI(I,J)
         AR(J,I)=AR(I,J)
         AI(J,I)=-AI(I,J)
         BR(J,I)=BR(I,J)
         BI(J,I)=-BI(I,J)
  110 CONTINUE
  100 CONTINUE
      WRITE(6,6010)
      DO 120 I=1,N
         WRITE(6,6020) AR(I,1),AI(I,1),AR(I,2),AI(I,2),&
                       AR(I,3),AI(I,3),AR(I,4),AI(I,4)
  120 CONTINUE
      WRITE(6,6030)
      DO 130 I=1,N
         WRITE(6,6020) BR(I,1),BI(I,1),BR(I,2),BI(I,2),&
                       BR(I,3),BI(I,3),BR(I,4),BI(I,4)
  130 CONTINUE
!
      CALL ZCGJAA(AR,AI, LNA, N, BR,BI, LNB, E, WORK, IERR)
!
      WRITE(6,6040) IERR
      DO 140 I=1,N,2
         WRITE(6,6050) (' EIGENVALUE',L=1,2)
         WRITE(6,6060) E(I),E(I+1)
         WRITE(6,6050) ('EIGENVECTOR',L=1,2)
         WRITE(6,6070) AR(1,I),AI(1,I),AR(1,I+1),AI(1,I+1)
         WRITE(6,6070) AR(2,I),AI(2,I),AR(2,I+1),AI(2,I+1)
         WRITE(6,6070) AR(3,I),AI(3,I),AR(3,I+1),AI(3,I+1)
         WRITE(6,6070) AR(4,I),AI(4,I),AR(4,I+1),AI(4,I+1)
  140 CONTINUE
!
      STOP
 6000 FORMAT(/,&
             1X,'***  ZCGJAA  ***',/,/,&
             1X,' **  INPUT  **',/,/,&
             1X,'     N = ',I4, '    LNA = ',I4, '   LNB = ',I4,/)
 6010 FORMAT(/,&
             1X,'     INPUT MATRIX A ( REAL , IMAGINARY )',/)
 6020 FORMAT(1X,5X,4('(',F5.1,',',F5.1,')'))
 6030 FORMAT(/,&
             1X,'     INPUT MATRIX B ( REAL , IMAGINARY )',/)
 6040 FORMAT(/,&
             1X,' **  OUTPUT  **',/,/,&
             1X,'     IERR =',I5,/)
 6050 FORMAT(/,&
             1X,14X,A11,22X,A11)
 6060 FORMAT(1X,12X,1PD14.7,19X,1PD14.7)
 6070 FORMAT(1X,5X,F12.8,' ,',F12.8,7X,F12.8,' ,',F12.8)
 6080 FORMAT(1X,' RE(A) = ',D10.2,' IM(A) = ',D10.2,&
                ' RE(B) = ',D10.2,' IM(B) = ',D10.2,&
                ' I,J = ',I2,3X,I2)
      END
```

(d) 出力結果

```
***  ZCGJAA  ***

 **  INPUT  **

     N =    4    LNA =    4   LNB =    4


     INPUT MATRIX A ( REAL , IMAGINARY )

     (  8.0,  0.0)(  3.0,  0.0)(  1.0, -2.0)( -1.0, -2.0)
     (  3.0,  0.0)(  9.0,  0.0)(  1.0,  2.0)( -1.0,  2.0)
     (  1.0,  2.0)(  1.0, -2.0)( 10.0,  0.0)( -3.0,  0.0)
     ( -1.0,  2.0)( -1.0, -2.0)( -3.0,  0.0)( 11.0,  0.0)
```

```
    INPUT MATRIX B ( REAL , IMAGINARY )

    (  8.0,  0.0)(  3.0,  0.0)(  1.0,  2.0)( -1.0,  2.0)
    (  3.0,  0.0)(  9.0,  0.0)(  1.0, -2.0)( -1.0, -2.0)
    (  1.0, -2.0)(  1.0,  2.0)( 10.0,  0.0)( -3.0,  0.0)
    ( -1.0, -2.0)( -1.0,  2.0)( -3.0,  0.0)( 11.0,  0.0)

**  OUTPUT  **

    IERR =    0

              EIGENVALUE                       EIGENVALUE
            1.6653903D+01                    3.6956492D+01

             EIGENVECTOR                      EIGENVECTOR
     -0.39854844 , -0.00036439         0.10790304 , -0.01173201
      0.34532368 ,  0.00125155        -0.10272938 , -0.00491675
      0.00738031 , -0.13233200         0.01670018 ,  0.32687634
     -0.00409275 , -0.12491736         0.01489428 ,  0.28145894

              EIGENVALUE                       EIGENVALUE
            1.0637116D+02                    2.1801844D+02

             EIGENVECTOR                      EIGENVECTOR
      0.17017623 , -0.00775370         0.09159450 ,  0.00390728
      0.20301495 , -0.00090493         0.10133798 ,  0.00028167
     -0.09985610 , -0.00146802         0.14661381 ,  0.00223902
      0.12968534 , -0.00655395        -0.16597958 , -0.00438924
```

4.19.2 ZCGJAN, CCGJAN
エルミート行列 (一般化固有値問題 $ABz = \lambda z$, B: 正定値) の全固有値

(1) **機 能**

エルミート行列 (2 次元配列型) (上三角型) (実数引数型) の一般化固有値問題

$$AB\boldsymbol{z} = \lambda \boldsymbol{z} \ (A: \text{エルミート行列}, \ B: \text{正定値エルミート行列})$$

をコレスキー分解, ハウスホルダー法, QR 法で解いて, 全固有値 λ を求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZCGJAN (AR, AI, LNA, N, BR, BI, LNB, E, WORK, IERR)

単精度サブルーチン:

CALL CCGJAN (AR, AI, LNA, N, BR, BI, LNB, E, WORK, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\begin{Bmatrix}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{Bmatrix}$

項番	引数名	型	大きさ	入出力	内　容
1	AR	$\begin{Bmatrix}D\\R\end{Bmatrix}$	LNA, N	入 力	エルミート行列 A の実部
				出 力	入力時の内容は保存されない
2	AI	$\begin{Bmatrix}D\\R\end{Bmatrix}$	LNA, N	入 力	エルミート行列 A の虚部
				出 力	入力時の内容は保存されない
3	LNA	I	1	入 力	配列 AR, AI の整合寸法
4	N	I	1	入 力	行列 A,B の次数
5	BR	$\begin{Bmatrix}D\\R\end{Bmatrix}$	LNB, N	入 力	エルミート行列 B の実部
				出 力	入力時の内容は保存されない
6	BI	$\begin{Bmatrix}D\\R\end{Bmatrix}$	LNB, N	入 力	エルミート行列 B の虚部
				出 力	入力時の内容は保存されない
7	LNB	I	1	入 力	配列 BR, BI の整合寸法
8	E	$\begin{Bmatrix}D\\R\end{Bmatrix}$	N	出 力	固有値 λ
9	WORK	$\begin{Bmatrix}D\\R\end{Bmatrix}$	4 × N	ワーク	作業領域
10	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $1 \leq \mathrm{N} \leq \mathrm{LNA}, \mathrm{LNB}$

(5) **エラーインディケータ**

IERR の値	意　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	E(1) ← A(1, 1) × B(1, 1) とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
4000	B が正定値でなかった.	
5000	標準の固有値問題の解法過程において, 処理が続行不能となった.	

(6) **注意事項**

(a) 配列 AR, AI, BR, BI には, 上三角部分のみにデータが格納されていればよい.

(b) 固有値は小さい順に格納される.

(c) 固有ベクトルを必要とするときは, 4.19.1 $\left\{\begin{matrix}\text{ZCGJAA}\\\text{CCGJAA}\end{matrix}\right\}$ を使用する.

(d) 行列 A のみが正定値行列であるときは, 4.20.2 $\left\{\begin{matrix}\text{ZCGKAN}\\\text{CCGKAN}\end{matrix}\right\}$ を使用する.

4.20 エルミート行列 (2 次元配列型) (上三角型) (実数引数型) の一般化固有値問題 ($BAz = \lambda z$)

4.20.1 ZCGKAA, CCGKAA

エルミート行列 (一般化固有値問題 $BAz = \lambda z$, B: 正定値) の全固有値・全固有ベクトル

(1) **機 能**

エルミート行列 (2 次元配列型) (上三角型) (実数引数型) の一般化固有値問題

$$BA\boldsymbol{z} = \lambda\boldsymbol{z} \ (A: \text{エルミート行列}\ ,\ B: \text{正定値エルミート行列})$$

をコレスキー分解, ハウスホルダー法, QR 法で解いて, 全固有値 λ とそれに対応する全固有ベクトル $\boldsymbol{z}$ を求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZCGKAA (AR, AI, LNA, N, BR, BI, LNB, E, WORK, IERR)

単精度サブルーチン:

CALL CCGKAA (AR, AI, LNA, N, BR, BI, LNB, E, WORK, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型　I: {32 ビット整数版では INTEGER(4) / 64 ビット整数版では INTEGER(8)}
R:単精度実数型　C:単精度複素数型

項番	引数名	型	大きさ	入出力	内　容
1	AR	D R	LNA, N	入 力	エルミート行列 A の実部
				出 力	固有ベクトル $\boldsymbol{z}$ の実部
2	AI	D R	LNA, N	入 力	エルミート行列 A の虚部
				出 力	固有ベクトル $\boldsymbol{z}$ の虚部
3	LNA	I	1	入 力	配列 AR, AI の整合寸法
4	N	I	1	入 力	行列 A,B の次数
5	BR	D R	LNB, N	入 力	エルミート行列 B の実部
				出 力	入力時の内容は保存されない
6	BI	D R	LNB, N	入 力	エルミート行列 B の虚部
				出 力	入力時の内容は保存されない
7	LNB	I	1	入 力	配列 BR, BI の整合寸法
8	E	D R	N	出 力	固有値 λ
9	WORK	D R	4 × N	ワーク	作業領域
10	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a) $1 \leq \mathrm{N} \leq \mathrm{LNA}, \mathrm{LNB}$

(5) エラーインディケータ

IERR の値	意　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	$\mathrm{E}(1) \leftarrow \mathrm{A}(1,1) \times \mathrm{B}(1,1)$, $\mathrm{A}(1,1) \leftarrow \sqrt{\mathrm{B}(1,1)}$ とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
4000	B が正定値でなかった.	
5000	標準の固有値問題の解法過程において, 処理が続行不能となった.	

(6) 注意事項

(a) 配列 AR, AI, BR, BI には, 上三角部分のみにデータが格納されていればよい.

(b) 固有値は小さい順に格納される.

(c) 各固有ベクトル $\boldsymbol{v}_i$ は $\boldsymbol{v}_j^* B^{-1} \boldsymbol{v}_k = \delta_{j,k}$ となるように正規直交化される.

(d) 固有ベクトルを必要としないときは, 4.20.2 $\left\{\begin{matrix}\mathrm{ZCGKAN} \\ \mathrm{CCGKAN}\end{matrix}\right\}$ を使用する.

(e) 行列 A のみが正定値行列であるときは, 4.19.1 $\left\{\begin{matrix}\mathrm{ZCGJAA} \\ \mathrm{CCGJAA}\end{matrix}\right\}$ を使用する.

(7) 使用例

(a) 問 題

4 次エルミート行列

$$A = \begin{bmatrix} 8 & 3 & 1-2i & -1-2i \\ 3 & 9 & 1+2i & -1+2i \\ 1+2i & 1-2i & 10 & -3 \\ -1+2i & -1-2i & -3 & 11 \end{bmatrix}$$

と, 各成分が共役である 4 次エルミート行列

$$B = \begin{bmatrix} 8 & 3 & 1+2i & -1+2i \\ 3 & 9 & 1-2i & -1-2i \\ 1-2i & 1+2i & 10 & -3 \\ -1-2i & -1+2i & -3 & 11 \end{bmatrix}$$

A, B に対して, 非エルミート行列 BA の固有値と固有ベクトルを求める.

(b) 入力データ

N=4, LNA=LNB=4, エルミート行列 A, B

(c) 主プログラム

```
      PROGRAM ACGKAA
! *** EXAMPLE OF ZCGKAA ***
      IMPLICIT NONE
!
      INTEGER N,LNA,LNB
      PARAMETER( N=4 , LNA=4 , LNB=4 )
      INTEGER IERR,I,J,L
      REAL(8) AR(LNA,N),BR(LNB,N),AI(LNA,N),BI(LNB,N)
      REAL(8) E(N),WORK(4*N)
      COMPLEX(8)  KEEP(LNA,N)
!
      KEEP(1,1)=( 8.D0, 0.D0)
      KEEP(2,2)=( 9.D0, 0.D0)
      KEEP(3,3)=( 10.D0, 0.D0)
      KEEP(4,4)=( 11.D0, 0.D0)
      KEEP(1,2)=( 3.D0, 0.D0)
      KEEP(1,3)=( 1.D0, 2.D0)
      KEEP(1,4)=(-1.D0, 2.D0)
      KEEP(2,3)=( 1.D0,-2.D0)
      KEEP(2,4)=(-1.D0,-2.D0)
      KEEP(3,4)=(-3.D0, 0.D0)
      WRITE(6,6000) N, LNA, LNB
      DO 100 I=1,N
      DO 110 J=I,N
         BR(I,J)=DBLE (KEEP(I,J))
         BI(I,J)=DIMAG(KEEP(I,J))
         KEEP(J,I)=CONJG(KEEP(I,J))
         AR(I,J)=BR(I,J)
         AI(I,J)=-BI(I,J)
         AR(J,I)=AR(I,J)
         AI(J,I)=-AI(I,J)
         BR(J,I)=BR(I,J)
         BI(J,I)=-BI(I,J)
  110 CONTINUE
  100 CONTINUE
      WRITE(6,6010)
      DO 120 I=1,N
         WRITE(6,6020) AR(I,1),AI(I,1),AR(I,2),AI(I,2),&
                       AR(I,3),AI(I,3),AR(I,4),AI(I,4)
  120 CONTINUE
      WRITE(6,6030)
      DO 130 I=1,N
         WRITE(6,6020) BR(I,1),BI(I,1),BR(I,2),BI(I,2),&
                       BR(I,3),BI(I,3),BR(I,4),BI(I,4)
  130 CONTINUE
!
      CALL ZCGKAA(AR,AI, LNA, N, BR,BI, LNB, E, WORK, IERR)
!
      WRITE(6,6040) IERR
      DO 140 I=1,N,2
         WRITE(6,6050) (’ EIGENVALUE’,L=1,2)
         WRITE(6,6060) E(I),E(I+1)
         WRITE(6,6050) (’EIGENVECTOR’,L=1,2)
         WRITE(6,6070) AR(1,I),AI(1,I),AR(1,I+1),AI(1,I+1)
         WRITE(6,6070) AR(2,I),AI(2,I),AR(2,I+1),AI(2,I+1)
         WRITE(6,6070) AR(3,I),AI(3,I),AR(3,I+1),AI(3,I+1)
         WRITE(6,6070) AR(4,I),AI(4,I),AR(4,I+1),AI(4,I+1)
  140 CONTINUE
!
      STOP
 6000 FORMAT(/,&
             1X,’***  ZCGKAA  ***’,/,/,&
             1X,’ **  INPUT  **’,/,/,&
             1X,’     N = ’,I4, ’    LNA = ’,I4, ’   LNB = ’,I4,/)
 6010 FORMAT(/,&
             1X,’     INPUT MATRIX A ( REAL , IMAGINARY )’,/)
 6020 FORMAT(1X,5X,4(’(’,F5.1,’,’,F5.1,’)’))
 6030 FORMAT(/,&
             1X,’     INPUT MATRIX B ( REAL , IMAGINARY )’,/)
 6040 FORMAT(/,&
             1X,’ **  OUTPUT  **’,/,/,&
             1X,’     IERR =’,I5,/)
 6050 FORMAT(/,&
             1X,14X,A11,22X,A11)
 6060 FORMAT(1X,12X,1PD14.7,19X,1PD14.7)
 6070 FORMAT(1X,5X,F12.8,’ ,’,F12.8,7X,F12.8,’ ,’,F12.8)
 6080 FORMAT(1X,’ RE(A) = ’,D10.2,’ IM(A) = ’,D10.2,&
                ’ RE(B) = ’,D10.2,’ IM(B) = ’,D10.2,&
                ’ I,J = ’,I2,3X,I2)
      END
```

(d) 出力結果

```
***  ZCGKAA  ***

 **  INPUT  **

     N =    4    LNA =    4   LNB =    4


     INPUT MATRIX A ( REAL , IMAGINARY )

     (  8.0,  0.0)(  3.0,  0.0)(  1.0, -2.0)( -1.0, -2.0)
     (  3.0,  0.0)(  9.0,  0.0)(  1.0,  2.0)( -1.0,  2.0)
     (  1.0,  2.0)(  1.0, -2.0)( 10.0,  0.0)( -3.0,  0.0)
     ( -1.0,  2.0)( -1.0, -2.0)( -3.0,  0.0)( 11.0,  0.0)
```

```
      INPUT MATRIX B ( REAL , IMAGINARY )

      (  8.0,  0.0)(  3.0,  0.0)(  1.0,  2.0)( -1.0,  2.0)
      (  3.0,  0.0)(  9.0,  0.0)(  1.0, -2.0)( -1.0, -2.0)
      (  1.0, -2.0)(  1.0,  2.0)( 10.0,  0.0)( -3.0,  0.0)
      ( -1.0, -2.0)( -1.0,  2.0)( -3.0,  0.0)( 11.0,  0.0)

 **  OUTPUT  **

      IERR =    0

                EIGENVALUE                        EIGENVALUE
              1.6653903D+01                     3.6956492D+01

               EIGENVECTOR                       EIGENVECTOR
       -1.62644471 ,  0.00000000         -0.65982846 , -0.00000000
        1.40924217 , -0.00381901          0.61762116 , -0.09721828
        0.02962467 ,  0.54006351          0.11386209 ,  1.98647301
       -0.01716825 ,  0.50976217          0.09493235 ,  1.71080318

                EIGENVALUE                        EIGENVALUE
              1.0637116D+02                     2.1801844D+02

               EIGENVECTOR                       EIGENVECTOR
        1.75695717 ,  0.00000000          1.35366373 ,  0.00000000
        2.09207784 , -0.08597803          1.49511825 ,  0.05961668
       -1.02812336 ,  0.06200047          2.16426058 ,  0.05923384
        1.33921828 ,  0.00664669         -2.45129806 , -0.03970066
```

4.20.2 ZCGKAN, CCGKAN
エルミート行列 (一般化固有値問題 $BAz = \lambda z$, B: 正定値) の全固有値

(1) **機 能**

エルミート行列 (2 次元配列型) (上三角型) (実数引数型) の一般化固有値問題

$$BA\boldsymbol{z} = \lambda \boldsymbol{z} \ (A: \text{ エルミート行列 },\ B: \text{ 正定値エルミート行列})$$

をコレスキー分解, ハウスホルダー法, QR 法で解いて, 全固有値 λ を求める.

(2) **使用法**

倍精度サブルーチン:

CALL ZCGKAN (AR, AI, LNA, N, BR, BI, LNB, E, WORK, IERR)

単精度サブルーチン:

CALL CCGKAN (AR, AI, LNA, N, BR, BI, LNB, E, WORK, IERR)

(3) **引 数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\left\{\begin{array}{l}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{array}\right\}$

項番	引数名	型	大きさ	入出力	内　容
1	AR	$\left\{\begin{array}{c}\text{D}\\\text{R}\end{array}\right\}$	LNA, N	入 力	エルミート行列 A の実部
				出 力	入力時の内容は保存されない
2	AI	$\left\{\begin{array}{c}\text{D}\\\text{R}\end{array}\right\}$	LNA, N	入 力	エルミート行列 A の虚部
				出 力	入力時の内容は保存されない
3	LNA	I	1	入 力	配列 AR, AI の整合寸法
4	N	I	1	入 力	行列 A,B の次数
5	BR	$\left\{\begin{array}{c}\text{D}\\\text{R}\end{array}\right\}$	LNB, N	入 力	エルミート行列 B の実部
				出 力	入力時の内容は保存されない
6	BI	$\left\{\begin{array}{c}\text{D}\\\text{R}\end{array}\right\}$	LNB, N	入 力	エルミート行列 B の虚部
				出 力	入力時の内容は保存されない
7	LNB	I	1	入 力	配列 BR, BI の整合寸法
8	E	$\left\{\begin{array}{c}\text{D}\\\text{R}\end{array}\right\}$	N	出 力	固有値 λ
9	WORK	$\left\{\begin{array}{c}\text{D}\\\text{R}\end{array}\right\}$	$4 \times \text{N}$	ワーク	作業領域
10	IERR	I	1	出 力	エラーインディケータ

(4) **制限条件**

(a) $1 \leq \text{N} \leq \text{LNA}, \text{LNB}$

(5) **エラーインディケータ**

IERR の値	意　　　　味	処　理　内　容
0	正常終了.	
1000	N = 1 であった.	E(1) ← A(1, 1) × B(1, 1) とする.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
4000	B が正定値でなかった.	
5000	標準の固有値問題の解法過程において, 処理が続行不能となった.	

(6) **注意事項**

(a) 配列 AR, AI, BR, BI には, 上三角部分のみにデータが格納されていればよい.

(b) 固有値は小さい順に格納される.

(c) 固有ベクトルを必要とするときは, 4.20.1 $\left\{\begin{matrix}\text{ZCGKAA}\\\text{CCGKAA}\end{matrix}\right\}$を使用する.

(d) 行列 A のみが正定値行列であるときは, 4.19.2 $\left\{\begin{matrix}\text{ZCGJAN}\\\text{CCGJAN}\end{matrix}\right\}$を使用する.

4.21 実対称バンド行列 (対称バンド型) の一般化固有値問題

4.21.1 DCGBFF, RCGBFF

実対称バンド行列 (一般化固有値問題) の固有値・固有ベクトル

(1) **機能**

実対称バンド行列 (対称バンド型) の一般化固有値問題 $A\boldsymbol{x} = \lambda B\boldsymbol{x}$ (A:実対称バンド行列, B:正値対称バンド行列) の固有値とそれに対応する固有ベクトルをサブスペース法により, 固有値の絶対値が小さい方から m 個, または大きい方から m 個求める.

(2) **使用法**

倍精度サブルーチン:

CALL DCGBFF (A, LMA, N, MAB, B, LMB, MBB, M, ITOL, NITE, E, VE, LNV, MST,
IS1, IS2, W1, IW1, IERR)

単精度サブルーチン:

CALL RCGBFF (A, LMA, N, MAB, B, LMB, MBB, M, ITOL, NITE, E, VE, LNV, MST,
IS1, IS2, W1, IW1, IERR)

(3) **引数**

D:倍精度実数型　Z:倍精度複素数型
R:単精度実数型　C:単精度複素数型

I: $\left\{\begin{matrix}\text{32 ビット整数版では INTEGER(4)}\\\text{64 ビット整数版では INTEGER(8)}\end{matrix}\right\}$

項番	引数名	型	大きさ	入出力	内容
1	A	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	LMA, N	入力	実対称バンド行列 A (対称バンド型) (付録 B 参照)
				出力	入力時の内容は保存されない
2	LMA	I	1	入力	配列 A の整合寸法
3	N	I	1	入力	行列 A, B の次数
4	MAB	I	1	入力	行列 A のバンド幅
5	B	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	LMB, N	入力	正値対称バンド行列 B (対称バンド型)
6	LMB	I	1	入力	配列 B の整合寸法
7	MBB	I	1	入力	行列 B のバンド幅
8	M	I	1	入力	求めたい固有値の個数 m
9	ITOL	I	1	入力	収束判定用トレランス (注意事項 (b) 参照)
10	NITE	I	1	入力	最大反復回数 (注意事項 (d) 参照)
11	E	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	内容参照	出力	固有値 **大きさ:** $(\min(2 \times m, \mathrm{N}, m + 8))$
12	VE	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	内容参照	出力	各固有値に対応する固有ベクトル (列ベクトル) **大きさ:** $((\mathrm{LNV}, \min(2 \times m, \mathrm{N}, m + 8)))$
13	LNV	I	1	入力	配列 VE の整合寸法
14	MST	I	1	出力	計算されなかった固有値数 (注意事項 (e) 参照)
15	IS1	I	1	入力	処理スイッチ IS1 ≤ 0: 固有値を絶対値の小さい方から求める. IS1 > 0: 大きい方から求める.

項番	引数名	型	大きさ	入出力	内　　容
16	IS2	I	1	入 力	スツルム列チェックスイッチ IS2 $\leq$ 0: チェックを行わない. IS2 $>$ 0: チェックを行う.
17	W1	$\begin{Bmatrix}\mathrm{D}\\\mathrm{R}\end{Bmatrix}$	内容参照	ワーク	作業領域 **大きさ:** IS2 ≤ 0 の時 : $2\times\mathrm{N}\times q+q\times q+2\times q+\mathrm{N}$ IS2 >0 の時 : $2\times\mathrm{N}\times q+q\times q+2\times q+\mathrm{N}+\ell\times\mathrm{N}$ ただし, ここで $q=\min(2\times m,\mathrm{N},m+8)$ IS1 ≤ 0 の時 : $\ell=\mathrm{MAB}+1$ IS1 >0 の時 : $\ell=\mathrm{MBB}+1$
18	IW1	I	N	ワーク	作業領域
19	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a) $0<\mathrm{N}\leq\mathrm{LNV}$

(b) $0\leq\mathrm{MAB}<\mathrm{N}$
$0\leq\mathrm{MBB}<\mathrm{N}$

(c) $\mathrm{MAB}<\mathrm{LMA}$

(d) $\mathrm{MBB}<\mathrm{LMB}$

(e) $0<\mathrm{M}\leq\mathrm{N}$

(5) エラーインディケータ

IERR の値	意　　味	処 理 内 容
0	正常終了.	
1000	N = 1 であった.	$\mathrm{E}(1)\leftarrow\mathrm{A}(1,1)/\mathrm{B}(1,1)$, $\mathrm{VE}(1,1)\leftarrow 1.0$ とする.
3000	制限条件 (a), (b), (c), (d) または (e) を満足しなかった.	処理を打ち切る.
4000	処理途中でエラーが起きた.	
$5000+i$	指定された回数以内で収束しなかった.	処理を打ち切る. i 番目までの固有値, 固有ベクトルは求められている.

(6) **注意事項**

(a) このサブルーチンは, 求めたい固有値数が行列の次数に比較してごく小さい場合 (m≪N) に有効である. それ以外の場合には, 他のサブルーチン 4.14.1 $\left\{\begin{matrix}\text{DCGSAA}\\ \text{RCGSAA}\end{matrix}\right\}$, 4.13.1 $\left\{\begin{matrix}\text{DCGGAA}\\ \text{RCGGAA}\end{matrix}\right\}$等を使用した方がよい.

(b) このサブルーチンでは, 以下の条件が満たされた時に固有値は収束したとみなす. この時, 固有ベクトルは ITOL/2 以上の精度を持つ.

$|\frac{a_i^n - a_i^{n-1}}{a_i^n}| \leq 10.0^{-\mathrm{ITOL}}$　　　　(a_i^n:n 回反復後の第 i 番目の固有値)

ITOL の入力値として 0 以下または $-$ LOG10 (ε) より大きい数が与えられた場合, 内部で最適値が使用される. (ε:誤差判定のための単位).

(c) 固有値は絶対値の小さい順 (または大きい順) に配列 E に格納される.

(d) NITE の入力値として 0 以下の数が与えられた場合は, 既定値として 20 を採用する.

(e) このサブルーチンには, 算出された固有値に対しスツルム列の性質を利用したチェックを行う機能がある. これにより, 計算されなかった固有値数が算出されるが, この時演算回数は $\mathrm{N} \times \mathrm{MAX(MAB, MBB)}^2$ 程度増加する.

例

6, 5, 3, 2, 1 を固有値として持つような一般化固有値問題に対して, 絶対値最小のものより 3 個の固有値を求めたとする. この時, 固有値の解として 5, 2, 1 が求められたとすると, 3 が解として求められなかったので MST には 1 が返される.

なお, この機能は固有値がすべて正の時のみ有効である.

(7) **使用例**

(a) 問 題

$$A = \begin{bmatrix} 611 & 196 & -192 & 407 & -8 & 0 & 0 & 0 \\ 196 & 899 & 113 & -192 & -71 & -43 & 0 & 0 \\ -192 & 113 & 899 & 196 & 61 & 49 & 8 & 0 \\ 407 & -192 & 196 & 611 & 8 & 44 & 59 & -23 \\ -8 & -71 & 61 & 8 & 411 & -599 & 208 & 208 \\ 0 & -43 & 49 & 44 & -599 & 411 & 208 & 208 \\ 0 & 0 & 8 & 59 & 208 & 208 & 99 & -911 \\ 0 & 0 & 0 & -23 & 208 & 208 & -911 & 99 \end{bmatrix}$$

$$B = \begin{bmatrix} 171 & 18 & 33 & -21 & -17 & 0 & 0 & 0 \\ 18 & 171 & -21 & 33 & 13 & -17 & 0 & 0 \\ 33 & -21 & 171 & 18 & 25 & -36 & -17 & 0 \\ -21 & 33 & 18 & 171 & -36 & 25 & 13 & -17 \\ -17 & 13 & 25 & -36 & 171 & 18 & 33 & -21 \\ 0 & -17 & -36 & 25 & 18 & 171 & -21 & 33 \\ 0 & 0 & -17 & 13 & 33 & -21 & 171 & 18 \\ 0 & 0 & 0 & -17 & -21 & 33 & 18 & 171 \end{bmatrix}$$

のとき, $A\boldsymbol{x} = \lambda B\boldsymbol{x}$ の固有値それに対応する固有ベクトルを, 固有値の絶対値最小のものより 3 個求める.

(b) 入力データ

行列 A, LMA=11, N=8, MAB=4, 行列 B, LMB=11, MBB=4, M=3, LNV=11

(c) 主プログラム

```
      PROGRAM BCGBFF
! *** EXAMPLE OF DCGBFF ***
      IMPLICIT REAL(8)(A-H,O-Z)
      CHARACTER*80 FMT
      PARAMETER ( LMA=11, LMB=11, LNV=10, LN=10, LNQ=10 )
      PARAMETER ( LW=LNQ*LNQ+2*LNQ+LN*(2*LNQ+1+LN) )
      DIMENSION A(LMA,LN), B(LMB,LN), E(LN), VE(LNV,LNQ),&
                W1(LW), IW1(LN)
!
      READ(5,*) N, MAB, MBB, M
      DO 10 J=1, MAB+1
         READ(5,*) (A(J,I), I=MAB-J+2, N)
   10 CONTINUE
      DO 20 J=1, MBB+1
         READ(5,*) (B(J,I), I=MBB-J+2, N)
   20 CONTINUE
!
      WRITE(6,1000) N, MAB, MBB, M
      WRITE(6,1100) ’A’
      DO 30 J=1, MAB+1
         WRITE(FMT,1200) (MAB-J+1)*7+6, N-MAB+J-1
         WRITE(6,FMT) (A(J,I), I=MAB-J+2, N)
   30 CONTINUE
      WRITE(6,1100) ’B’
      DO 40 J=1, MBB+1
         WRITE(FMT,1200) (MBB-J+1)*7+6, N-MBB+J-1
         WRITE(6,FMT) (B(J,I), I=MBB-J+2, N)
   40 CONTINUE
!
      CALL DCGBFF(A,LMA,N,MAB,B,LMB,MBB,M,0,0,E,VE,LNV,MST,0,1,W1,IW1,&
                  IERR)
!
      WRITE(6,1300) IERR
!
      DO 60 K=1, M-3, 4
         WRITE(6,1400) (’EIGENVALUE ’, I=1, 4)
         WRITE(6,1500) (E(I), I=K, K+3)
         WRITE(6,1400) (’EIGENVECTOR’, I=1, 4)
         DO 50 J=1, N
            WRITE(6,1500) (VE(J,I), I=K, K+3)
   50    CONTINUE
   60 CONTINUE
      IF(MOD(M,4).NE.0) THEN
         WRITE(6,1400) (’EIGENVALUE ’, I=M/4*4+1, M)
         WRITE(6,1500) (E(I), I=M/4*4+1, M)
         WRITE(6,1400) (’EIGENVECTOR’, I=M/4*4+1, M)
         DO 70 J=1, N
            WRITE(6,1500) (VE(J,I), I=M/4*4+1, M)
   70    CONTINUE
      ENDIF
      WRITE(6,1600) MST
      STOP
!
 1000 FORMAT(’ ’,/,/,&
             ’ ***  DCGBFF  ***’,/,/,&
             ’  **  INPUT  **’,/,/,&
             ’      N   = ’, I2,/,/,&
             ’      MAB = ’, I2,/,/,&
             ’      MBB = ’, I2,/,/,&
             ’      M   = ’, I2)
 1100 FORMAT(’ ’,/,&
             ’      INPUT MATRIX ’,A1,/)
 1200 FORMAT(’( ,’, I3,’X,’, I2,’(F7.1))’)
 1300 FORMAT(’ ’,/,/,&
             ’  **  OUTPUT  **’,/,/,&
             ’      IERR = ’, I4)
 1400 FORMAT(’ ’,/,1X, 4(5X, A11, 2X))
 1500 FORMAT(’ ’, 4(2X, 1PD14.7, 2X))
 1600 FORMAT(’ ’,/,&
             ’      MISSED EIGENVALUES = ’, I2)
      END
```

(d) 出力結果

```
***  DCGBFF  ***

 **  INPUT  **

     N   =  8

     MAB =  4

     MBB =  4

     M   =  3

     INPUT MATRIX A

                                    -8.0  -43.0    8.0  -23.0
                             407.0  -71.0   49.0   59.0  208.0
                     -192.0 -192.0   61.0   44.0  208.0  208.0
              196.0  113.0  196.0    8.0 -599.0  208.0 -911.0
       611.0  899.0  899.0  611.0  411.0  411.0   99.0   99.0

     INPUT MATRIX B

                                   -17.0  -17.0  -17.0  -17.0
                             -21.0   13.0  -36.0   13.0  -21.0
                       33.0   33.0   25.0   25.0   33.0   33.0
               18.0  -21.0   18.0  -36.0   18.0  -21.0   18.0
       171.0  171.0  171.0  171.0  171.0  171.0  171.0  171.0


 **  OUTPUT  **

     IERR =    0

     EIGENVALUE        EIGENVALUE        EIGENVALUE
  -2.8694668D-02     1.1418645D-01     4.6073986D+00

     EIGENVECTOR       EIGENVECTOR       EIGENVECTOR
  -2.9518370D-02    -3.3280520D-02     1.3357085D-03
   1.8549154D-02     1.2860714D-02    -2.3833439D-02
  -1.8508500D-02    -1.2749259D-02    -1.4890226D-02
   2.7733942D-02     3.5436570D-02     5.2891703D-04
   3.2017538D-02    -3.0821630D-02    -3.4967437D-02
   3.0482006D-02    -3.2110386D-02     3.4727146D-02
   1.5596117D-02    -1.6287379D-02    -2.6667696D-02
   1.7726852D-02    -1.3617717D-02     2.5869490D-02

     MISSED EIGENVALUES =  1
```

付 録 A　用語説明

(1) **行列**

$m \times n$ の行列 A とは, $m \times n$ 個の元 $a_{i,j}$ $(i = 1, 2, \cdots, m;\ j = 1, 2, \cdots, n)$ の矩形の配置をいう.

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}$$

元 $a_{i,j}$ を行列 A の (i, j) 要素と呼ぶ. 行列の元としては, 複素数または実数を考える. 特に, 複素数を元とする行列を複素行列, 実数を元とする行列を実行列と呼ぶ. また, 特に $m = n$ である場合, 行列 A を正方行列と呼ぶ. 行列 A は, (a_{ij}) と略記する場合がある. なお, 本書では必要に応じて行の添字 i と列の添字 j を区別するために $(a_{i,j})$ を用いる.

(2) **(数) ベクトル**

$1 \times n$ の行列を n 次の行ベクトル, $m \times 1$ の行列を m 次の列ベクトルと呼ぶ. 両者を特に区別する必要がない場合には単にベクトルと呼ぶ. なお, 数学的にはさらに抽象化した概念としてベクトルを定義し, ここで述べた「ベクトル」は数ベクトルと呼ばれる. 抽象化したベクトルの定義については「ベクトル空間」についての説明を参照されたい.

(3) **行列積**

2 つの行列 $A = (a_{i,j})$, $B = (b_{k,l})$ において, 行列 A の列数と行列 B の行数が等しい場合にのみ行列積 $A \bullet B = (c_{i,l})$ が定義されて,

$$c_{i,l} = \sum_j a_{i,j} \bullet b_{j,l}$$

となる.

(4) **行列ベクトル積**

行列積 $A \bullet B$ において, 特に行列 B が列ベクトル$\boldsymbol{x}$である場合, 積 $A\boldsymbol{x}$ を行列ベクトル積という.

(5) **行列の転置**

$m \times n$ の行列 $A = (a_{i,j})$ $(i = 1, 2, \cdots, m;\ j = 1, 2, \cdots, n)$ において行と列を入れ換えてできる行列 $A' = (a_{j,i})$ を行列 A の転置行列と呼び, A^T で表す. なお, 転置行列は ${}^tA$ と表す場合もある.

(6) **行列の (主) 対角**

$n \times n$ の正方行列 $A = (a_{i,j})$ $(i, j = 1, 2, \cdots, n)$ において, $a_{i,i}$ $(i = 1, 2, \cdots, n)$ の並びを (主) 対角と呼び, その要素を (主) 対角要素と呼ぶ. また特に, 対角にのみ非零要素 $a_{i,i}$ を持つ行列を対角行列 $(\mathrm{diag}(a_{i,i}))$ と呼ぶ.

(7) **単位行列**

$n \times n$ の対角行列 A=diag$(a_{i,i})$ において, 対角要素 $a_{i,i}$ $(i = 1, 2, \cdots, n)$ がすべて 1 の行列を単位行列と呼び, 記号 E または I を用いて表す.

(8) **逆行列**

正方行列 A に対して, $A \cdot B = B \cdot A = E$ (E は単位行列) を満たす正方行列 B が存在する場合に, 行列 B を行列 A の逆行列と呼び, 記号 A^{-1} で表す.

(9) **一般逆行列**

$m \times n$ の行列 A に対して, 以下の関係を満たすような $n \times m$ 行列 X が一意的に存在し, この行列 X を行列 A の (Moore-Penrose の) 一般逆行列と呼び, 記号 $A^{\dagger}$ で表す.

- $AXA \; = \; A$
- $XAX \; = \; X$
- $(AX)^T \; = \; AX$
- $(XA)^T \; = \; XA$

(10) **行列の下三角ならびに上三角**

$n \times n$ の正方行列 $A = (a_{i,j}) \; (i, j = 1, 2, \cdots, n)$ において, $a_{i,j} \; (i > j)$ の要素をまとめて下三角, $a_{i,j} \; (i < j)$ の要素をまとめて上三角と呼ぶ. なお, 上三角ならびに下三角の定義に対角を含める場合もある. 対角を含めた下三角にのみ非零要素を持つ行列を下三角行列, 対角を含めた上三角にのみ非零要素を持つ行列を上三角行列とそれぞれ呼ぶ.

(11) **共役転置行列**

複素行列 A の各要素の共役な複素数を要素とする行列の転置行列を共役転置行列と呼び, 記号 A^* で表す. 行列の要素が実数の場合には $A^* = A^T$ である.

(12) **対称行列**

$A = A^T$ が成立する正方行列を対称行列と呼ぶ. 対称行列では, $a_{i,j} = a_{j,i}$ である.

(13) **エルミート行列**

$A = A^*$ が成立する正方行列をエルミート行列と呼ぶ. エルミート行列では, $a_{i,j}$ と $a_{j,i}$ は複素共役である.

(14) **ユニタリ行列**

$UU^* = I(I$ は単位行列) が成立する正方行列 U をユニタリ行列と呼ぶ.

(15) **直交行列**

$AA^T = I(I$ は単位行列) が成立する実正方行列 A を直交行列と呼ぶ.

(16) **行列の副対角**

$n \times n$ の正方行列 $A = (a_{i,j}) \; (i, j = 1, 2, \cdots, n)$ において, $a_{i,i+p} \; (i = 1, 2, \cdots, n-p)$ の並びを第 p 上副対角, $a_{i+q,i} \; (i = 1, 2, \cdots, n-q)$ の並びを第 q 下副対角と呼び, その要素を, それぞれ第 p 上副対角要素, 第 q 下副対角要素と呼ぶ. また, 両者をまとめて単に副対角要素と呼ぶこともある.

(17) **バンド行列**

$n \times n$ の正方行列 $A = (a_{i,j}) \; (i, j = 1, 2, \cdots, n)$ において, 主対角とそれに隣接するいくつかの上副対角ならびに下副対角にのみ 0 でない要素がある行列をバンド行列と呼ぶ. 対角からもっともはなれた 0 でない要素を含む副対角が第 u 上副対角と第 l 下副対角である場合に, 値 u と l をそれぞれ上バンド幅, 下バンド幅と呼ぶ. 特に, $u = l$ の場合これを単にバンド幅と呼ぶ.

(18) **3 重対角行列**

特に, 上バンド幅ならびに下バンド幅が 1 の行列を 3 重対角行列と呼ぶ.

(19) **ヘッセンベルグ行列**

$n \times n$ の正方行列 $A = (a_{i,j}) \; (i, j = 1, 2, \cdots, n)$ において, 第 1 下副対角を除いた下三角のすべての要素が 0 である行列をヘッセンベルグ行列と呼ぶ. 行列の固有値を求める場合に, 一般の行列をこの行列に変換する.

(20) **準上三角行列**

$n \times n$ の正方行列 $A = (a_{i,j})$ $(i, j = 1, 2, \cdots, n)$ において, 第 1 下副対角の連続する 2 つの副対角要素の少なくとも一方が必ず 0 であり, 第 1 下副対角を除いた下三角のすべての要素が 0 である行列を準上三角行列行列と呼ぶ. ヘッセンベルグ行列の特殊な場合である.

(21) **スパース行列**

一般に, 非零要素の数が全要素数に比べて少ない行列をスパース行列と呼ぶ. スパース行列のうちで要素の並びに規則性があり, この規則性を活用することによって, 問題を解く有効な解法が作成されている場合に, この行列を特に, 規則スパース行列と呼んでいる. 規則スパース行列でない行列は, 不規則スパースと呼ばれている. たとえば, バンド幅が小さいバンド行列は規則スパース行列の一種である.

(22) **正則行列, 特異行列**

正方行列 A が逆行列を持つとき, 行列 A は正則 (*regular*) であるという. 正則でない行列は特異 (*singular*) であるという. 正則な行列を係数に持つ連立 1 次方程式の解は, 一意に定まる. ただし現実には, 有限桁で計算を行うので, 丸め誤差の影響が避けられず, 正則な行列と特異な行列の区別は曖昧になる. たとえば, 数学的に特異な行列を用いて数値的に連立 1 次方程式を解いた場合でも, 見かけ上解が得られる場合がある. したがって, 特にほとんど特異な行列を係数に持つ連立 1 次方程式を解く場合, 見かけ上得られる解については, その妥当性について十分な吟味が必要である.

(23) **LU 分解**

直接法で, 連立 1 次方程式 $A\boldsymbol{x} = \boldsymbol{b}$ を解く場合には, まず係数行列 A を下三角行列 L と上三角行列 U の積に $A = LU$ と分解する. この分解のことを LU 分解と呼ぶ. このような分解を行えば, 連立 1 次方程式の解$\boldsymbol{x}$は

$$\begin{aligned} L\boldsymbol{y} &= \boldsymbol{b} \\ U\boldsymbol{x} &= \boldsymbol{y} \end{aligned}$$

を逐次解くことによって得られる. この 2 つの連立 1 次方程式は係数行列が三角行列であるので前進代入ならびに後退代入を用いて容易に解くことができる. なお, 行列 A の LU 分解は A が正則であれば, たとえば行列 L の対角要素を 1 に固定することによって一意に定まる. また, 連立 1 次方程式を解く場合には, 一般に部分軸選択を行いながら LU 分解を行うので, 軸選択による行交換行列を P として $PA = LU$ を満たす三角行列 L, U をそれぞれ求める.

(24) $\mathrm{U^TDU}$ **分解**

連立 1 次方程式の係数行列が対称行列である場合には, 軸選択を行わないで LU 分解を行って得られる下三角行列 L と上三角行列 U の間には, $L = U^T D$ の関係がある. ここで D は対角行列である. したがって, L または U の片方と D のみを陽に求めれば, 連立 1 次方程式を解くことができる. 係数行列から U と D を陽に求める分解を $\mathrm{U^TDU}$ 分解と呼ぶ.

(25) $\mathrm{U^*DU}$ **分解**

連立 1 次方程式の係数行列がエルミート行列である場合には, 軸選択を行わないで LU 分解を行って得られる下三角行列と上三角行列 U の間には, $L = U^* D$ の関係がある. ここで D は対角行列である. したがって, L または U の片方と D のみを陽に求めれば, 連立 1 次方程式を解くことができる. 係数行列から U と D を陽に求める分解を $\mathrm{U^*DU}$ 分解と呼ぶ.

(26) **正定値 (Positive definite)**

実対称行列またはエルミート行列 A は, 任意のベクトル$\boldsymbol{x}$ $(\boldsymbol{x} \neq \boldsymbol{0})$ に対して, $\boldsymbol{x}^* A \boldsymbol{x} > 0$ を満たす場合, 正 (定) 値, $\boldsymbol{x}^* A \boldsymbol{x} < 0$ を満たす場合, 負値とそれぞれ呼ぶ. 行列 A が正定値行列であることは, 次の 2 つの条件と同値である.

(a) 行列 A の固有値がすべて正である.

(b) 行列 A の主小行列式がすべて正である.

数学的には, 正定値行列は軸選択を行わなくても LU 分解することができるが, 現実には軸選択を行わなければ, 数値的に安定して LU 分解を行えない場合がある.

(27) **実数固有値 (Real eigenvalue)**

実数成分の正方行列の固有値が全て実数であることの必要十分条件は, 2 つの実対称行列の積であることである. また, 複素数成分の正方行列の固有値が全て実数であることの必要十分条件は, 2 つのエルミート行列の積であることである.

(28) **対角優位 (Diagonally dominant)**

$n \times n$ の正方行列 $A = (a_{i,j})$ $(i, j = 1, 2, \cdots, n)$ において,

$$|a_{i,i}| \;>\; \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{i,j}| \;\; (i = 1, 2, \cdots, n)$$

が成立する場合, 行列 A を対角優位な行列と呼ぶ. 数学的には, 対角優位な行列は軸選択を行わなくても LU 分解することができるが, 現実には軸選択を行わなければ, 数値的に安定して LU 分解を行えない場合がある.

(29) **ベクトル空間**

集合 V が次の条件 (a), (b) を満足するとき, V をベクトル空間とよびその要素をベクトルと呼ぶ.

(a) V の 2 つの要素$\boldsymbol{a}$, $\boldsymbol{b}$に対して和 $\boldsymbol{a}+\boldsymbol{b}$ が V の要素として一意に定まり, 次の性質を満たす.

  i. $(\boldsymbol{a}+\boldsymbol{b})+\boldsymbol{c} = \boldsymbol{a}+(\boldsymbol{b}+\boldsymbol{c})$ (結合則)
  ただし, $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$は, V の任意の要素.

  ii. $\boldsymbol{a}+\boldsymbol{b} = \boldsymbol{b}+\boldsymbol{a}$ (交換則)
  ただし, $\boldsymbol{a}$, $\boldsymbol{b}$は, V の任意の要素.

  iii. 零ベクトルと呼ばれる V の要素$\boldsymbol{0}$が存在し, V の任意の要素$\boldsymbol{a}$に対して, $\boldsymbol{a}+\boldsymbol{0} = \boldsymbol{a}$

  iv. V の任意の要素$\boldsymbol{a}$に対して, $\boldsymbol{a}+\boldsymbol{b} = \boldsymbol{0}$ となる V の要素$\boldsymbol{b}$がただ一つ存在する. なお, このとき, $\boldsymbol{b}$は $-\boldsymbol{a}$ と表される.

(b) V の任意の要素$\boldsymbol{a}$と複素数 c に対して, $\boldsymbol{a}$の c 倍 $c\boldsymbol{a}$ が V の要素として一意に定まり, 次の性質を満たす (スカラ倍).

  i. $c(\boldsymbol{a}+\boldsymbol{b}) = c\boldsymbol{a}+c\boldsymbol{b}$ (ベクトル分配則)

  ii. $(c+d)\boldsymbol{a} = c\boldsymbol{a}+d\boldsymbol{a}$ (スカラ分配則)

  iii. $(cd)\boldsymbol{a} = c(d\boldsymbol{a})$

  iv. $1\boldsymbol{a} = \boldsymbol{a}$

(30) **一次結合, 一次独立, 一次従属**

ベクトル空間 V の k 個のベクトル $\boldsymbol{a}_1$, $\cdots$, $\boldsymbol{a}_k$ と複素数 c_1, $\cdots$, c_k とによって作られるベクトル

$$c_1\boldsymbol{a}_1 + \cdots + c_k\boldsymbol{a}_k$$

を $\boldsymbol{a}_1$, $\cdots$, $\boldsymbol{a}_k$ の一次結合といい, c_1, $\cdots$, c_k をその係数という. すべてが 0 ではないある係数 c_1, $\cdots$, c_k に対して

$$c_1\boldsymbol{a}_1 + \cdots + c_k\boldsymbol{a}_k = \boldsymbol{0}$$

となるとき, ベクトルの集合 $\{\boldsymbol{a}_1, \cdots, \boldsymbol{a}_k\}$ は一次従属であるといい, そうでないときは, 一次独立であるという.

(31) **基底**

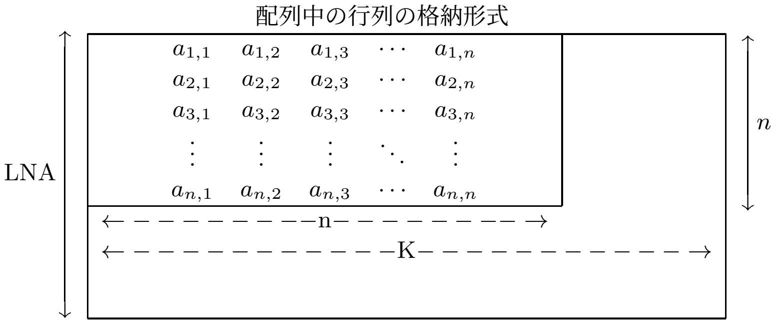
S をベクトル空間 V の任意の部分集合とし, S に含まれる一次独立なベクトルの組を $\{\boldsymbol{a}_1,\ \cdots,\ \boldsymbol{a}_k\}$ とする. S の任意のベクトル$\boldsymbol{b}$に対して $\{\boldsymbol{a}_1,\ \cdots,\ \boldsymbol{a}_k,\ \boldsymbol{b}\}$ が一次従属であるとき, $\{\boldsymbol{a}_1,\ \cdots,\ \boldsymbol{a}_k\}$ は S において極大であると呼ばれ, S としてベクトル空間 V そのものをとった場合, この一次独立なベクトルの組をベクトル空間 V の基底と呼ぶ. なお, V の基底を構成するベクトルの個数を V の次元と呼ぶ. また, n 次元ベクトル空間 V_n の任意の基底を $\{\boldsymbol{u}_1,\ \cdots,\ \boldsymbol{u}_n\}$ とすると, V_n の任意のベクトル$\boldsymbol{a}$ は, $\{\boldsymbol{u}_1,\ \cdots,\ \boldsymbol{u}_n\}$ の一次結合として一意に表される.

付 録 B　配列データの取扱い方法

B.1　行列に対応した配列データ

本ライブラリにおいては, しばしば行列に対応した配列データが使用されるが, 以下にその取扱い方法を述べる. 配列データを使用するサブルーチンを引用する場合, 利用者は引用する側のプログラム内で, その配列を宣言しておかなければならない. 宣言された配列を A (LNA, K) とすると, $n \times n$ 型行列 $A = (a_{i,j})$ $(i = 1, 2, \cdots, n;\ j = 1, 2, \cdots, n)$ は次の図のように格納される. この時の LNA を**整合寸法**という. 行列に対応した配列を引数として使用する場合には, 引数として配列名, 次数のほかに, この整合寸法もサブルーチンに引渡さなければならない. これは, 行列の要素 $a_{i,j}$ $(i = 1, 2, \cdots$,LNA; $j = 1, 2, \cdots$,K) は, 配列の要素 A(i, j) (i=1, 2, $\cdots$, LNA; j=1, 2, $\cdots$, K) と次のように主記憶上で対応している必要があるためである.

図 B−1　配列中の行列の格納形式



備　考
- a.　LNA$\geq n$, K$\geq n$ でなければならない.
- b.　行列の要素 $a_{i,j}$ は配列の要素 A(i, j) に対応する.

$a_{1,1}$	$a_{2,1}$	$\cdots$	$a_{\mathrm{LNA},1}$	$a_{1,2}$	$a_{2,2}$	$\cdots$
$\updownarrow$	$\updownarrow$	$\cdots$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\cdots$
A(1, 1)	A(2, 1)	$\cdots$	A(LNA, 1)	A(1, 2)	A(2, 2)	$\cdots$

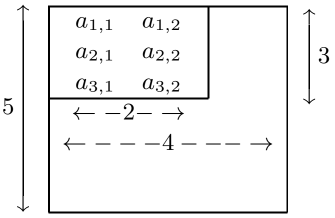
例 DAM1AD(実行列の和) の場合

3×2 型行列 A, B の和を行列 C に求めるとする. 対応する配列 A, B, C の大きさをすべて (5, 4) で宣言すると, 宣言文および CALL 文は次のようになる.

```
      REAL(8) A(5, 4), B(5, 4), C(5, 4)
      INTEGER IERR
!
      CALL DAM1AD(A, 5, 3, 2, B, 5, C, 5, IERR)
```

配列 A には, データが次のように格納される. 配列 B, C についても同様である.

図 B–2　配列 A 中の格納形式



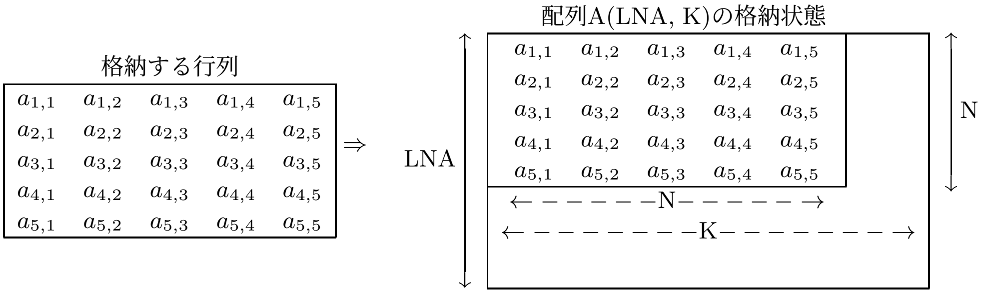
次数の異なるいくつかの配列をデータとして取り扱う場合には, そのうち最も大きな次数を LNA とするような配列を一つ用意しておけば, この配列を逐次利用することができる. ただし, この時, 整合寸法として常に LNA の値を与える必要がある.

B.2　データの格納方法

行列データの格納方法は, その行列の型によって異なっている. 以下にその方法を示す.

B.2.1　実行列 (2 次元配列型)

図 B–3　実行列 (2 次元配列型) の格納形式



備　考

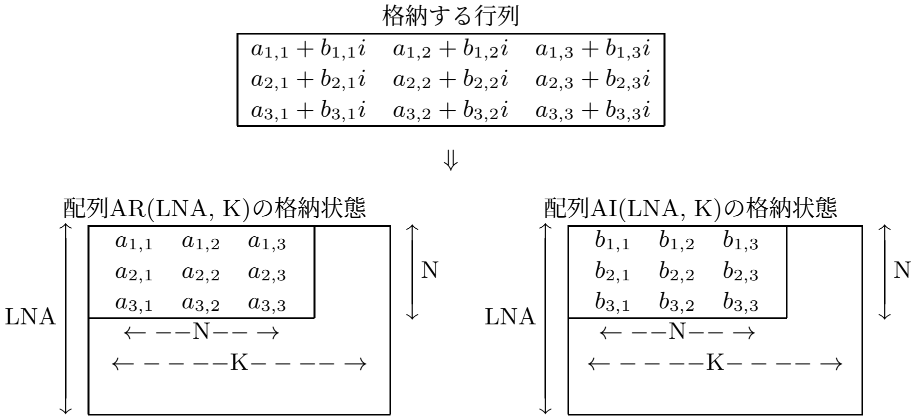
a.　LNA $\geq$ N, K $\geq$ N を満たさなければならない.

B.2.2　複素行列

(1) **2 次元配列型, 実数引数型**

実部と虚部に分けて別々の配列に格納する.

図 B−4　複素行列 (2 次元配列型)(実数引数型) の格納形式

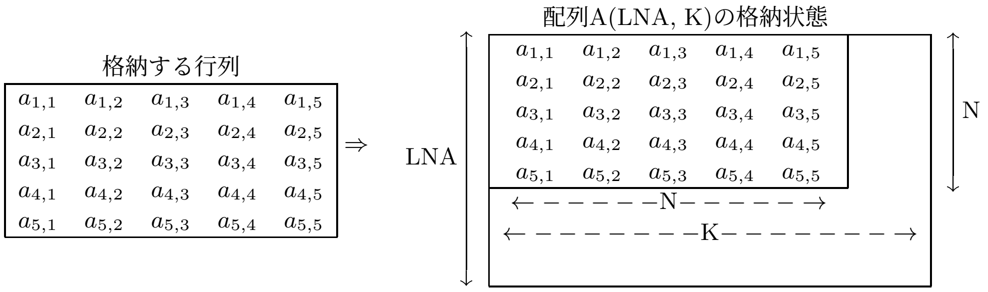


備　考

a.　LNA $\geq$ N, K $\geq$ N を満たさなければならない.

(2) **2 次元配列型, 複素引数型**

図 B−5　複素行列 (2 次元配列型)(複素引数型) の格納形式



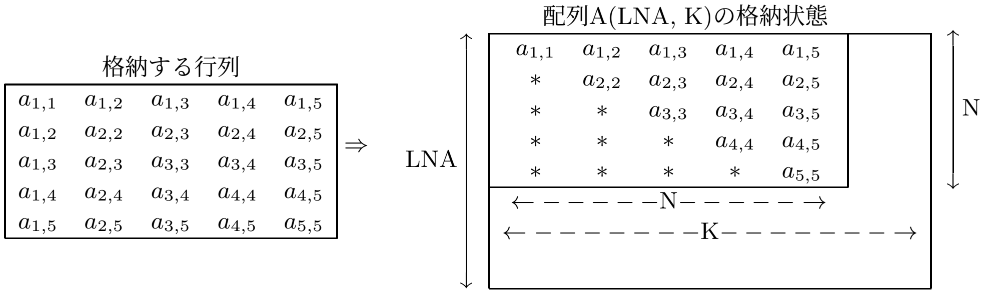
備　考

a.　LNA $\geq$ N, K $\geq$ N を満たさなければならない.

B.2.3 実対称行列, 正値対称行列

(1) 2 次元配列型, 上三角型

図 B−6 実対称行列 (2 次元配列型)(上三角型) の格納形式



$$\begin{array}{ccccc} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\ a_{1,2} & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} \\ a_{1,3} & a_{2,3} & a_{3,3} & a_{3,4} & a_{3,5} \\ a_{1,4} & a_{2,4} & a_{3,4} & a_{4,4} & a_{4,5} \\ a_{1,5} & a_{2,5} & a_{3,5} & a_{4,5} & a_{5,5} \end{array} \Rightarrow \begin{array}{ccccc} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\ * & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} \\ * & * & a_{3,3} & a_{3,4} & a_{3,5} \\ * & * & * & a_{4,4} & a_{4,5} \\ * & * & * & * & a_{5,5} \end{array}$$

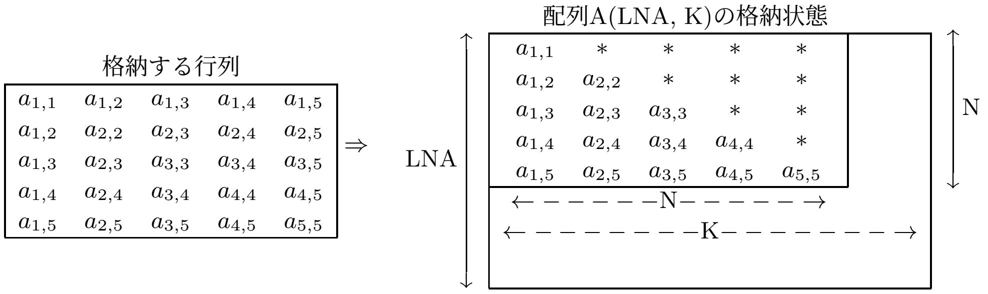
備 考

a. $*$ は, 任意の値であることを示す.

b. LNA $\geq$ N, K $\geq$ N を満たさなければならない.

(2) 2 次元配列型, 下三角型

図 B−7 実対称行列 (2 次元配列型)(下三角型) の格納形式



$$\begin{array}{ccccc} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\ a_{1,2} & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} \\ a_{1,3} & a_{2,3} & a_{3,3} & a_{3,4} & a_{3,5} \\ a_{1,4} & a_{2,4} & a_{3,4} & a_{4,4} & a_{4,5} \\ a_{1,5} & a_{2,5} & a_{3,5} & a_{4,5} & a_{5,5} \end{array} \Rightarrow \begin{array}{ccccc} a_{1,1} & * & * & * & * \\ a_{1,2} & a_{2,2} & * & * & * \\ a_{1,3} & a_{2,3} & a_{3,3} & * & * \\ a_{1,4} & a_{2,4} & a_{3,4} & a_{4,4} & * \\ a_{1,5} & a_{2,5} & a_{3,5} & a_{4,5} & a_{5,5} \end{array}$$

備 考

a. $*$ は, 任意の値であることを示す.

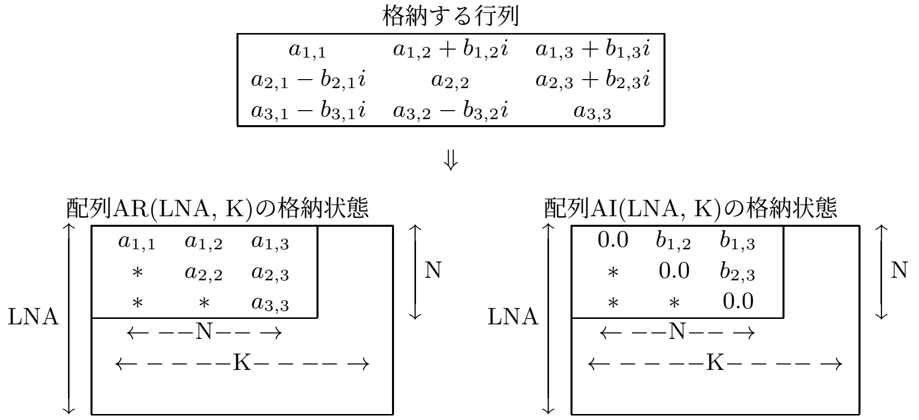
b. LNA $\geq$ N, K $\geq$ N を満たさなければならない.

B.2.4 エルミート行列

(1) **2 次元配列型, 実数引数型, 上三角型**

上三角部分の実部と虚部を別々の配列に格納する.

図 B−8　エルミート行列 (2 次元配列型)(実数引数型)(上三角型) の格納形式



格納する行列

$$\begin{array}{|ccc|}\hline a_{1,1} & a_{1,2}+b_{1,2}i & a_{1,3}+b_{1,3}i \\ a_{2,1}-b_{2,1}i & a_{2,2} & a_{2,3}+b_{2,3}i \\ a_{3,1}-b_{3,1}i & a_{3,2}-b_{3,2}i & a_{3,3} \\ \hline \end{array}$$

⇓

配列AR(LNA, K)の格納状態

$$\begin{array}{ccc} a_{1,1} & a_{1,2} & a_{1,3} \\ * & a_{2,2} & a_{2,3} \\ * & * & a_{3,3} \end{array}$$

配列AI(LNA, K)の格納状態

$$\begin{array}{ccc} 0.0 & b_{1,2} & b_{1,3} \\ * & 0.0 & b_{2,3} \\ * & * & 0.0 \end{array}$$

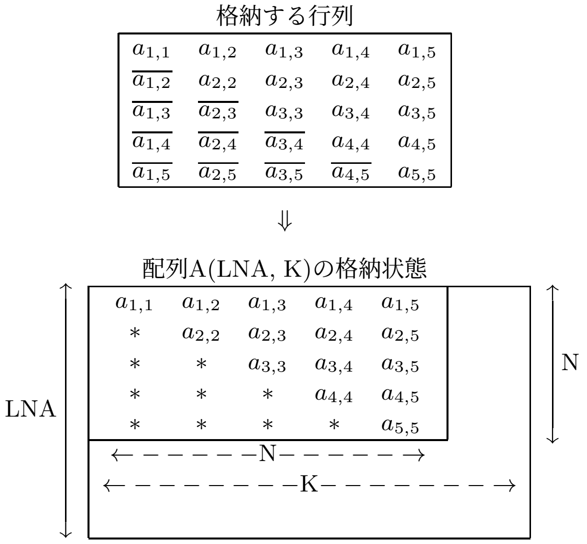
備　考

a.　$*$ は, 任意の値であることを示す.

b.　LNA $\geq$ N, K $\geq$ N を満たさなければならない.

(2) **2 次元配列型, 複素引数型, 上三角型**

図 B−9　エルミート行列 (2 次元配列型)(複素引数型)(上三角型) の格納形式



格納する行列

$$\begin{array}{|ccccc|}\hline a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\ \overline{a_{1,2}} & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} \\ \overline{a_{1,3}} & \overline{a_{2,3}} & a_{3,3} & a_{3,4} & a_{3,5} \\ \overline{a_{1,4}} & \overline{a_{2,4}} & \overline{a_{3,4}} & a_{4,4} & a_{4,5} \\ \overline{a_{1,5}} & \overline{a_{2,5}} & \overline{a_{3,5}} & \overline{a_{4,5}} & a_{5,5} \\ \hline \end{array}$$

⇓

配列A(LNA, K)の格納状態

$$\begin{array}{ccccc} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\ * & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} \\ * & * & a_{3,3} & a_{3,4} & a_{3,5} \\ * & * & * & a_{4,4} & a_{4,5} \\ * & * & * & * & a_{5,5} \end{array}$$

備　考

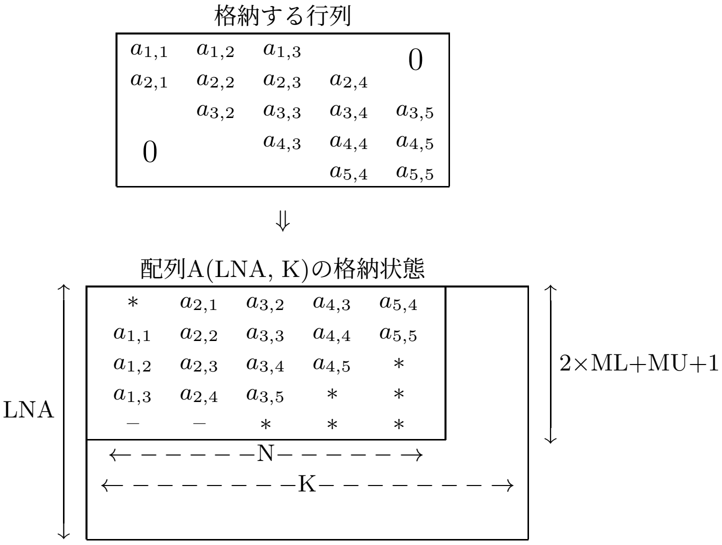
a.　x の複素共役を $\overline{x}$ で表している.

b.　$*$ は, 任意の値であることを示す.

c.　LNA $\geq$ N, K $\geq$ N を満たさなければならない.

B.2.5　実バンド行列 (バンド型)

図 B−10　実バンド行列 (バンド型) の格納形式



備　考

a.　$*$ は, 任意の値であることを示す.

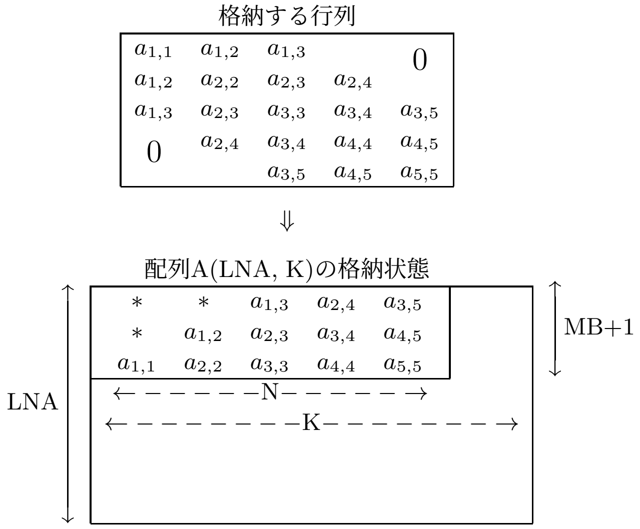
b.　–は, 行列の LU 分解時に必要となる領域である.

c.　MU は上バンド幅, ML は下バンド幅である.

d.　LNA $\geq 2 \times$ ML + MU + 1, K $\geq$ N を満たさなければならない. (ただし, LU 分解を伴わない場合には, LNA $\geq$ ML + MU + 1, K $\geq$ N でよい).

B.2.6　実対称バンド行列, 正値対称バンド行列 (対称バンド型)

図 B−11　実対称バンド行列 (対称バンド型) の格納形式



備　考

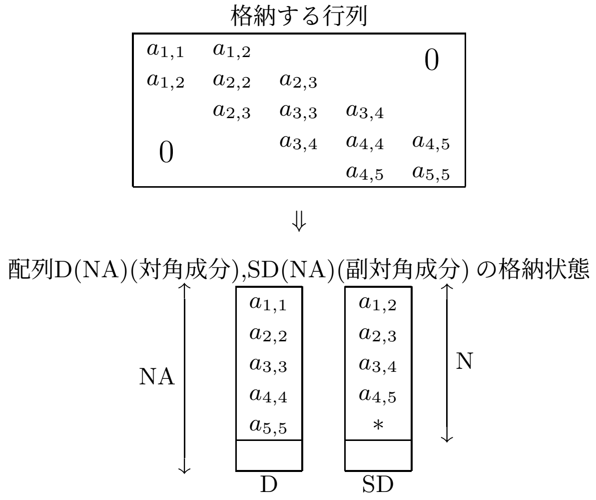
a.　$*$ は, 任意の値であることを示す.

b.　MB は, バンド幅である.

c.　$\mathrm{LNA} \geq \mathrm{MB}+1$, $\mathrm{K} \geq \mathrm{N}$ を満たさなければならない.

B.2.7 実対称 3 重対角行列, 正値対称 3 重対角行列 (ベクトル型)

図 B−12 実対称 3 重対角行列 (ベクトル型) の格納形式



備 考

a. $*$ は, 任意の値であることを示す.

b. NA $\geq$ N を満たさなければならない.

B.2.8 三角行列

(1) **2 次元配列型**

実対称行列 (2 次元配列型)(上三角型) または実対称行列 (2 次元配列型)(下三角型) と格納方法は同じである.

B.2.9　不規則スパース行列 (対称行列専用)

(1) **行方向 1 次元リスト型 (対称行列の場合)**

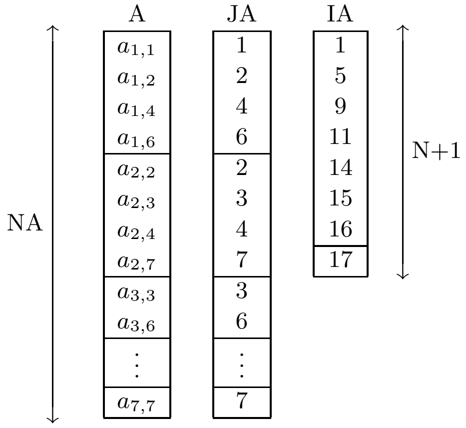
図 B−13　対称不規則スパース行列 (行方向 1 次元リスト型) の格納形式

格納する行列 A

$$
\begin{array}{|ccccccc|}
\hline
a_{1,1} & a_{1,2} & 0.0 & a_{1,4} & 0.0 & a_{1,6} & 0.0 \\
a_{1,2} & a_{2,2} & a_{2,3} & a_{2,4} & 0.0 & 0.0 & a_{2,7} \\
0.0 & a_{2,3} & a_{3,3} & 0.0 & 0.0 & a_{3,6} & 0.0 \\
a_{1,4} & a_{2,4} & 0.0 & a_{4,4} & a_{4,5} & a_{4,6} & 0.0 \\
0.0 & 0.0 & 0.0 & a_{4,5} & a_{5,5} & 0.0 & 0.0 \\
a_{1,6} & 0.0 & a_{3,6} & a_{4,6} & 0.0 & a_{6,6} & 0.0 \\
0.0 & a_{2,7} & 0.0 & 0.0 & 0.0 & 0.0 & a_{7,7} \\
\hline
\end{array}
$$

⇓

配列A(NA),JA(NA),IA(N+1)の格納状態



A	JA	IA
$a_{1,1}$	1	1
$a_{1,2}$	2	5
$a_{1,4}$	4	9
$a_{1,6}$	6	11
$a_{2,2}$	2	14
$a_{2,3}$	3	15
$a_{2,4}$	4	16
$a_{2,7}$	7	17
$a_{3,3}$	3	
$a_{3,6}$	6	
⋮	⋮	
$a_{7,7}$	7	

備　考

a.　N は, 行列 A の次数

b.　NA は, 非零上三角要素数

c.　A には, 行列 A の非零下三角要素を第 1 行から順番に格納する.

d.　JA には, A に格納した各要素の元の行列 A 上での列番号を格納する.

e.　IA には, 各対角要素の配列 A での位置を格納する. ただし, N+1 番目には NA+1 の値を格納する.

(2) **列方向 1 次元リスト型 (対称行列の場合)**

図 B−14　対称不規則スパース行列 (列方向 1 次元リスト型) の格納形式 (上三角部分を入力する場合)

格納する行列 A

$$\begin{array}{|ccccccc|}\hline a_{1,1} & a_{1,2} & 0.0 & a_{1,4} & 0.0 & a_{1,6} & 0.0 \\ a_{1,2} & a_{2,2} & a_{2,3} & a_{2,4} & 0.0 & 0.0 & a_{2,7} \\ 0.0 & a_{2,3} & a_{3,3} & 0.0 & 0.0 & a_{3,6} & 0.0 \\ a_{1,4} & a_{2,4} & 0.0 & a_{4,4} & a_{4,5} & a_{4,6} & 0.0 \\ 0.0 & 0.0 & 0.0 & a_{4,5} & a_{5,5} & 0.0 & 0.0 \\ a_{1,6} & 0.0 & a_{3,6} & a_{4,6} & 0.0 & a_{6,6} & 0.0 \\ 0.0 & a_{2,7} & 0.0 & 0.0 & 0.0 & 0.0 & a_{7,7} \\ \hline \end{array}$$

$$\Downarrow$$

配列 VALUES(NNZ), IRWIND(NNZ), IPONTR(N + 1) の格納状態

	VALUES	IRWIND	IPONTR	
NNZ	$a_{1,1}$	1	1	N+1
	$a_{1,2}$	1	2	
	$a_{2,2}$	2	4	
	$a_{2,3}$	2	6	
	$a_{3,3}$	3	9	
	$a_{1,4}$	1	11	
	$a_{2,4}$	2	15	
	$a_{4,4}$	4	17	
	$a_{4,5}$	4		
	$a_{5,5}$	5		
	$a_{1,6}$	1		
	$a_{3,6}$	3		
	$a_{4,6}$	4		
	$a_{6,6}$	6		
	$a_{2,7}$	2		
	$a_{7,7}$	7		

変数 ITYPE = $\boxed{2}$

備　考

a.　N は, 行列 A の次数.

b.　NNZ は, 行列 A の上三角部分の非零要素数.

c.　配列 VALUES には, 行列 A の上三角部分の非零要素を第 1 列から順番に格納する.

d.　配列 IRWIND には, VALUES に格納した各要素の行列 A 上での行番号を格納する.

e.　配列 IPONTR には, 行列 A の各列最初の要素を格納した配列 VALUES での位置を格納する.
ただし, IPONTR (1) には 1 を, IPONTR (N + 1) には NNZ + 1 を格納する.

図 B−15　対称不規則スパース行列 (列方向 1 次元リスト型) の格納形式 (下三角部分を入力する場合)

格納する行列 A

$a_{1,1}$	$a_{2,1}$	0.0	$a_{4,1}$	0.0	$a_{6,1}$	0.0
$a_{2,1}$	$a_{2,2}$	$a_{3,2}$	$a_{4,2}$	0.0	0.0	$a_{7,2}$
0.0	$a_{3,2}$	$a_{3,3}$	0.0	0.0	$a_{6,3}$	0.0
$a_{4,1}$	$a_{4,2}$	0.0	$a_{4,4}$	$a_{5,4}$	$a_{6,4}$	0.0
0.0	0.0	0.0	$a_{5,4}$	$a_{5,5}$	0.0	0.0
$a_{6,1}$	0.0	$a_{6,3}$	$a_{6,4}$	0.0	$a_{6,6}$	0.0
0.0	$a_{7,2}$	0.0	0.0	0.0	0.0	$a_{7,7}$

⇓

配列 VALUES(NNZ), IRWIND(NNZ), IPONTR(N + 1) の格納状態

VALUES	IRWIND	IPONTR
$a_{1,1}$	1	1
$a_{2,1}$	2	5
$a_{4,1}$	4	9
$a_{6,1}$	6	11
$a_{2,2}$	2	14
$a_{3,2}$	3	15
$a_{4,2}$	4	16
$a_{7,2}$	7	17
$a_{3,3}$	3	
$a_{6,3}$	6	
$a_{4,4}$	4	
$a_{5,4}$	5	
$a_{6,4}$	6	
$a_{5,5}$	5	
$a_{6,6}$	6	
$a_{7,7}$	7	

NNZ

N+1

変数 ITYPE = $\boxed{2}$

備　考

a.　N は, 行列 A の次数.

b.　NNZ は, 行列 A の下三角部分の非零要素数.

c.　配列 VALUES には, 行列 A の下三角部分の非零要素を第 1 列から順番に格納する.

d.　配列 IRWIND には, VALUES に格納した各要素の行列 A 上での行番号を格納する.

e.　配列 IPONTR には, 行列 A の各列最初の要素を格納した配列 VALUES での位置を格納する.
ただし, IPONTR (1) には 1 を, IPONTR (N + 1) には NNZ + 1 を格納する.

B.2.10　不規則スパース行列

(1) ELLPACK 型

図 B−16　非対称不規則スパース行列 (ELLPACK 型) の格納形式

格納する行列 A

$$\begin{array}{|ccccccc|}\hline
a_{1,1} & a_{1,2} & 0.0 & a_{1,4} & 0.0 & a_{1,6} & 0.0 \\
a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & 0.0 & 0.0 & a_{2,7} \\
0.0 & a_{3,2} & a_{3,3} & 0.0 & 0.0 & a_{3,6} & 0.0 \\
a_{4,1} & a_{4,2} & 0.0 & a_{4,4} & a_{4,5} & a_{4,6} & 0.0 \\
0.0 & 0.0 & 0.0 & a_{5,4} & a_{5,5} & 0.0 & 0.0 \\
a_{6,1} & 0.0 & a_{6,3} & a_{6,4} & 0.0 & a_{6,6} & a_{6,7} \\
0.0 & a_{7,2} & 0.0 & 0.0 & 0.0 & 0.0 & a_{7,7} \\ \hline
\end{array}$$

$$\Downarrow$$

配列 A(LNA, M), JA(LNA, M) の格納状態

A

$a_{1,1}$	$a_{1,2}$	$a_{1,6}$	$a_{1,4}$	$*$
$a_{2,2}$	$a_{2,1}$	$a_{2,3}$	$a_{2,4}$	$a_{2,7}$
$a_{3,3}$	$a_{3,2}$	$a_{3,6}$	$*$	$*$
⋮	⋮	⋮	⋮	⋮
$a_{6,6}$	$a_{6,1}$	$a_{6,3}$	$a_{6,4}$	$a_{6,7}$
$a_{7,7}$	$a_{7,2}$	$*$	$*$	$*$
$<$ − − − −		M		− − − − $>$

LNA

JA

1	2	6	4	0
2	1	3	4	7
3	2	6	0	$*$
⋮	⋮	⋮	⋮	⋮
6	1	3	4	7
7	2	0	$*$	$*$
$<$ − − − −		M		− − − − $>$

N

備　考

a.　N は, 行列 A の次数.

b.　LNA $\geq$ N を満たさなければならない.

c.　M は, 行列 A の非零要素を格納する配列 A の列数.

d.　配列 A には, 行列 A の非零要素を次のように格納する.
- 第 1 列に対角要素を格納する.
- 第 2〜M 列には下三角部分および上三角部分の非零要素をつめて各行ごとに格納する. ここで, 各行の非零要素を格納する順序は, 順不同である.
- 残りの部分の $*$ となっている位置に対応する要素は任意の値でよい.

e.　配列 JA には, 配列 A に格納した各要素に対応する箇所に行列 A 上での列番号を格納する.
M − 1 が行内の下三角部分および下三角部分の非零要素数より大きくなるような行については, 行列 A の非零要素の列番号を詰めた JA 内の領域の最右端の右隣の位置には 0 を格納する.
残りの部分の $*$ となっている位置には任意の値を格納する.

(2) 列方向 1 次元リスト型

図 B−17　実非対称不規則スパース行列 (列方向 1 次元リスト型) の格納形式

格納する行列 A

$$\begin{array}{|ccccccc|}\hline a_{1,1} & a_{1,2} & 0.0 & a_{1,4} & 0.0 & a_{1,6} & 0.0 \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & 0.0 & 0.0 & a_{2,7} \\ 0.0 & a_{3,2} & a_{3,3} & 0.0 & 0.0 & a_{3,6} & 0.0 \\ a_{4,1} & a_{4,2} & 0.0 & a_{4,4} & a_{4,5} & a_{4,6} & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & a_{5,5} & 0.0 & 0.0 \\ a_{6,1} & 0.0 & a_{6,3} & a_{6,4} & 0.0 & a_{6,6} & a_{6,7} \\ 0.0 & a_{7,2} & 0.0 & 0.0 & 0.0 & 0.0 & a_{7,7} \\ \hline \end{array}$$

$$\Downarrow$$

配列 VALUES(NNZ), IRWIND(NNZ), IPONTR(N + 1) の格納状態

VALUES	IRWIND	IPONTR
$a_{1,1}$	1	1
$a_{2,1}$	2	5
$a_{4,1}$	4	10
$a_{6,1}$	6	13
$a_{1,2}$	1	17
$a_{2,2}$	2	19
$a_{3,2}$	3	23
$a_{4,2}$	4	26
$a_{7,2}$	7	
$a_{2,3}$	2	
$a_{3,3}$	3	
$a_{6,3}$	6	
$a_{1,4}$	1	
$a_{2,4}$	2	
$a_{4,4}$	4	
$a_{6,4}$	6	
$a_{4,5}$	4	
$a_{5,5}$	5	
$a_{1,6}$	1	
$a_{3,6}$	3	
$a_{4,6}$	4	
$a_{6,6}$	6	
$a_{2,7}$	2	
$a_{6,7}$	6	
$a_{7,7}$	7	

NNZ

N+1

変数 ITYPE = $\boxed{1}$

備　考

a.　N は, 行列 A の次数.

b.　NNZ は, 行列 A の非零要素数.

c.　配列 VALUES には, 行列 A の非零要素を第 1 列から順番に格納する.

d.　配列 IRWIND には, 配列 VALUES に格納した各要素の行列 A 上での行番号を格納する.

e.　配列 IPONTR には, 行列 A の各列最初の要素を格納した配列 VALUES での位置を格納する.
ただし, IPONTR (1) には 1 を, IPONTR (N + 1) には NNZ + 1 を格納する.

(3) JAD 格納型

図 B–18　JAD 格納型の格納形式

格納する行列 A

$$\begin{array}{|ccccccc|}\hline
a_{1,1} & a_{1,2} & 0.0 & a_{1,4} & 0.0 & a_{1,6} & 0.0 \\
a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & 0.0 & 0.0 & a_{2,7} \\
0.0 & a_{3,2} & a_{3,3} & 0.0 & 0.0 & a_{3,6} & 0.0 \\
a_{4,1} & a_{4,2} & 0.0 & a_{4,4} & a_{4,5} & a_{4,6} & 0.0 \\
0.0 & 0.0 & 0.0 & a_{5,4} & a_{5,5} & 0.0 & 0.0 \\
a_{6,1} & 0.0 & a_{6,3} & a_{6,4} & 0.0 & a_{6,6} & 0.0 \\
0.0 & a_{7,2} & 0.0 & 0.0 & 0.0 & 0.0 & a_{7,7} \\ \hline
\end{array}$$

⇓

配列 JADORD の格納状態

3	N
1	
5	
2	
6	
4	
7	

N

7

MJAD

5

配列 AJAD の格納状態

$\leftarrow - - - \mathrm{MJAD} - - - \rightarrow$

$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,7}$
$a_{4,1}$	$a_{4,2}$	$a_{4,4}$	$a_{4,5}$	$a_{4,6}$
$a_{1,1}$	$a_{1,2}$	$a_{1,4}$	$a_{1,6}$	
$a_{6,1}$	$a_{6,3}$	$a_{6,4}$	$a_{6,6}$	
$a_{3,2}$	$a_{3,3}$	$a_{3,6}$		
$a_{5,4}$	$a_{5,5}$			
$a_{7,2}$	$a_{7,7}$			

配列 JAJAD の格納状態

$\leftarrow - \mathrm{MJAD} - \rightarrow$

1	2	3	4	7
1	2	4	5	6
1	2	4	6	
1	3	4	6	
2	3	6		
4	5			
2	7			

配列 IAJAD の格納状態

$\leftarrow - (\mathrm{MJAD} + 1) - \rightarrow$

1	8	15	20	24	26

備　考

a.　行列の要素のデータ型は, 実数でも複素数でもかまわない.

b.　N は, 行列 A の次数

c.　行列 A の零でない全ての要素を行ごとに左方向に詰め, 零でない要素の数について行を降順にソートしたときにできるデータ配置を考える. このデータ配置における各列を **jagged diagonal** という. MJAD には, jagged diagonal の本数を格納する. このデータ配置において jagged diagonal に沿って左端列から右端列に向かって連続した順番に要素を取り出し, 配列 AJAD に格納する.

d.　配列 JAJAD には, 配列 AJAD に格納した各要素に対応する行番号を格納する.

e.　配列 IAJAD には, 配列 AJAD における各 jagged diagonal の先頭位置を格納する. ただし, MJAD+1 番目には AJAD に格納される要素数に 1 を足した値を格納する.

f.　IAJAD (1) には値 1 が設定される.

g.　(JAD 格納型において配列 AJAD, JAJAD に格納される要素の数) = IAJAD (MJAD+1) −1

h.　上の例は, 行列 A が構造対称である場合について図示したものであるが, A が一般の非対称行列である場合についても JAD 格納形式で扱うことができる.

(4) **行方向 1 次元ブロックリスト型**

図 B−19　行方向 1 次元ブロックリスト型の格納形式 (M = 2 の場合)

格納する行列 A

B	0	C	0
D	F	0	G
0	0	H	0
0	K	0	L

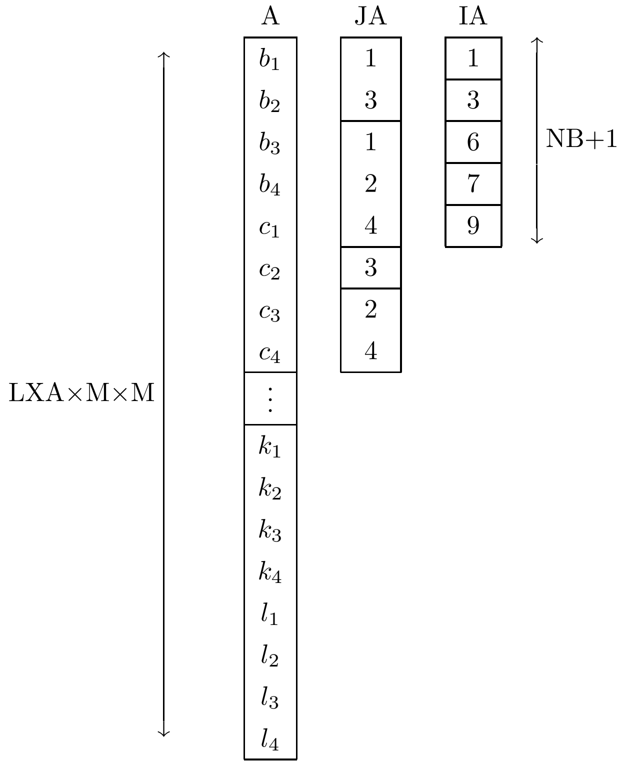
$$B=\begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \end{bmatrix},\quad C=\begin{bmatrix} c_1 & c_2 \\ c_3 & c_4 \end{bmatrix},\quad D=\begin{bmatrix} d_1 & d_2 \\ d_3 & d_4 \end{bmatrix},$$

$$F=\begin{bmatrix} f_1 & f_2 \\ f_3 & f_4 \end{bmatrix},\quad G=\begin{bmatrix} g_1 & g_2 \\ g_3 & g_4 \end{bmatrix},\quad H=\begin{bmatrix} h_1 & h_2 \\ h_3 & h_4 \end{bmatrix},$$

$$K=\begin{bmatrix} k_1 & k_2 \\ k_3 & k_4 \end{bmatrix},\quad L=\begin{bmatrix} l_1 & l_2 \\ l_3 & l_4 \end{bmatrix},\quad 0=\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\Downarrow$$

配列A(LXA×M×M), JA(LXA), IA(NB+1)の格納状態



備　考

a.　LXA は, 非零のブロック行列の数

b.　A には, 行列 A の非零のブロック行列をブロック行の第 1 行から順番に格納する. ブロック行列の要素を格納する順番は, ブロック行列の第 1 行から順番に行う.

c.　JA には, A に格納した各ブロック行列の元の行列 A 上でのブロック列番号を格納する.

d.　IA には, 行列 A の各ブロック行の最初の非零のブロック行列での, A に格納したブロック行列の個数の合計を格納する. ただし, NB+1 番目には LXA+1 の値を格納する.

(5) **MJAD 格納型**

図 B−20　MJAD 格納型の格納形式 (M = 2 の場合)

格納する行列 A

$$\begin{array}{|cccc|}\hline B & 0 & C & 0 \\ D & F & 0 & G \\ 0 & 0 & H & 0 \\ 0 & K & 0 & L \\ \hline\end{array}$$

$$B=\begin{bmatrix} b_1 & b_2 \\ b_3 & b_2 \end{bmatrix},\quad C=\begin{bmatrix} c_1 & c_2 \\ c_3 & c_4 \end{bmatrix},\quad D=\begin{bmatrix} d_1 & d_2 \\ d_3 & d_4 \end{bmatrix},\quad F=\begin{bmatrix} f_1 & f_2 \\ f_3 & f_4 \end{bmatrix},\quad G=\overset{\leftarrow -\mathrm{M}- \rightarrow}{\begin{bmatrix} g_1 & g_2 \\ g_3 & g_4 \end{bmatrix}},\ \updownarrow \mathrm{M}=2$$

$$H=\begin{bmatrix} h_1 & h_2 \\ h_3 & h_4 \end{bmatrix},\quad K=\begin{bmatrix} k_1 & k_2 \\ k_3 & k_4 \end{bmatrix},\quad L=\begin{bmatrix} l_1 & l_2 \\ l_3 & l_4 \end{bmatrix},\quad 0=\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\Downarrow$$

NB

4

MJAD

3

配列 JADORD の格納状態

↑ NB ↓

3
1
4
2

配列 IAJAD の格納状態

MJAD+1

← − − − − − − →

1	5	8	9

(⋆) 配列 AJAD の格納状態

← −M × M− →

↑ LXA ↓

d_1	d_2	d_3	d_4
k_1	k_2	k_3	k_4
b_1	b_2	b_3	b_4
h_1	h_2	h_3	h_4
f_1	f_2	f_3	f_4
l_1	l_2	l_3	l_4
c_1	c_2	c_3	c_4
g_1	g_2	g_3	g_4

配列 JAJAD の格納状態

← MJAD →

1	2	4
2	4	
1	3	
3		

備　考

a.　行列の要素のデータ型は, 実数でも複素数でもかまわない.

b.　NB は, 行列 A を M×M のブロック行列で区分けした場合の行のブロック数 (または列のブロック数)

c.　行列 A の零行列でない全てのブロック行列を行ごとに左方向に詰め, 零行列でないブロック行列の数について行を降順にソートしたときにできるデータ配置を考える. このデータ配置における各ブロック列を **jagged diagonal** という. MJAD には, jagged diagonal の本数を格納する. 配列 AJAD に格納する方法は次の通り. まず, 各ブロック行列 (D, K, B, H, F, C, G) から 1 行 1 列の要素を取り出す. ここで, 各ブロック行列から 1 行 1 列の要素を取り出す順番は, 上述の jagged diagonal に沿ってブロック行列が出てくる順番とする $(d_1, k_1, b_1, h_1, f_1, c_1, g_1)$. これを各ブロック行列の M 行 M 列の要素まで繰り返し, 取り出した要素を配列 AJAD に格納する.

d.　配列 JAJAD には, 配列 AJAD に格納した各ブロック行列に対応するブロック列番号を格納する.

e.　配列 IAJAD には, 配列 AJAD における各 jagged diagonal の先頭位置を格納する. ただし, MJAD+1 番目には AJAD に格納されるブロック行列数に 1 を足した値を格納する.

f.　IAJAD (1)　には値 1 が設定される.

g.　(MJAD 格納形式において格納される要素の数) = (IAJAD (MJAD+1) −1)×M×M

h.　同じブロックにある各要素のメモリ上の位置は, LXA 個の飛びを持つ (⋆). 例えば, 同じブロック D にある要素 (d_1,d_2,d_3,d_4) のメモリ上の位置は, LXA 個の飛びを持つ.

B.2.11　エルミートスパース行列 (エルミート行方向 1 次元リスト型) (上三角型)

図 B−21　エルミートスパース行列 (エルミート行方向 1 次元リスト型) (上三角型) の格納形式

格納する行列 A

$$\begin{array}{|ccccccc|}
\hline
a_{1,1} & a_{1,2} & 0. & a_{1,4} & 0. & a_{1,6} & 0. \\
\overline{a_{1,2}} & a_{2,2} & a_{2,3} & a_{2,4} & 0. & 0. & a_{2,7} \\
0. & \overline{a_{2,3}} & a_{3,3} & 0. & 0. & a_{3,6} & 0. \\
\overline{a_{1,4}} & \overline{a_{2,4}} & 0. & a_{4,4} & a_{4,5} & a_{4,6} & 0. \\
0. & 0. & 0. & \overline{a_{4,5}} & a_{5,5} & 0. & 0. \\
\overline{a_{1,6}} & 0. & \overline{a_{3,6}} & \overline{a_{4,6}} & 0. & a_{6,6} & 0. \\
0. & \overline{a_{2,7}} & 0. & 0. & 0. & 0. & a_{7,7} \\
\hline
\end{array}$$

$\Downarrow$

N	NA
7	16

配列 A(NA), JA(NA), IA(N+1) の格納状態

A	JA	IA
$a_{1,1}$	1	1
$a_{1,2}$	2	5
$a_{1,4}$	4	9
$a_{1,6}$	6	11
$a_{2,2}$	2	14
$a_{2,3}$	3	15
$a_{2,4}$	4	16
$a_{2,7}$	7	17
$a_{3,3}$	3	
$a_{3,6}$	6	
$a_{4,4}$	4	
$a_{4,5}$	5	
$a_{4,6}$	6	
$a_{5,5}$	5	
$a_{6,6}$	6	
$a_{7,7}$	7	

NA

N+1

備　考

a.　x の複素共役を $\overline{x}$ で表している.

b.　N は, 行列 A の次数

c.　NA は, 行列 A の対角成分および零でない上三角成分の個数

d.　配列 A には, 行列 A の対角成分および零でない上三角成分を第 1 行から順番に行方向に詰めて格納する.

e.　配列 JA には, 配列 A に格納した各要素の元の行列 A 上での列番号を格納する.

f.　配列 IA には, 各対角成分の配列 A での位置を格納する. ただし, N+1 番目には NA+1 の値を格納する.

g.　$1 \leq \mathrm{N} \leq \mathrm{NA}$ を満たさなければならない.

付 録 C　ASL で使用している計算機依存定数

C.1　誤差判定のための単位

ASL では, 浮動小数点演算における誤差判定のための単位として次の値を設定している. 誤差判定のための単位は, 浮動小数点データの内部表現によって決まる数値であり, ASL ではこの単位を収束判定, 零判定などに用いることがある.

表 C–1　誤差判定のための単位

単精度演算	倍精度演算
$2^{-23}(\simeq 1.19 \times 10^{-7})$	$2^{-52}(\simeq 2.22 \times 10^{-16})$

備考 誤差判定の単位 ε はマシン ε と呼ばれることもあり, 通常, 対応する浮動小数点形式で $1+\varepsilon$ の計算結果が 1 と異なるような最小の正の定数として定義される. したがって, 誤差判定の単位を見れば, その浮動小数点形式での (仮数部の) 演算の最大有効桁数がわかる.

C.2　浮動小数点データの値の最大値・最小値

ASL の内部で定義している浮動小数点データの値の最大値, 最小値を以下に示す.
なお, 以下の最大値, 最小値はハードウェアが実際に採用している浮動小数点形式のそれとは異なる場合があるので注意されたい.

表 C–2　浮動小数点データの値の最大値・最小値

	単精度演算	倍精度演算
最大値	$2^{127}(2-2^{-23})$ $(\simeq 3.40 \times 10^{38})$	$2^{1023}(2-2^{-52})$ $(\simeq 1.80 \times 10^{308})$
正の最小値	2^{-126} $(\simeq 1.17 \times 10^{-38})$	2^{-1022} $(\simeq 2.23 \times 10^{-308})$
負の最大値	-2^{-126} $(\simeq -1.17 \times 10^{-38})$	-2^{-1022} $(\simeq -2.23 \times 10^{-308})$
最小値	$-2^{127}(2-2^{-23})$ $(\simeq -3.40 \times 10^{38})$	$-2^{1023}(2-2^{-52})$ $(\simeq -1.80 \times 10^{308})$

索引

アプリケーションシステム

科学技術計算ライブラリ

ASL **ユーザーズガイド**

〈 基本機能編 第 1 分冊 〉

2023 年 3 月 ASL (1.1)
付属説明書 3.0.0-230301

日 本 電 気 株 式 会 社