JAERI-Data/Code
99-016

JP9950284

誤差評価ライブラリ（Ver.1.0）
理論説明書／利用手引書

1999年3月

市原　潔*・志澤由久*・岸田則生

日本原子力研究所
Japan Atomic Energy Research Institute

編集兼発行　日本原子力研究所

誤差評価ライブラリ（Ver.1.0）理論説明書／利用手引書

日本原子力研究所計算科学技術推進センター
市原　潔*・志澤　由久**・岸田　則生

（1999年2月9日受理）

「誤差評価ライブラリ」(以下、「本ライブラリ」) は、行列計算の数値解析結果の誤差の分析を支援するためのサブルーチン集である。連立一次方程式誤差評価ルーチン群、エルミート行列固有値問題誤差評価ルーチン群、試験行列生成ルーチン群によって構成される。一部のサブルーチンは、分散メモリ型並列計算機のMPI環境で動作可能である。

連立一次方程式誤差評価ルーチンは、残差ベクトルのノルム、行列の条件数、および誤差限界を計算する。エルミート行列固有値問題誤差評価ルーチンは、Korn-Katoの公式に基づいて固有値の存在範囲を計算する。試験行列生成ルーチンは、数学研究に基づく試験行列、乱数行列、アプリケーション・プログラムに現れる典型的な行列を生成する。

本書は、本ライブラリで使用している精度評価公式および試験行列の要点とサブルーチンの利用方法についてまとめたものである。

日本原子力研究所：〒153-0061　東京都目黒区中目黒2-2-54
*　株式会社 三菱総合研究所
**　株式会社 情報数理研究所

Subroutine Library for Error Estimation of Matrix Computation (Ver.1.0)

Kiyoshi ICHIHARA*, Yoshihisa SHIZAWA** and Norio KISHIDA

Center for Promotion of Computational Science and Engineering
Japan Atomic Energy Research Institute
Nakameguro, Meguro-ku, Tokyo

(Received February 9, 1999)

"Subroutine Library for Error Estimation of Matrix Computation" is a subroutine library which aids the users in obtaining the error ranges of the linear system's solutions or the Hermitian matrices' eigen values. This library contains routines for both sequential computers and parallel computers.

The subroutines for linear system error estimation calulate norms of residual vectors, matrices's condition numbers, error bounds of solutions and so on. The subroutines for error estimation of Hermitian matrix eigen values derive the error ranges of the eigen values according to the Korn-Kato's formula. The test matrix generators supply the matirces appeared in the mathematical research, the ones randomly generated and the ones appeared in the application programs.

This user's manual contains a brief mathematical background of error analysis on linear algebra and usage of the subroutines.

Keywords: Error Analysis, Matrix Computation, Linear Equations, Eigen Value Problem, Test, Matrix, Parallel Computers, Numerical Analysis

---

* Mitsubishi Research Institute Inc.
** Information and Mathematical Science Laboratory Inc.

目　　次

Contents

Part I

プログラム設計仕様

1. 序論

このライブラリは、連立一次方程式を計算機で解いた結果の誤差評価を分散メモリ型並列計算機で行うために必要な機能を提供するプログラムライブラリである。

稼動環境は、分散メモリ型並列計算機であり、プロセッサ間通信は、国際標準MPI (Message Passing Interface) を用いている。

1.1 機能概要

このライブラリは、分散メモリ型並列計算機上で、行列のノルム及び条件数の計算、及びそれらを利用した近似誤差限界の評価を行う。

ノルムは1－ノルム及び∞－ノルムを使用する。また、条件数の計算方法は、定義どおりの計算及び近似計算が選択できる。

1.2 対象とする行列

処理対象とする行列は、連立一次方程式誤差評価では実数または複素数の密正方行列、固有値誤差評価では、実対称の密、帯、疎行列またはエルミート密行列である。試験行列では、種々の行列を扱っており、目次等参照されたい。

1.3 稼動環境

分散メモリ型並列計算機

1.4 使用言語

FORTRAN 77
プロセッサ間通信：MPI

1.5 開発環境

以下の環境で開発・試験を実施した。

- 日本原子力研究所 計算機科学技術推進センター所有IBM SP （IBM RS/6000 SP）, AIX Version 4

なお、複素数連立一次方程式誤差評価分割ローカルと実数対称行列固有値誤差評価分割ローカルについては、 IBM SP 以外に、CRAY T94, FUJITSU VPP300, HITACHI SR2201, NEC SX4 の４機種でも動作確認を行った。

1.6 利用方法

SPMD(Single Program Multiple Data モデル) で並列化されたプログラム内での利用を想定する。すなわち、各並列プロセス内で本ライブラリを呼び出す。

行列は利用者があらかじめ分割し、各PEのメモリ上に存在するものとする。行列の保持形態としては、以下の２方式をサポートする。

重複ローカル

各分散メモリが、同じ行列の全体を重複して持つ。（処理のみ分割して並行に行う。）

分割ローカル

各分散メモリが、行列の部分行列を分担して持つ。

なお、疎行列については、部分行列同士に重複を許している。

2. 誤差評価の理論

2.1 連立一次方程式の誤差評価

ノルムの定義、条件数の定義及び近似条件数の計算法、逆行列の計算法、近似誤差限界の評価方法について述べる。

2.1.1 ノルムの定義

まずノルムの定義を述べる。行列のノルムは、以下のように表される。

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|}$$

より具体的には、ノルムが1-ノルム、2-ノルム、∞-ノルムのいずれであるかに応じて、以下のように表される。

$$\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^{n} |a_{ij}|$$

$$\|A\|_2 = \sqrt{\max |\sigma(A^*A)|}$$

$$\|A\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^{n} |a_{ij}|$$

1-ノルム、∞-ノルムについては、

$$\|A\|_\infty = \|A^T\|_1$$

の関係がある。また、2-ノルムについては、$\sigma(A^*A)$ すなわち、行列積 A^*A の絶対値最大固有値を求め、次にその平方根を求める必要がある。なお、行列 A がエルミート行列（または実対称行列）の場合は、$\sqrt{\max |\sigma(A^*A)|} = |\sigma(A)|$ なので、以下のように簡略化される。

$$\|A\|_2 = \max |\sigma(A)|$$

2.1.2 条件数の定義及び計算法

次に条件数の定義を述べる。行列 A の条件数を $\kappa(A)$ で表すとすると、その定義は以下のように表される。

$$\kappa(A) = \|A\| \|A^{-1}\|$$

使用するノルムは、1-ノルム、2-ノルム、∞-ノルムのいずれでもよい。いずれのノルムにせよ、$\|A\|$ を計算することは、それほど難しいことではないが、問題は、$\|A^{-1}\|$ の計算である。以下に、1-ノルムの場合の条件数の近似計算法について概説を述べる。1-ノルムの計算法がわかれば、行列を転置することにより ∞-ノルムも容易に求めることができる。2-ノルムの場合の条件数の近似計算法については、村田[3]を参照されたい。

まず1-ノルムの近似計算法であるが、最初に Hager のアルゴリズムを述べ、ついで Higham による改良アルゴリズムを示す。いずれも詳細は Higham[4]を参照されたい。

Hager のアルゴリズムでは、以下の目的関数の最大値を求めることを目指す。

$$F(x) = \|Bx\|_1$$

ここで、

$$S = x \in R^n : \|x\|_1 \le 1$$

y が x に十分近く、かつ $F(x)$ が微分可能であるとき、

$$F(y) = F(x) + \nabla F(x)^T(y - x)$$

ここで、

$$|\nabla F(x)^T y| \le \|F(x)\|_\infty \|y\|_1 \le \|F(x)\|_\infty$$

であることから、$\|F(x)\|_\infty \le \nabla F(x)^T x$ なる x が、 S における F の極大値を与える。

なお、$(Bx)_i = 0$ となる i が存在する場合、 $F(x)$ は、微分係数を定義できない(絶対値関数の折り目の部分にあたる)ので、その場合は、

$$F(y) \ge F(x) + g^T(y - x)$$

を満たすベクトル g を探し出す。そのような g をサブグラディエント(subgradient)と呼ぶ。x における F のサブグラディエントの集合 ∂F は、ベクトル $B^T\xi$ であらわされ、ここで ξ は以下のように表される。

$$\xi = \begin{cases} 1 & \text{if } (Bx)_i > 0\ , \\ -1 & \text{if } (Bx)_i < 0\ , \\ \text{[-1,1] の任意の実数} & \text{if } (Bx)_i = 0 \end{cases}$$

ここで以下のベクトル関数を定義する。

$$sign(x) = (s_i), \quad s_i = \begin{cases} 1 & \text{if } x_i \ge 0\ , \\ -1 & \text{if } x_i < 0 \end{cases}$$

$B = A^{-1}$ の場合に $\gamma \le \|A^{-1}\|_1$ を評価する Hager のアルゴリズムを述べる。

(1) $\|x\|_1 = 1$ となる x を選ぶ。

(2) 以下を繰り返す。
$Ay = x$ を解く(A は予めLU分解されているものとする)。
$\xi := sign(y)$
$A^T z = \xi$ を解く。$\|z\|_\infty \le z^T x$ のとき、
$\gamma = \|y\|_1$ として終了。

Higham は Hager のアルゴリズムについて数値実験を行い、その結果から導かれるヒューリスティックな知見に基づいて、より実用的な解法へと改良した。その知見とは、以下のようなものである。

- 通常の（性質の悪くない）行列の場合は、２回程度、多くとも５回程度で収束する。
- 収束しない（性質の悪い）行列の場合でも、

$$b_i := (-1)^{i+1}\left(1+\frac{i-1}{n-1}\right),\quad i=1,\cdots,n,\quad \|b\|_1=\frac{3n}{2}$$

なる b について、$Ax=b$ を解き、$\frac{2\|x\|_1}{(3n)}$ を計算すると、それが A の1-ノルムのよい近似になっていることがある。

$\gamma \le \|B\|_1$ を評価する Higham の改良アルゴリズムを以下に述べる。得られるのは、$v = Bw$ であり、$\gamma = \|v\|_1/\|w\|_1$ が計算される。

(1) 以下の初期値を設定する。$v := B(n^{-1}e)$

もし、$n=1$ のとき、$\gamma := |v_1|$ として終了。

$\gamma := \|v\|_1$

$\xi := sign(v)$

$x := B^T\xi$

$k := 2$

(2) 以下を繰り返す。

$j := \min\{i : |x_i| = \|x\|_\infty\}$

$v := Be_j$

$\bar{\gamma} := \gamma$

$\gamma := \|v\|_1$

もし、$sign(v) = \xi$ または$\gamma \le \bar{\gamma}$ のとき、

(3) に行く。

$\xi := sign(v)$

$x := B^T\xi$

$k := k+1$

$\|x\|_\infty = x_j$ または $k=5$ のとき、

(3) に行く。

(3) $i=1,2,\cdots,n$ に対して、

$x_i := (-1)^{i+1}\left(1+\frac{i-1}{n-1}\right)$

$x := Bx$

もし、$2\|x\|_1/(3n) > \gamma$ のとき、

$v := x$

$\gamma := 2\|x\|_1/(3n)$

ここで、$x := By$ すなわち $x := A^{-1}y$ のような表現は、A^{-1} を明示的に求めて $A^{-1}y$ を計算しているわけではなく、A があらかじめLU分解されているものとして、$y = Ax$ を前進代入・後退代入によって解くことを意味する。また、$x := B^Ty$ は、$y = A^Tx$ を解くことを意味する。

2.1.3 逆行列の計算法

本ライブラリにおいて、逆行列の計算法はＬＵ分解に基づくアルゴリズムを採用した。具体的には以下のとおりである。

行列Ａをピボット選択しながらLU分解する。ここで、下三角行列Ｌの対角要素l_{ii}が１になるようにする。すると、Aの逆行列$C = A^{-1}$は、

$$CLU = I$$

となる。これより、

$$(CL)U = I \quad を \quad (CL) について解く。 \tag{1}$$

$$CL = U^{-1} \quad を \quad C について解く。 \tag{2}$$

式(1)より、対角線を含む上三角部分について、$\frac{n(n+1)}{2}$ 個の方程式を得る。これより CL が求まる。CL は、下三角部分がゼロの行列である。

また、式(2)より、n^2 個の方程式を得る。これより、C が求まる。

例えば、4次の行列の場合、式(1) $(CL)U = I$ は以下のようになる。ここで、CL のかわりに $B = \{b_{ij}\}$ とおいた。

$$\begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ & b_{22} & b_{23} & b_{24} \\ & & b_{33} & b_{34} \\ & & & b_{44} \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ & u_{22} & u_{23} & u_{24} \\ & & u_{33} & u_{34} \\ & & & u_{44} \end{pmatrix} = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix}$$

これより、以下のように順次方程式を解いていけばよい。

$$\begin{array}{lllllllllll}
b_{11}u_{11} & & & & & & & = & 1 & \to & b_{11} \text{ が 求まる。} \\
b_{11}u_{12} & + & b_{12}u_{22} & & & & & = & 0 & \to & b_{12} \text{ が 求まる。} \\
 & & b_{22}u_{22} & & & & & = & 1 & \to & b_{22} \text{ が 求まる。} \\
b_{11}u_{13} & + & b_{12}u_{23} & + & b_{13}u_{33} & & & = & 0 & \to & b_{13} \text{ が 求まる。} \\
 & & b_{22}u_{23} & + & b_{23}u_{33} & & & = & 0 & \to & b_{23} \text{ が 求まる。} \\
 & & & + & b_{33}u_{33} & & & = & 1 & \to & b_{33} \text{ が 求まる。} \\
b_{11}u_{14} & + & b_{12}u_{24} & + & b_{13}u_{34} & + & b_{14}u_{44} & = & 0 & \to & b_{14} \text{が 求まる。} \\
 & & b_{22}u_{24} & + & b_{23}u_{34} & + & b_{24}u_{44} & = & 0 & \to & b_{24} \text{が 求まる。} \\
 & & & & b_{33}u_{34} & + & b_{34}u_{44} & = & 0 & \to & b_{34} \text{が 求まる。} \\
 & & & & & & b_{44}u_{44} & = & 1 & \to & b_{44} \text{が 求まる。}
\end{array}$$

実際の計算において、B と U は同一の行列に格納する。すなわち、最初に U が格納されている行列において、上三角成分の左側の列から順次 B の成分で置き換えられていく。なお、下三角部分には L が格納

されているので、この部分は値を変更しないように注意する。

$$\begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ & u_{22} & u_{23} & u_{24} \\ & & u_{33} & u_{34} \\ & & & u_{44} \end{pmatrix} \rightarrow \begin{pmatrix} b_{11} & b_{12} & u_{13} & u_{14} \\ & b_{22} & u_{23} & u_{24} \\ & & u_{33} & u_{34} \\ & & & u_{44} \end{pmatrix} \rightarrow \begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ & b_{22} & b_{23} & b_{24} \\ & & b_{33} & b_{34} \\ & & & b_{44} \end{pmatrix}$$

こうして、B (すなわち U^{-1}) が求まったとして、式(2) $CL = B(= U^{-1})$ は以下のようになる。

$$\begin{pmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \\ c_{41} & c_{42} & c_{43} & c_{44} \end{pmatrix} \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & \\ l_{31} & l_{32} & 1 & \\ l_{41} & l_{42} & l_{43} & 1 \end{pmatrix} = \begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ & b_{22} & b_{23} & b_{24} \\ & & b_{33} & b_{34} \\ & & & b_{44} \end{pmatrix}$$

これについても、以下のように順次方程式を解いていけばよい。

$$\begin{array}{ccccccccccc}
 & & & & & & c_{14} & = & b_{14} & \rightarrow & c_{14}\text{が 求まる。} \\
 & & & & & & c_{24} & = & b_{24} & \rightarrow & c_{24}\text{が 求まる。} \\
 & & & & & & c_{34} & = & b_{34} & \rightarrow & c_{34}\text{が 求まる。} \\
 & & & & & & c_{44} & = & b_{44} & \rightarrow & c_{44}\text{が 求まる。} \\
 & & & & c_{13} & + & c_{14}l_{43} & = & b_{13} & \rightarrow & c_{13}\text{が 求まる。} \\
 & & & & c_{23} & + & c_{24}l_{43} & = & b_{23} & \rightarrow & c_{23}\text{が 求まる。} \\
 & & & & c_{33} & + & c_{34}l_{43} & = & b_{33} & \rightarrow & c_{33}\text{が 求まる。} \\
 & & & & c_{43} & + & c_{44}l_{43} & = & 0 & \rightarrow & c_{43}\text{が 求まる。} \\
 & & c_{12} & + & c_{13}l_{32} & + & c_{14}l_{42} & = & b_{12} & \rightarrow & c_{12}\text{が 求まる。} \\
 & & c_{22} & + & c_{23}l_{32} & + & c_{24}l_{42} & = & b_{22} & \rightarrow & c_{22}\text{が 求まる。} \\
 & & c_{32} & + & c_{33}l_{32} & + & c_{34}l_{42} & = & 0 & \rightarrow & c_{32}\text{が 求まる。} \\
 & & c_{42} & + & c_{43}l_{32} & + & c_{44}l_{42} & = & 0 & \rightarrow & c_{42}\text{が 求まる。} \\
a_{11} & + & c_{12}l_{21} & + & c_{13}l_{31} & + & c_{14}l_{41} & = & b_{11} & \rightarrow & c_{11}\text{が 求まる。} \\
a_{21} & + & c_{22}l_{21} & + & c_{23}l_{31} & + & c_{24}l_{41} & = & 0 & \rightarrow & c_{21}\text{が 求まる。} \\
a_{31} & + & c_{32}l_{21} & + & c_{33}l_{31} & + & c_{34}l_{41} & = & 0 & \rightarrow & c_{31}\text{が 求まる。} \\
a_{41} & + & c_{42}l_{21} & + & c_{43}l_{31} & + & c_{44}l_{41} & = & 0 & \rightarrow & c_{41}\text{が 求まる。}
\end{array}$$

実際の計算において、A と L と B は同一の行列に格納される。すなわち、最初に 下三角部分に L 、対角部分と上三角部分に B が格納されている行列において、右側の列から順次 C の成分で置き換えられていく。

$$\begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ l_{21} & b_{22} & b_{23} & b_{24} \\ l_{31} & l_{32} & b_{33} & b_{34} \\ l_{41} & l_{42} & l_{43} & b_{44} \end{pmatrix} \rightarrow \begin{pmatrix} b_{11} & b_{12} & c_{13} & c_{14} \\ l_{21} & b_{22} & c_{23} & c_{24} \\ l_{31} & l_{32} & c_{33} & c_{34} \\ l_{41} & l_{42} & c_{43} & c_{44} \end{pmatrix} \rightarrow \begin{pmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \\ c_{41} & c_{42} & c_{43} & c_{44} \end{pmatrix}$$

逆行列を得るためには、LU 分解時のピボット選択のインデックスに従って C の列を入れ替える必要があ

る（「行」ではなく「列」を入れ替えることに注意する)。なお、逆行列のノルム $\|A^{-1}\|$ のみが必要で、逆行列 A^{-1} そのものが不要であれば、この列の入れ替えは不要である。

2.1.4 連立一次方程式の近似誤差限界

本ライブラリでは、連立一次方程式について以下の2通りの誤差限界の値を与える。

公式1: 連立一次方程式の事前誤差解析 [1)]

$$\frac{\|x-\bar{x}\|}{\|x\|} \leq \kappa(A)\epsilon$$

公式2: 連立一次方程式の事後誤差解析 [2)]

$$\frac{\|x-\bar{x}\|}{\|x\|} \leq \kappa(A)\frac{\|A\bar{x}-b\|}{\|b\|}$$

ここで、

x : 真の解。

$\bar{x}$: 数値計算解。

$\kappa(A)$: 行列 A の条件数$= \|A\|\|A^{-1}\|$

ϵ : マシン ε 。

マシン ε は、ユーザが計算機マニュアルを参照するか、他ライブラリのルーチン(たとえば、LAPACKのdlamch() など) を用いるかして、求めておく必要がある。不明のときは、倍精度の場合、1.11×10^{-16}として試みられたい。

2.2 エルミート行列固有値問題の誤差解析

本節に述べる公式は、エルミート行列の固有値問題について成立するものである。したがって、実対称行列の場合にももちろん成立する。

最初に、Krylov-Weinstein の不等式、次に Kato-Temple の不等式、最後に、Korn-Kato の公式について述べる。本ツールに含まれるプログラムでは、Korn-Kato の公式を使用している。詳細は、シャトラン [5] を参照されたい。

2.2.1 Krylov-Weinstein の不等式

$||u||_2 = 1$ なる任意の u にたいして Rayleigh 商 $\rho \equiv (u, Au)$ をとる。このとき

$$|\lambda - \rho| \leq ||Au - \rho u||_2 \tag{3}$$

を満たす A の固有値 λ が存在する。[5] $(||x||_2 \equiv \sqrt{\sum_{i=1}^{n} |x_i|^2}.)$

例2.2.1-1) 行列 A を以下のように定める。

$$A = \begin{bmatrix} 1 & 10^{-5} & 10^{-5} \\ 10^{-5} & 2 & 10^{-5} \\ 10^{-5} & 10^{-5} & 3 \end{bmatrix}$$

u として、A の近似固有ベクトルにとる。

$$u = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

このとき

$$\rho = 1, \quad 2, \quad 3$$

であり、Krylov-Weinstein の不等式より、

$$|\lambda_i - i| \leq \sqrt{2} \cdot 10^{-5}. \qquad i = 1, 2, 3.$$

が成立する。

ρ と A の他の固有値との距離に関してさらにいくらかの情報が得られているときには Krylov-Weinstein の不等式の改良が可能である。次にこの改良である Kato-Temple の不等式について述べる。

2.2.2 Kato-Templeの不等式

開区間 $(\underline{\lambda},\overline{\lambda})$ にはHermite行列 A の固有値 λ が一個だけとρ とが含まれているということが知られているとする。このとき[5)]

$$\rho - \frac{\varepsilon^2}{\overline{\lambda}-\rho} \leq \lambda \leq \rho + \frac{\varepsilon^2}{\rho - \underline{\lambda}}, \tag{4}$$

および

$$\sin\theta \leq \frac{2}{\overline{\lambda}-\underline{\lambda}}\left[\left(\rho - \frac{\underline{\lambda}+\overline{\lambda}}{2}\right)^2 + \varepsilon^2\right]^{1/2} \tag{5}$$

が成り立つ。ここで

$$\varepsilon \equiv ||Au - \rho u||_2 \tag{6}$$

であり、θはベクトルuの方向の直線とλに付随する固有部分空間とで挟まれる鋭角である。

不等式(4)はKato-Templeの不等式という名で知られている。$\varepsilon^2 < (\overline{\lambda}-\rho)(\rho-\underline{\lambda})$ のとき、Krylov-Weinsteinの不等式の改良となっている。

例2.2.2-1）再び、行列Aとして

$$A = \begin{bmatrix} 1 & 10^{-5} & 10^{-5} \\ 10^{-5} & 2 & 10^{-5} \\ 10^{-5} & 10^{-5} & 3 \end{bmatrix},$$

u として A の近似固有ベクトル

$$u = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

の場合を考察する。

このときKato-Templeの不等式より、

$$\begin{aligned} 1 - \frac{2\cdot 10^{-10}}{1-\sqrt{2}\cdot 10^{-5}} \leq \quad & \lambda_1 && \leq 1, \\ & |\lambda_2 - 2| && \leq \frac{2\cdot 10^{-10}}{1-\sqrt{2}\cdot 10^{-5}}, \\ 3 \leq \quad & \lambda_3 && \leq 3 + \frac{2\cdot 10^{-10}}{1-\sqrt{2}\cdot 10^{-5}} \end{aligned}$$

が導かれる。

2.2.3 Korn-Katoの公式

Korn-Katoの公式[11)]を示す。

エルミート行列 A の真の固有値を λ_i とし、対応する真の固有ベクトルを x_i とする。また、固有値は実数であり、以下の様に順序づけられているとする。

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n \tag{7}$$

計算によって得られた固有ベクトルを $\hat{x}_i$ とする。

このとき λ_k の存在範囲は以下の式で与えられる。[Korn-Kato の公式]

$$\hat{\lambda}_k^{(R)} - \frac{(\hat{\lambda}_k^{(S)})^2 - (\hat{\lambda}_k^{(R)})^2}{\lambda_{k+1} - \hat{\lambda}_k^{(R)}} \leq \lambda_k \leq \hat{\lambda}_k^{(R)} + \frac{(\hat{\lambda}_k^{(S)})^2 - (\hat{\lambda}_k^{(R)})^2}{\hat{\lambda}_k^{(R)} - \lambda_{k-1}} \tag{8}$$

ここで、$\hat{\lambda}_k^{(R)}$、$\hat{\lambda}_k^{(S)}$ は、以下のように定義される。

$$\hat{\lambda}_k^{(R)} = \frac{(\hat{x}_k, A\hat{x}_k)}{(\hat{x}_k, \hat{x}_k)} \tag{9}$$

$$\hat{\lambda}_k^{(S)} = \sqrt{\frac{(A\hat{x}_k, A\hat{x}_k)}{(\hat{x}_k, \hat{x}_k)}} \tag{10}$$

式(8)において、$\lambda_{k-1}, \lambda_{k+1}$ の真の値が不明であるので計算値（近似値）で代用する。

補足 Korn-Katoの公式はKato-Templeの不等式において

$$\begin{aligned}
\underline{\lambda} &\longrightarrow \lambda_{k-1}, \\
\overline{\lambda} &\longrightarrow \lambda_{k+1}, \\
\lambda &\longrightarrow \lambda_k, \\
\rho &\longrightarrow \hat{\lambda}_k^{(R)}, \\
\varepsilon &\longrightarrow (\hat{\lambda}_k^{(S)})^2 - (\hat{\lambda}_k^{(R)})^2
\end{aligned} \tag{11}$$

としたものに対応している。

3.　試験行列：数学研究に現れる行列

3.1　Hilbert 行列

行列成分が以下のように表される。

$$A_{i,j} = \frac{1}{i+j-1}$$

例えば、5 × 5 行列の場合は、以下のようになる。

$$\begin{pmatrix} 1 & 1/2 & 1/3 & 1/4 & 1/5 \\ 1/2 & 1/3 & 1/4 & 1/5 & 1/6 \\ 1/3 & 1/4 & 1/5 & 1/6 & 1/7 \\ 1/4 & 1/5 & 1/6 & 1/7 & 1/8 \\ 1/5 & 1/6 & 1/7 & 1/8 & 1/9 \end{pmatrix}$$

Hilbert 行列は、次数の増大とともに条件数が極めて大きくなる（悪条件になる）ことが、知られている。また、逆行列の厳密解が明示的に式表現できる。

$$A^{-1}_{i,j} = \frac{(-1)^{i+j}(n+i-1)!(n+j-1)!}{(i+j-1)[(i-1)!(j-1)!]^2(n-i)!(n-j)!}$$

したがって、Hilbert 行列を逆行列解法の数値誤差に関する感度分析に用いることができる。

3.2 Pascal 行列

Pascal 行列と呼ばれるものには、以下の２種類がある。

- 対角成分と下三角行列が Pascal の三角形を形成し、上三角成分がゼロのもの P
- 上記 P とその転置行列 P^t を用いて、$Q = PP^t$ で表される対称行列

3.2.1 下三角行列が Pascal の三角形を形成する P

行列成分が以下のように表される。

$$P_{i,j} = \begin{cases} {}_{i-1}C_{j-1} & (i \geq j) \\ 0 & (i < j) \end{cases}$$

例えば、５×５行列の場合は、以下のようになる。

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 \\ 1 & 3 & 3 & 1 & 0 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

P の逆行列は、以下のように表される。

$$P^{-1}_{i,j} = \begin{cases} (-1)^{i+j}{}_{i-1}C_{j-1} & (i \geq j) \\ 0 & (i < j) \end{cases}$$

例えば、５×５行列の場合は、以下のようになる。

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ -1 & 3 & -3 & 1 & 0 \\ 1 & -4 & 6 & -4 & 1 \end{pmatrix}$$

3.2.2 $Q = PP^t$ で表される対称行列

行列成分が以下のように表される。

$$Q_{i,j} = (PP^t)_{i,j} = \frac{(i+j-2)!}{(i-1)!(j-1)!}$$

例えば、5×5行列の場合は、以下のようになる。

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 3 & 6 & 10 & 15 \\ 1 & 4 & 10 & 20 & 35 \\ 1 & 5 & 15 & 35 & 70 \end{pmatrix}$$

Qの逆行列は、上記Pを用いて以下のように計算できる。

$$Q^{-1} = (PP^t)^{-1} = {P^{-1}}^t P^{-1}$$

Pascal行列は、次数の増大とともに条件数が大きくなる。しかし、その程度はHilbert行列ほどではない。また、逆行列の厳密解が明示的に式表現でき、逆行列の要素は全て整数になることが知られている。

3.3 Frank行列

行列成分が以下のように表される（nは、行列の次数）。

$$A_{i,j} = n + 1 - \max(i, j)$$

例えば、5×5行列の場合は、以下のようになる。

$$\begin{pmatrix} 5 & 4 & 3 & 2 & 1 \\ 4 & 4 & 3 & 2 & 1 \\ 3 & 3 & 3 & 2 & 1 \\ 2 & 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Frank行列は、固有値の厳密解が以下のように表現できることが知られている。

$$\lambda_i = \frac{1}{2}\left(1 - \cos\frac{(2i-1)\pi}{2n+1}\right)^{-1}$$

Frank行列は、次数の増大とともに悪条件になるが、その度合いは、Hilbert行列ほどひどくはない。固有値の絶対値が小さくなるにつれて密集する性質があるので、固有値解法のプログラムのチェックに適している。

3.4 Pei 行列

行列成分が以下のように表される。

$$A_{i,j} = \begin{cases} d & (\text{対角成分}) \\ 1 & (\text{非対角成分}) \end{cases}$$

例えば、５×５行列の場合は、以下のようになる。

$$\begin{pmatrix} d & 1 & 1 & 1 & 1 \\ 1 & d & 1 & 1 & 1 \\ 1 & 1 & d & 1 & 1 \\ 1 & 1 & 1 & d & 1 \\ 1 & 1 & 1 & 1 & d \end{pmatrix}$$

d は１より大きな実数であり、１に近くなるほど悪条件になる。つまり、条件数が行列の次元以外に、d の値によっても大きく変化する。

Pei 行列の逆行列の厳密解は、以下のように表現できる。

$$A_{i,j}^{-1} = \begin{cases} \dfrac{d+n-2}{d(d+n-2)-(n-1)} & (i=j) \\ \\ \dfrac{-1}{d(d+n-2)-(n-1)} & (i \neq j) \end{cases}$$

Pei 行列の固有値の厳密解は、以下のように表現できる。

$$\begin{aligned} \lambda_i &= d-1 \qquad (i=1,2,\ldots,n-1) \\ \lambda_n &= d-1+n \end{aligned}$$

つまり、Pei 行列は、固有値がｎ－１重に縮退している。縮退している固有値に対応する固有ベクトルの直行性をチェックするのに適している。

3.5 Lehmer 行列

行列成分が以下のように表される。

$$A_{i,j} = \begin{cases} i \;/\; j & (i \le j) \\ j \;/\; i & (i > j) \end{cases}$$

例えば、5×5行列の場合は、以下のようになる。

$$\begin{pmatrix} 1 & 1/2 & 1/3 & 1/4 & 1/5 \\ 1/2 & 1 & 2/3 & 2/4 & 2/5 \\ 1/3 & 2/3 & 1 & 3/4 & 3/5 \\ 1/4 & 2/4 & 3/4 & 1 & 4/5 \\ 1/5 & 2/5 & 3/5 & 4/5 & 1 \end{pmatrix}$$

Lehmer 行列の逆行列の厳密解は、以下のように表現できる。

$$A^{-1}_{i,j} = \begin{cases} \dfrac{4i^3}{4i^2-1} & (i = j < n) \\[2ex] \dfrac{n^2}{2n-1} & (i = j = n) \\[2ex] \dfrac{-i(i+1)}{2i+1} & (j = i+1) \\[2ex] \dfrac{-j(j+1)}{2j+1} & (i = j+1) \\[2ex] 0 & (|i-j| > 1) \end{cases}$$

3.6 三重対角行列の特殊な形

行列成分が以下のように表される。

$$A_{i,j} = \begin{cases} -2 & (i = j) \\ 1 & (|i-j| = 1) \\ 0 & (|i-j| > 1) \end{cases}$$

例えば、５×５行列の場合は、以下のようになる。

$$\begin{pmatrix} -2 & 1 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 1 & -2 \end{pmatrix}$$

この行列の逆行列の厳密解は、以下のように表現できる。

$$A^{-1}_{i,j} = \begin{cases} \dfrac{-i(n-j+1)}{n+1} & (i \le j) \\[2ex] \dfrac{-j(n-i+1)}{n+1} & (i > j) \end{cases}$$

また、この行列の固有値の厳密解は、以下のように表現できる。

$$\lambda = -4\sin^2 \frac{k\pi}{2(n+1)}$$

4. 試験行列：乱数行列

4.1 乱数行列の概要

行列成分を乱数によって生成することを考える。本ライブラリの乱数行列生成ルーチンでは、行列のみを生成し、右辺ベクトルや解ベクトルは生成しない。試験行列としての乱数行列の具体的利用方法は、例えば以下のようになろう。

[乱数行列の利用方法の一例]

(1) A を生成。
乱数行列生成ルーチンを用いて乱数行列 A を生成する。

(2) x の指定。
解ベクトル x を適当に与える。具体的には、乱数で与えたり、定数(例えばオール1)を与えればよい。

(3) b の計算。
右辺ベクトル $b = Ax$ を計算する。

(4) 解法ルーチンの妥当性チェック。
A と b を連立一次方程式の解法ルーチンに与えて、$Ax = b$ を数値的に解く。その結果 $\tilde{x}$ を、上記(2) で与えた x と比べて、妥当性を確認する。

試験行列としての利用を考慮し、ある程度行列の性質を指定できるようにしている。

[乱数行列生成に係わるパラメタ]

(1) 対角優位の指定。
行列が対角優位となることを指定できる。非対角成分を乱数で生成したあと、対角優位となるような対角成分の値を計算する。さらに、全成分が正であることを指定すれば、正定値の行列が得られる。

(2) 乱数生成方式の指定。
乱数生成方式として、乗算合同法/混合合同法のいずれかを指定できる。

(3) 乱数分布の指定。
乱数分布として、一様分布/正規分布のいずれかを指定できる。一様乱数の場合は、乱数の区間、すなわち最小値と最大値も指定する。正規分布の場合は、平均値と標準偏差も指定する。

(4) 行列の形式。
倍精度実数および倍精度複素数のルーチン群が用意されている。倍精度実数の場合、一般正方行列、対称行列、対称帯行列、非対称帯行列、対称疎行列、非対称疎行列、三角行列を扱う。帯、疎、三角については、非ゼロ成分のみを格納する圧縮形式と、ゼロ成分も含めて正方行列に配置する非圧縮形式が用意されている。

倍精度複素数の場合は、上記の「対称行列」を「エルミート行列」に読み換えればよい。

4.2 対角優位の指定

対角優位には、行対角優位と列対角優位がある。

[行対角優位]

対角成分の絶対値が、非対角成分の行和よりも小さくない行列である。数学的な定義では以下のようになる。

$$|a_{i,i}| \geq (\sum_{j=1}^{n} |a_{i,j}|) - |a_{i,j}|$$

但し、この式で等号が成立しているケースでは、数学的な定義では行対角優位であるが、数値解析に適用した場合に丸め誤差等の影響で行対角優位と認識されない可能性がある。そこで、非対角成分を乱数で生成した後に、その絶対値の和にさらに正の乱数を加えたものを対角成分とするようにしている。すなわち、＝のかわりに＞として、実質的には下式のような行対角優位行列を生成している。

$$|a_{i,i}| > (\sum_{j=1}^{n} |a_{i,j}|) - |a_{i,j}|$$

[列対角優位]

対角成分の絶対値が、非対角成分の列和よりも小さくない行列である。数学的な定義では以下のようになる。

$$|a_{j,j}| \geq (\sum_{i=1}^{n} |a_{i,j}|) - |a_{j,j}|$$

行対角優位の場合と同様の理由で、本ライブラリでは、下式を満たすように行列を生成している。

$$|a_{j,j}| > (\sum_{i=1}^{n} |a_{i,j}|) - |a_{j,j}|$$

なお、対称行列を前提として、絶対値ではない対角成分の値そのものが非対角成分の行和よりも大きければ、「正定値」となることが知られている。したがって、本ライブラリの乱数対称行列では、「行対角優位」かつ「対角成分がすべて正」をオプションで指定すれば、「正定値」の行列が得られることになる。

4.3 乱数の生成方法

乱数生成方式として、乗算合同法/混合合同法のいずれかを指定できる。詳細は参考文献[13]を参照されたい。

乗算合同法とは、適当な λ 、 m を与えて、以下の式によって逐次的に区間 [0 , m] の一様擬似乱数を生成するものである。

$$r_i = mod(\lambda r_{i-1}, m)$$

混合合同法とは、適当な λ 、 μ 、 m を与えて、以下の式によって逐次的に区間 [0 , m] の一様擬似乱数を生成するものである。

$$r_i = mod(\lambda r_{i-1} + \mu, m)$$

乱数分布として、一様分布/正規分布のいずれかを指定できる。一様乱数の場合は、乱数の区間、すなわち最小値と最大値も指定する。正規乱数の場合は、平均値と標準偏差も指定する。

5. 試験行列：アプリケーションベース

5.1 3次元流体7点差分疎行列生成

5.1.1 概要

3次元非圧縮流体解析に現れる圧力方程式の係数行列と右辺ベクトルおよび数値解をファイルに格納する。物理的境界条件として、細い直方体領域において、

- 入口で Poiseille flow の流速プロファイル
- 上下面ですべりなし、
- 左右面および出口で加速度なし、

を与える。

この条件のもとに3次元 Navier–Stokes 方程式をスタッガードメッシュに基づく7点差分で離散化し、速度場と圧力の交互反復計算によって数値解を得る。時間項は陽的差分によって扱われる。

収束時の流速分布は、計算領域の至る所でほとんど Poiseille flow となり、圧力は入口から出口に向かう x 座標についてほとんど線形の関数となる。出力対象が、流速分布ではなく、圧力方程式である点に注意されたい。

5.1.2 3次元非圧縮流体解析の要点

方程式は時間項を含む非圧縮 NS 方程式とする。現時点では、乱流モデル等は導入せず、移流項には低次の差分法を用いているため、高レイノルズ数の流れの解析は困難である。レイノルズ数は、100程度とするのが適当である。

連続の式は、以下のように表される。

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0$$

運動方程式は、以下のように表される。

$$\frac{\partial u}{\partial t}+\frac{\partial u^2}{\partial x}+\frac{\partial (vu)}{\partial y}+\frac{\partial (wu)}{\partial z}=-\frac{\partial p}{\partial x}+\frac{1}{Re}(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2})$$

$$\frac{\partial v}{\partial t}+\frac{\partial (uv)}{\partial x}+\frac{\partial v^2}{\partial y}+\frac{\partial (wv)}{\partial z}=-\frac{\partial p}{\partial y}+\frac{1}{Re}(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 v}{\partial z^2})$$

$$\frac{\partial w}{\partial t}+\frac{\partial (uw)}{\partial x}+\frac{\partial (vw)}{\partial y}+\frac{\partial w^2}{\partial z}=-\frac{\partial p}{\partial z}+\frac{1}{Re}(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}+\frac{\partial^2 w}{\partial z^2})$$

N-S 方程式の解法はtime-splitting method を用いる。空間的離散化としてスタッガードメッシュを採用し、時間に関しては陽的差分をとる。時間項を含めた方程式を解いて、解の変動が十分小さくなったときに、定常状態に達したと解釈する。

以下の説明では、流れの速度ベクトルを $\mathbf{u}=(u,v,w)$ と表すことにする。

(1) **u** の初期設定

u に適当な初期値を与える。

(2) $\mathbf{u}^{\star}$ の計算

拡散項と粘性項のみから仮の速度 $\mathbf{u}^{\star}$ を求める。

$$\mathbf{u}^{\star} = \boldsymbol{\Delta} \times \left(\mathbf{u} \cdot \nabla \mathbf{u} + \frac{\mathbf{1}}{\mathbf{Re}} \nabla^{\mathbf{2}} \mathbf{u} \right)$$

(3) $\mathbf{p}$ の計算

タイムステップを一つ進め、速度場と圧力に関する以下の Poisson 方程式を解く。

$$\nabla^{\mathbf{2}} \mathbf{p}^{\mathbf{n+1}} = \frac{\mathbf{1}}{\boldsymbol{\Delta}\mathbf{t}} \nabla \cdot \mathbf{u}^{\star}$$

前回に求めた圧力と今回求めた圧力との差が十分小さくなったとき、定常状態に達したと解釈して終了。

(4) $\mathbf{u}$ の計算

この $\mathbf{p}$ から速度 $\mathbf{u}$ を求める。

$$\mathbf{u}^{\mathbf{n+1}} = \mathbf{u}^{\star} - \boldsymbol{\Delta}\mathbf{t} \nabla \mathbf{p}^{\mathbf{n+1}}$$

(2) に行く。

5.2　2次元流体5点差分疎行列生成

5.2.1　概要

2次元非圧縮流体解析に現れる圧力方程式の係数行列と右辺ベクトルおよび数値解をファイルに格納する。物理的境界条件として、横に細長い2次元長方形領域において、

- 入口で Poiseille flow の流速プロファイル
- 上下面ですべりなし、
- 出口で加速度なし、

を与える。座標軸は流れ方向をx軸、x軸と垂直方向をz軸とする(3次元xyz座標から、xz面を切り出してきたと考える)。

この条件のもとに2次元 Navier–Stokes 方程式をスタッガードメッシュに基づく5点差分で離散化し、速度場と圧力の交互反復計算によって数値解を得る。時間項は陽的差分によって扱われる。

収束時の流速分布は、計算領域の至る所でほとんど Poiseille flow となり、圧力は入口から出口に向かうx座標についてほとんど線形の関数となる。出力対象が、流速分布ではなく、圧力方程式である点に注意されたい。

5.2.2　2次元非圧縮流体解析の要点

方程式は時間項を含む非圧縮 NS 方程式とする。本プログラムの場合、3次元バージョンdg3zpfdssとの比較を意識し、x、y、z成分のうち、y成分を落とす形で定式化している。すなわち、以下のようになる。

連続の式は、以下のように表される。

$$\frac{\partial u}{\partial x}+\frac{\partial w}{\partial z}=0$$

運動方程式は、以下のように表される。

$$\frac{\partial u}{\partial t}+\frac{\partial u^2}{\partial x}+\frac{\partial (wu)}{\partial z}=-\frac{\partial p}{\partial x}+\frac{1}{Re}(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial z^2})$$

$$\frac{\partial w}{\partial t}+\frac{\partial (uw)}{\partial x}+\frac{\partial w^2}{\partial z}=-\frac{\partial p}{\partial z}+\frac{1}{Re}(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial z^2})$$

N-S方程式の解法は基本的に3次元の場合と同様なので、5.1.2 を参照されたい。

5.3 2次元有限要素法帯生成

5.3.1 概要

連立一次方程式の数値解法ルーチンの開発・改良を行った場合には、適当な例題を用いて計算結果の妥当性を検証する必要がある。このような場合のテスト問題となる行列と右辺ベクトルおよび理論的厳密解を、任意の規模の方程式について生成するプログラムがあると便利である。

本プログラムは、有限要素法によって2次元応力問題を解く際に現れる剛性行列（左辺行列）と荷重ベクトル（右辺ベクトル）を生成して、ファイルに格納するプログラムである。これらの他に、変位ベクトルの理論的厳密解もファイルに格納する。ユーザは、剛性行列と荷重ベクトルをファイルから読み込んで適当な連立一次方程式解法ルーチンを用いて解き、その結果を理論的厳密解と比べることができる。本節では、本プログラムが生成する試験行列の物理的意味について述べる。

5.3.2 2次元応力問題の基礎方程式

2次元応力問題における基礎方程式は以下のようになる。

(1) つり合い方程式: 応力 $\boldsymbol{\sigma}$ と外力 $\boldsymbol{F}$ との関係を示す。

$$A^T\boldsymbol{\sigma} = -\boldsymbol{F}$$

(2) 構成方程式(generalized Hook's law): 応力 $\boldsymbol{\sigma}$ と歪 $\boldsymbol{\varepsilon}$ の関係を示す。

$$\boldsymbol{\sigma} = A\boldsymbol{\varepsilon}$$

(3) 歪−変位関係式: 歪 $\boldsymbol{\varepsilon}$ と変位 $\boldsymbol{e}$ との関係を示す。

$$\boldsymbol{\varepsilon} = A\boldsymbol{e}$$

ここで、A は、歪と変位を表す作用素、D は、応力と歪の関係を表す行列である。

$$A = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} , \quad D = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & 0 \\ \frac{\nu}{1-\nu} & 1 & 0 \\ 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix}$$

ここで、E は Young 率、ν は Poisson 比である。

また、外力 $\boldsymbol{F}$ 、応力 $\boldsymbol{\sigma}$ 、歪 $\boldsymbol{\varepsilon}$ 、変位 $\boldsymbol{e}$ の成分を以下のように表す。

$$\boldsymbol{F} = \begin{bmatrix} F_x \\ F_y \end{bmatrix} , \quad \boldsymbol{\sigma} = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} , \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} , \quad \boldsymbol{e} = \begin{bmatrix} u \\ v \end{bmatrix}$$

基礎方程式から、応力 $\boldsymbol{\sigma}$ と歪 $\boldsymbol{\varepsilon}$ を消去し、変位 $\boldsymbol{e}$ と外力 $\boldsymbol{F}$ の関係式を得る。

$$A^T DA\boldsymbol{e} = -\boldsymbol{F}$$

あとはこの連立偏微分方程式を適当な境界条件のもとに解けばよい。一般的な問題に対しては、次節に述べる有限要素法などを用いて数値的に解くことになる。

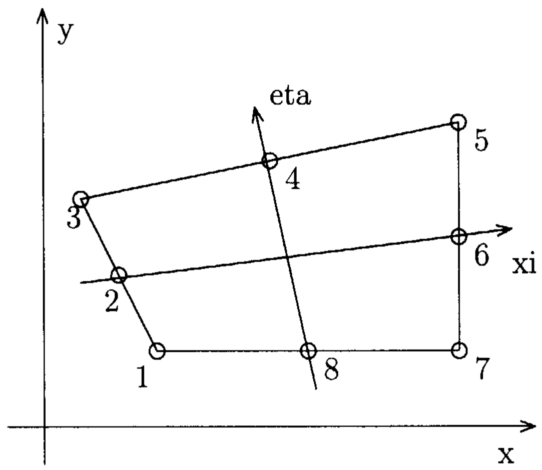

Fig. 5.3.3-1 The global and local coordinate systems of the 2-dimensional 8-noded element.

5.3.3 有限要素法による離散化

有限要素法そのものの説明は専門書[15) 16)]に譲り、ここでは、本プログラムが使用している定式化および要素形状関数について説明する。前節で得た変位 $\boldsymbol{e}$ と外力 $\boldsymbol{F}$ の関係式を Galerkin 法に基づいて定式化すると、要素剛性行列 K_M は、以下のようになる。

$$K_M = \iint (AN)^T D(AN)dxdy$$

ここで、N は形状関数である。本プログラムでは2次元4辺形8節点要素を用いる。すなわち、4辺形要素の頂点4個と辺の中点4個の計8個を節点にとり、節点の x、y 座標を $(x_i, y_i)(i = 1, 2, \cdots, 8)$、 u、v 変位を $(u_i, v_i)(i = 1, 2, \cdots, 8)$ で表すことにする。

要素内の任意の点の x、y 座標および u、v 変位が、節点の x、y 座標 (x_i, y_i) と u、v 変位 (u_i, v_i) および要素形状関数 $N_i(\xi, \eta)$ により、次のように表現されるものとする。

$$x = \sum_{i=1}^{8} N_i(\xi,\eta)x_i \ , \ y = \sum_{i=1}^{8} N_i(\xi,\eta)y_i \ , \ u = \sum_{i=1}^{8} N_i(\xi,\eta)u_i \ , \ v = \sum_{i=1}^{8} N_i(\xi,\eta)v_i$$

ここで、4辺形8節点の要素形状関数 $N_i(\xi, \eta)$ は、要素局所座標 ξ、η によって以下のように表される。

$$\boldsymbol{N}(\xi,\eta) = \begin{bmatrix} N_1(\xi,\eta) \\ N_2(\xi,\eta) \\ N_3(\xi,\eta) \\ N_4(\xi,\eta) \\ N_5(\xi,\eta) \\ N_6(\xi,\eta) \\ N_7(\xi,\eta) \\ N_8(\xi,\eta) \end{bmatrix} = \begin{bmatrix} \frac{1}{4}(1-\xi)(1-\eta)(-\xi-\eta-1) \\ \frac{1}{2}(1-\xi)(1-\eta^2) \\ \frac{1}{4}(1-\xi)(1+\eta)(-\xi+\eta-1) \\ \frac{1}{2}(1-\xi^2)(1+\eta) \\ \frac{1}{4}(1+\xi)(1+\eta)(\xi+\eta-1) \\ \frac{1}{2}(1+\xi)(1-\eta^2) \\ \frac{1}{4}(1+\xi)(1-\eta)(\xi-\eta-1) \\ \frac{1}{2}(1-\xi^2)(1-\eta) \end{bmatrix}$$

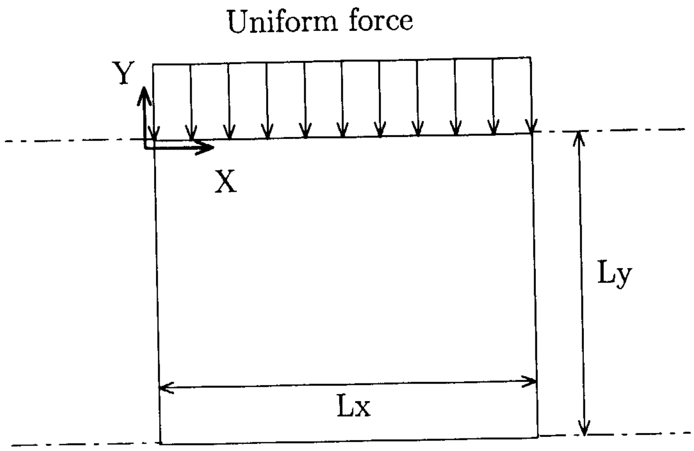

Fig. 5.3.4-1 Uniformly loaded two dimensional elastic body.

積分は、Gauss の求積法を用いる。本プログラムでは1つの次元について3点のサンプリング・ポイントをとっている。また、積分変数を x、y から ξ、η に置き換える必要がある。

$$
\begin{aligned}
K_M &= \iint (AN)^T D(AN) dxdy \\
&= \iint (AN)^T D(AN) \left| \frac{\partial(x,y)}{\partial(\xi,\eta)} \right| d\xi d\eta \\
&= \sum_{i=1}^{3} \sum_{j=1}^{3} k_i k_j f(\xi_i, \eta_j)
\end{aligned}
$$

ここで、

$$
f(\xi_i, \eta_j) = (AN)^T D(AN) \left| \frac{\partial(x,y)}{\partial(\xi,\eta)} \right|
$$

とおいた。ξ_i、η_j はサンプリング・ポイントであり、k_i、k_j は重みである。3点の場合のサンプリング・ポイントと重みは以下の通りである。

$$
\begin{aligned}
\xi_i\ ,\ \eta_j &= -\frac{\sqrt{15}}{5}\ ,0\ ,+\frac{\sqrt{15}}{5} \\
k_i\ ,\ k_j &= \frac{5}{9}\ ,\frac{8}{9}\ ,\ \frac{5}{9}
\end{aligned}
$$

このようにして作成した要素剛性行列 K_M を全体行列 K_B に重ね合わせる。荷重ベクトルも、一様荷重の境界積分を上記同様Gaussの求積法で求めればよい。

5.3.4 対象とする問題と厳密解

本プログラムで生成する剛性行列の元になる問題の境界条件と厳密解の導出について述べる。

［問題］ x方向に無限の長さ、y方向に l_y の高さを持つ2次元完全弾性体があり、この上面に一様荷重が与えられるものとする。この弾性体の任意位置における変位を求めよ。

この問題は、無限長の弾性体から有限の長さの部分を切り出して、左右両端においてx方向固定の境界条件を与えたものと等価となる。すなわち、具体的には以下のように書き換えることが可能である。

［上記に等価な問題］ x 方向長さ l_x 、y 方向長さ l_y の長方形の2次元完全弾性体がある。長方形の左上頂点に原点をとり、右向きに x 軸、上向きに y 軸をとる。以下の境界条件を与えるとき、長方形内の任意位置における変位を求めよ。

(1) 荷重: 上面 $0 \leq x \leq l_x, y = 0$ において、鉛直下向きに一様な荷重 w_y が働く。

$$S_x = 0 \ , \ S_y = -w_y$$

(2) 側面位置変位: 左右面 $x = 0$, l_x , $-l_y \leq y \leq 0$ において、x 方向の変位はゼロ。

$$u = 0$$

(3) 底面位置変位: 底面 $0 \leq x \leq l_x, y = -l_y$ において、x , y 方向の変位はゼロ。

$$u = 0 \ , \ v = 0$$

この問題の厳密解を求めてみる。いま、歪－応力関係式を書き下すと以下のようになる。

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & 0 \\ \frac{\nu}{1-\nu} & 1 & 0 \\ 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix} \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix}$$

題意より、せん断応力がゼロ、したがってせん断歪もゼロ。
また、x 方向歪もゼロである (x 方向応力はゼロとは限らないことに注意)。
y 方向応力は、上面の一様荷重 $w_y(const.)$ と等しくなる。

$$\tau_{xy} = 0 \ , \ \gamma_{xy} = 0 \ , \ \varepsilon_x = 0 \ , \ \sigma_y = w_y(const.)$$

これから、ε_y と σ_x を以下のように表すことができる。

$$\varepsilon_y = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} w_y \quad , \quad \sigma_x = \frac{\nu}{(1+\nu)} w_y$$

したがって、$(x\ ,\ y)$ における変位 $(u\ ,\ v)$ の厳密解は、以下のようになる。

$$\begin{aligned} u(x,y) &= 0 \\ v(x,y) &= \varepsilon_y(-l_y - y) = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} w_y(-l_y - y) \end{aligned}$$

6.　データ構造

本ライブラリにおける配列への行列の格納方法を示す。

6.1　一般行列

重複ローカル、逐次の場合

本ライブラリで扱う一般行列の格納形式について説明をする。重複ローカル配置では、全PEが同じ行列の全体を重複して持ち、処理のみ分割して並列に行なう。すなわち、重複ローカル配置は逐次処理の場合と同じ格納形式をとる。

例　6×6 の行列を３つのPEで処理する場合について述べる。

なお行列は以下のようにサイクリック列分割されているとする。

PE0	PE1	PE2	PE0	PE1	PE2
11	12	13	14	15	16
21	22	23	24	25	26
31	32	33	34	35	36
41	42	43	44	45	46
51	52	53	54	55	56
61	62	63	64	65	66

このとき、重複ローカルでは各PEが行列の全体を重複して持つため、各PEのメモリ上には以下のような行列が存在することになる。

```
   ( 11 12 13 14 15 16 )
   ( 21 22 23 24 25 26 )
A: ( 31 32 33 34 35 36 )
   ( 41 42 43 44 45 46 )
   ( 51 52 53 54 55 56 )
   ( 61 62 63 64 65 66 )
```

分割ローカルの場合

分割ローカル配置では、各PEが行列の部分行列を持つ。

例 6×6 の行列を3つのPEで処理する場合について述べる。

なお行列は以下のようにサイクリック列分割されているとする。

PE0	PE1	PE2	PE0	PE1	PE2
11	12	13	14	15	16
21	22	23	24	25	26
31	32	33	34	35	36
41	42	43	44	45	46
51	52	53	54	55	56
61	62	63	64	65	66

このとき、分割ローカルでは各PEがサイクリック列分割された行列の一部分のみ持つため、各PEのメモリ上には以下のような行列が存在することになる。

```
      ( 11 14 )         ( 12 15 )         ( 13 16 )
      ( 21 24 )         ( 22 25 )         ( 23 26 )
PE0:  ( 31 34 )   PE1:  ( 32 35 )   PE2:  ( 33 36 )
      ( 41 44 )         ( 42 45 )         ( 43 46 )
      ( 51 54 )         ( 52 55 )         ( 53 56 )
      ( 61 64 )         ( 62 65 )         ( 63 66 )
```

6.2 帯行列

重複ローカル、逐次の場合

本ライブラリで扱う帯行列の格納形式は、LAPACKの場合と同様である。重複ローカル配置では、全PEが同じ行列の全体を重複して持ち、処理のみ分割して並列に行なう。すなわち、重複ローカル配置は逐次処理の場合と同じ格納形式をとる。

上方向のバンド幅が ku 、下方向のバンド幅が kl である $n \times n$ の帯行列を格納する。

この帯行列を対角成分に平行な非ゼロ要素部分を (kl+ku+1)×n の矩形行列の上から下に向かって

$$a(1,*), a(2,*), \cdots, a(\mathrm{ku}+\mathrm{kl}+1,*)$$

順に格納することにより、$a(\mathrm{ku}+\mathrm{kl}+1,n)$ の大きさの行列に圧縮して格納する。

つまり、上方向のバンド幅がku、下方向のバンド幅がklであるので、ある列jに対して、

$$(j-\mathrm{ku},j),(j-\mathrm{ku}+1,j),\cdots,(j+\mathrm{kl}-1,j),(j+\mathrm{kl},j)$$

の非ゼロ要素部分があり、それぞれを

$$a(1,j),a(2,j),\cdots,a(\mathrm{ku}+\mathrm{kl},j),a(\mathrm{ku}+\mathrm{kl}+1,j)$$

に格納する。

例　ku = 1,kl = 2 である 6×6 の帯行列を重複ローカルで3PEで1列ずつブロックサイクリック処理する場合について述べる。

正方行列イメージでは、A は以下のように書ける。

$$A=\begin{pmatrix} 11 & 12 & & & & \\ 21 & 22 & 23 & & & \\ 31 & 32 & 33 & 34 & & \\ & 42 & 43 & 44 & 45 & \\ & & 53 & 54 & 55 & 56 \\ & & & 64 & 65 & 66 \end{pmatrix}$$

この行列が以下のように配列 $a(4,6)$ に格納される。PE0～PE2は、各列の処理を担当するPEを表す。

```
           PE0 PE1 PE2 PE0 PE1 PE2
a(1,*) = (  *   12  23  34  45  56 )
a(2,*) = ( 11   22  33  44  55  66 )
a(3,*) = ( 21   32  43  54  65   * )
a(4,*) = ( 31   42  53  64   *   * )
```

分割ローカル

分割ローカル配置では、各PEが行列の部分行列を持つ。

例　ku = 1,kl = 2 である 6×6 の帯行列を分割ローカルで3PEで1列ずつブロックサイクリック処理する場合について述べる。

グローバル正方行列イメージでは、A は以下のように書ける。

$$A=\begin{pmatrix} 11 & 12 & & & & \\ 21 & 22 & 23 & & & \\ 31 & 32 & 33 & 34 & & \\ & 42 & 43 & 44 & 45 & \\ & & 53 & 54 & 55 & 56 \\ & & & 64 & 65 & 66 \end{pmatrix}$$

この行列が以下のように配列 $a(4,2)$ に格納される。PE0～PE2は、各列の処理を担当するPEを表す。

```
PE0
            a(1,*) = (  *  34 )
            a(2,*) = ( 11  44 )
            a(3,*) = ( 21  54 )
            a(4,*) = ( 31  64 )
PE1
            a(1,*) = ( 12  45 )
            a(2,*) = ( 22  55 )
            a(3,*) = ( 32  65 )
            a(4,*) = ( 42   * )
PE2
            a(1,*) = ( 23  56 )
            a(2,*) = ( 33  66 )
            a(3,*) = ( 43   * )
            a(4,*) = ( 53   * )
```

6.3　7重対角疎行列

重複ローカル、逐次の場合

格納方法は、重複ローカルと逐次で同様なのでまとめて述べる。逐次処理は、重複ローカル配置でPE数が1の場合に相当する。

1. 3次元7点差分に現れる7重対角疎行列の成分を7本のベクトル

   a1(*),a2(*),...,a7(*)

   に格納する。

2. a4(*)に対角成分を格納し、行列の下から上に向かって

   a1(*),a2(*),...,a7(*)

   に格納する。

3. 3次元直方体をx,y,z方向にm1,m2,m3個に分割するとき、3次元7点差分は点(i,j,k)に対し、

   (i,j,k-1),(i,j-1,k),(i-1,j,k), (i+1,j,k),(i,j+1,k),(i,j,k+1)

   と関係を持つ。

4. 3次元の点をx,y,zの順に辞書式番号付けを行う。すなわち、

   (1,1,1),(2,1,1),...,(m1,1,1),(1,2,1),(2,2,1),...

   について1,2,3...という具合に番号を振る。

   すると、点(i,j,k)は、i + (j-1)*m1 + (k-1)*m1*m2 であり、これをあらためて

   I = i + (j-1)*m1 + (k-1)*m1*m2

とおくと、上記3) の関係は点Ｉに対し、

I-m1*m2, I-m1, I-1, I+1, I+m1, I+m1*m2

との関係になる。

5. ダミーエリアは特に設けない。

注) 疎行列の格納方式として、前後にm1*m2個のダミーゼロエリアを設ける形式のものもよく使われているので、混同しないように注意されたい。

重複ローカルでは、７重対角疎行列は

Aの (i,i-m1*m2) 成分が　a1(i) に
Aの (i,i-m1) 成分が　a2(i) に
Aの (i,i-1) 成分が　a3(i) に
Aの (i,i) 成分が　a4(i) に
Aの (i,i+1) 成分が　a5(i) に
Aの (i,i+m1) 成分が　a6(i) に
Aの (i,i+m1*m2) 成分が　a7(i) に

に入っているものとして処理する。

例　2×2×2で分割した直方体についての
ｖ７点中心差分の行列を４つのPEで処理する場合について述べる。

このとき、行列は以下のように非ゼロ要素が存在する。
なお、この行列は以下のようにサイクリック行分割されているとする。

PE0	11	12	13		15			
PE1	21	22	23	24		26		
PE2	31	32	33	34	35		37	
PE3		42	43	44	45	46		48
PE0	51		53	54	55	56	57	
PE1		62		64	65	66	67	68
PE2			73		75	76	77	78
PE3				84		86	87	88

このとき、重複ローカルでは、各PEが行列の全体を重複して持つため、各PEのメモリ上には以下のような7本のベクトルが存在することになる。

```
a1(*)= (  0  0  0  0 51 62 73 84 )
a2(*)= (  0  0 31 42 53 64 75 86 )
a3(*)= (  0 21 32 43 54 65 76 87 )
a4(*)= ( 11 22 33 44 55 66 77 88 )
a5(*)= ( 12 23 34 45 56 67 78  0 )
a6(*)= ( 13 24 35 46 57 68  0  0 )
a7(*)= ( 15 26 37 48  0  0  0  0 )
```

分割ローカルの場合

分割ローカル配置でも、重複ローカルと同様に7本のベクトルに格納されるが、各PEが行列の部分行列を持つ。

例　$2\times2\times2$で分割した直方体についての7点中心差分の行列を4つのPEで処理する場合について述べる。

このとき、行列は以下のように非ゼロ要素が存在する。
また、この行列は以下のようにサイクリック行分割されているとする。

PE0	11	12	13		15			
PE1	21	22	23	24		26		
PE2	31	32	33	34	35		37	
PE3		42	43	44	45	46		48
PE0	51		53	54	55	56	57	
PE1		62		64	65	66	67	68
PE2			73		75	76	77	78
PE3				84		86	87	88

このときこの行列は、重複ローカルと同様に7本のベクトルに格納されるが、分割ローカルでは、各PEがサイクリック行分割された行列の一部分のみ持つため、各PEのメモリ上には以下のような7本のベクトルが存在することになる。

```
PE0
    a1(*)= (  0 51 )
```

```
    a2(*)= (  0 53 )
    a3(*)= (  0 54 )
    a4(*)= ( 11 55 )
    a5(*)= ( 12 56 )
    a6(*)= ( 13 57 )
    a7(*)= ( 15  0 )

PE1
    a1(*)= (  0 62 )
    a2(*)= (  0 64 )
    a3(*)= ( 21 65 )
    a4(*)= ( 22 66 )
    a5(*)= ( 23 67 )
    a6(*)= ( 24 68 )
    a7(*)= ( 26  0 )

PE2
    a1(*)= (  0 73 )
    a2(*)= ( 31 75 )
    a3(*)= ( 32 76 )
    a4(*)= ( 33 77 )
    a5(*)= ( 34 78 )
    a6(*)= ( 35  0 )
    a7(*)= ( 37  0 )

PE3
    a1(*)= (  0 84 )
    a2(*)= ( 42 86 )
    a3(*)= ( 43 87 )
    a4(*)= ( 44 88 )
    a5(*)= ( 45  0 )
    a6(*)= ( 46  0 )
    a7(*)= ( 48  0 )
```

注) 規則的疎行列の分割ローカルの格納形式として、前後にm1*m2個のダミーゼロエリアを設ける形式のものもよく使われている。たとえば、PE0 が担当する行列成分は、以下のようになる。

```
PE0
    a1(*)= (  0  0  0  0  0 51  0  0  0  0 )
```

```
a2(*)= (  0  0  0  0  0 53  0  0  0  0 )
a3(*)= (  0  0  0  0  0 54  0  0  0  0 )
a4(*)= (  0  0  0  0 11 55  0  0  0  0 )
a5(*)= (  0  0  0  0 12 56  0  0  0  0 )
a6(*)= (  0  0  0  0 13 57  0  0  0  0 )
a7(*)= (  0  0  0  0 15  0  0  0  0  0 )
```

本書で述べるサブルーチンでは、このような前後のダミーエリアは不要であるので注意されたい。

6.4　3重対角行列

本ライブラリで扱う3重対角行列の格納形式は、帯行列の場合の$ku = 1, kl = 1$の場合に相当する。本ライブラリでの処理形態は、逐次処理のみである（試験行列）。

例　6×6の3重対角行列の場合について述べる

正方行列イメージでは、Aは以下のように書ける。

$$A = \begin{pmatrix} 11 & 12 & & & & \\ 21 & 22 & 23 & & & \\ & 32 & 33 & 34 & & \\ & & 43 & 44 & 45 & \\ & & & 54 & 55 & 56 \\ & & & & 65 & 66 \end{pmatrix}$$

この行列が以下のように配列$a(3,6)$に格納される。

```
a(1,*) = (  *  12  23  34  45  56 )
a(2,*) = ( 11  22  33  44  55  66 )
a(3,*) = ( 21  32  43  54  65   * )
```

6.5　三角行列

本ライブラリで扱う三角行列の格納形式について説明をする。本ライブラリでの処理形態は、逐次処理のみである（試験行列）。

$n \times n$の下三角行列を格納する。

この下三角行列の非ゼロ要素部分を列ごとに取りだし、続けて並べると以下のようになる。

$$a(1,1), a(2,1), \cdots, a(n,1), a(2,2), \cdots, a(n,2), \cdots, a(n,n)$$

これをこの順番に、$n(n+1)/2$の大きさの一次元配列に格納する。

例 5×5 の下三角行列の場合について述べる。

正方行列イメージでは、 A は以下のように書ける。

$$A = \begin{pmatrix} 11 & & & & \\ 21 & 22 & & & \\ 31 & 32 & 33 & & \\ 41 & 42 & 43 & 44 & \\ 51 & 52 & 53 & 54 & 55 \end{pmatrix}$$

この行列が以下のように配列 $a(10)$ に格納される。

```
a(*) = (11  21  31  41  51  22  32  42  52  33  43  53  44  54  55)
```

7. 並列化のアルゴリズム

ここでは1-ノルムと∞-ノルムの計算をするときのアルゴリズムを示す.

7.1 1-ノルム

Fig. 7.1-1 に示すように，それぞれのプロセッサが担当の列の合計を計算して，その中で最大の合計値を選択する．例えはプロセッサ0番は列1と列5を担当して，列1の合計であるsum1と列5の合計であるsum5を計算する．その後でsum1とsum5を比較して大きい方を選択する(ここではsum1 > sum5とする). この動作を4つのプロセッサが独立に行い，それぞれの最大値(この例ではsum1，sum6，sum7，sum4)を計算し，最終的に4つのプロセッサ同士が通信することによって，それぞれの最大値の中で一番大きな値を計算する(この例ではsum6となる).

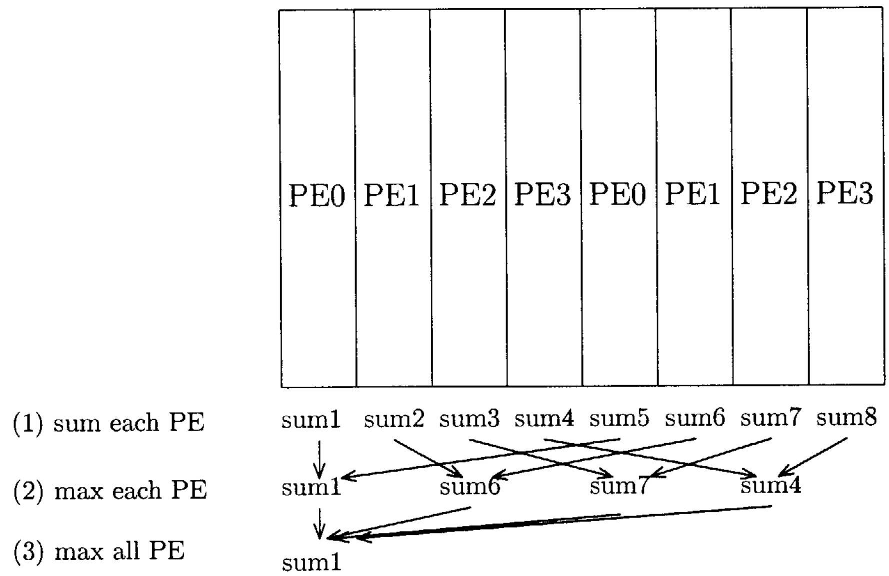

Fig. 7.1-1 1–norm

7.2 ∞-ノルム

Fig. 7.2-1 に示すように，それぞれのプロセッサが担当の列の分の行の合計を計算する．例えはプロセッサ0番は列1と列5を担当し，列1と列5の1行〜8行のそれぞれの行の合計を計算する(ここではPE0_sum1〜PE0_sum8にあたり，PE0_sum1=$|a_{11}|+|a_{15}|$となる). この動作を4つのプロセッサが独立に行い，4つのプロセッサ同士が通信することによって，それぞれの行の合計がプロセッサ0番に送られる(sum1〜sum8). 最後にプロセッサ0番が送られたそれぞれの行の合計の最大値を計算する(この例ではsum3となる).

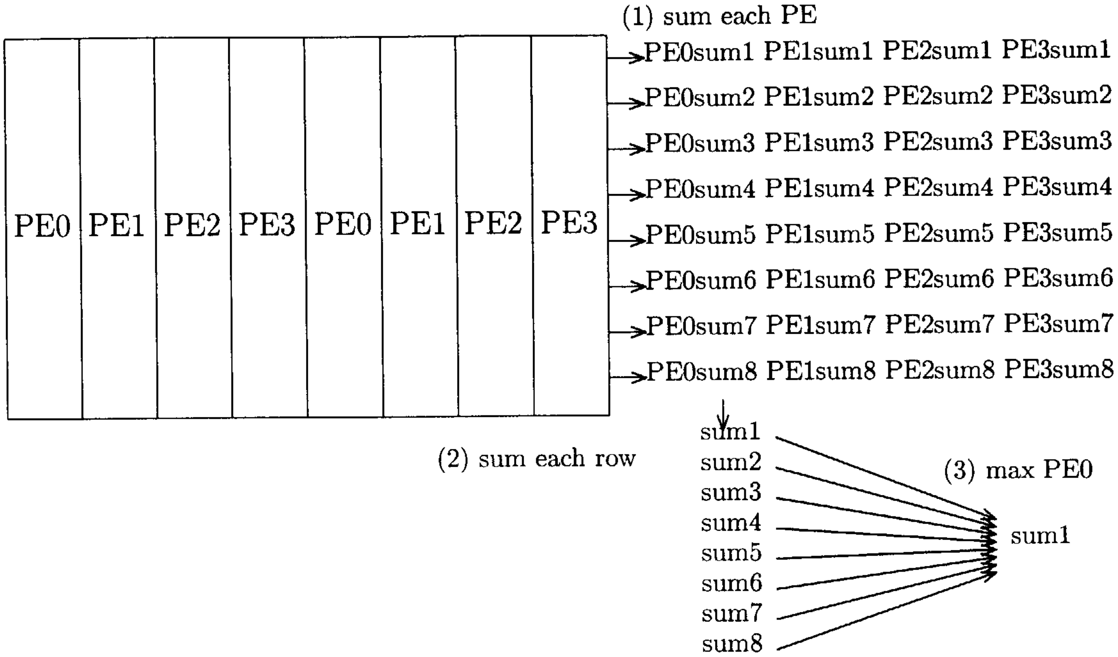

Fig. 7.2-1 ∞–norm

7.3 逆行列

本ライブラリでは、まずLU分解を行い、次に逆行列を求めている。ピボット軸の入れ替え、LU分解、逆行列計算については処理を並列化している。特に内積計算に帰着できる部分については、比較的容易に並列化が可能である。

$\|A^{-1}\|$ の近似計算部分では、LU分解結果を用いた前進代入、後退代入が必要となり、この部分も形の上では並列化可能であるが、並列化を優先すると頻繁に通信するアルゴリズムを採用せざるをえないため、台数が多くなりすぎると、性能は逆に劣化する。

参考文献

1) E.Anderson 他 (小国力訳) 「行列計算パッケージLAPACK利用の手引」. 丸善, 1995.

2) Stewart, G.W. Introduction to Matrix Computations. Academic Press 1973.

3) 村田健郎「線形代数と線形計算法序説」. サイエンス社, 1987.

4) N.J.Higham. "FORTRAN Codes for Estimating the One-Norm of a Real or Complex Matrix, with Applications to Condition Estimation". ACM Trans. Math. Soft., Vol.14,No.4, pp.381-396. ,Dec.,1988.

5) F. シャトラン. 行列の固有値. シュプリンガー・フェアラーク東京 1993.

6) R.Brawer and M.Pirovino "The linear algebra of the Pascal matrix". Linear Algebra and Appl., 174,1992, pp.13-23

7) C.A.J.Fletcher "Computational Techiniques for Fluid Dynamics, Vol.1 and Vol.2". Springer Verlag,1991

8) N.J.Higham "The Test Matrix Toolbox for MATLAB(Version3.0)". Numerical Analysis Report No.276, Dept. of Mathematics, Univ. of Manchester,Sept. 1995

9) M.Newman and J.Todd "The evaluation of matrix inversion programs". J.Soc.Indust.Appl.Math. 6(4) pp.466-476, 1958

10) M.L.Pei "A test matrix for inversion procedures". Comm. ACM, 5(10),1962, p.508

11) 戸川隼人「マトリクスの数値計算」. オーム社 1971

12) J.R. ウェストレイク (戸川隼人訳) 「コンピュータのための線形計算ハンドブック」. 倍風館、1972.

13) 渡辺力 他 編「Fortran77 による数値計算ソフトウェア」. 丸善、1989

14) 張紹良, 藤野清次「ランチョス・プロセスに基づく積型反復解法」. 応用数理 Vol.5, No.4, 1995, pp. 343-360.

15) I.M.Smith (戸川隼人訳). 有限要素法のプログラミング　構造・流体・地盤への応用. ワイリー・ジャパン、(1984).

16) O.C.Zienkiewics and R.L.Taylor. The Finite Element Method 4th. ed. Vol.1. McGRAW-Hill、(1994).

17) J.E.Akin. Finite Element Analysis for Undergraduate. Academic Press、(1986).

This is a blank page.

Part II

プログラム使用方法

This is a blank page.

8. 連立一次方程式誤差評価

8.1 実数一般密行列誤差評価(MPI分割ローカル版) JAERI_dgdzlhipm

8.1.1 機能

実数密正方行列を係数行列とする連立一次方程式について係数行列の条件数および誤差限界を見積もる(MPI分割ローカル版)。本サブルーチンは主要入力データとして、行列、計算された解、右辺ベクトルをとり、主要出力データとして係数行列のノルム、逆行列のノルム、条件数、誤差限界を返す。

ノルムは、1-ノルムと無限大ノルムの両方について求める。逆行列のノルムについては、逆行列を明示的に計算せずに近似的に求める Higham [4)]の方法と、実際に逆行列を計算してそのノルムを計算する厳密な方法の2通りを行い、それぞれについて、1-ノルムと無限大ノルムを計算するので、組み合わせの数としては、4通りとなる。したがって、条件数と誤差限界も、この4通りに対応したものが計算される。

本サブルーチンは、分散メモリ型並列計算機での分割ローカル配置(すなわち各PEが行列の部分行列を持つ)形態を想定する。つまり、本サブルーチンが呼ばれる前にMPIのイニシャライズ等が行われ、かつ、行列等が分割配置されていることを前提とする。具体的には本サブルーチンとともに供されるテストプログラムを参照されたい。

8.1.2 コーリングシーケンス

```
    call        JAERI_dgdzlhipm
   &                   (mpicw ,nnG
   &                   ,whohas,nclmL ,mapG2L,mapL2G,EPSMCH
   &                   ,LDA          ,AL     ,XX     ,BB
   &                   ,RNRM         ,BNRM
   &                   ,ANRM         ,AVNRD ,AVNRX ,RCOND ,RCONX
   &                   ,ERBF1D,ERBF1X,ERBF2D,ERBF2X
   &                   ,IWORK ,DWORK ,XWORK ,IRTN   )
```

8.1.3 引数

用法: in　入力用。値は壊されない。
　　　out　出力用。
　　　i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。
　　　w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)
　　　D　倍精度実数型(double precision)
　　　Z　倍精度複素数型(complex*16)
　　　L　論理型(logical)
　　　C　文字型(character $*$ ($*$))

変数／配列(次元)	I/O	型	意味
mpicw	in	I	MPIコミュニケータ
nnG	in	I	連立一次方程式の実際の次元を示す。$nnG \leq LDA$
whohas(LDA)	in	I	元の行列であるグローバル行列の各列を処理担当するランク番号を格納する配列。グローバル行列を仮にAG(*,*)で表すとすると、そのj列目AG(*,j)を処理担当するランクの値$(=0,1,\cdots,PE数-1)$がwhohas(j)に格納されているものとする。whohasの内容は、全PE共通。
nclmL(0:PE数-1)	in	I	各ランク毎のローカル行列ALにおいて実際に処理対象となる列の数。ip$(=0,1,\cdots,PE数-1)$番目のランクが処理担当する列の数が、nclmL(ip)に格納されているものとする。nclmLの内容は、全PE共通。
mapG2L(LDA)	in	I	元の行列であるグローバル行列からローカル行列aL(*,*)への列の対応を定義する。グローバル行列を仮にAG(*,*)で表すとするとAGのjG番目の列が、aLのjL番目の列に格納されているとき、jL = mapG2L(jG)の関係にある。AGのjG番目の列が、自ランクのaLに存在しない場合は、-1が格納されているものとする。mapG2Lの内容は、PE毎に異なる。
mapL2G(ncmaxL)	in	I	ローカル配列aL(*,*)から、元の行列であるグローバル配列への列の対応を定義する。グローバル行列を仮にAG(*,*)で表すとするとaLのjL番目の列が、AGのjG番目の列に格納されているとき、jG = mapL2G(jL)の関係にある。mapL2Gの内容は、PE毎に異なる。
EPSMCH	in	I	倍精度実数のマシン・イプシロン$(=\epsilon)$。LAPACKに含まれる関数DLAMCH()などによりあらかじめ計算しておくとよい。
LDA	in	I	行列格納用配列aLの一次元目の宣言サイズ。
AL(LDA,ncmaxL)	i/o	D	実密行列。元の行列をブロック・サイクリック列分割したものを縮小した形で与える。戻り値は逆行列となる。
XX(LDA)	in	I	$Ax=b$の解x(精度評価対象)。
BB(LDA)	in	I	$Ax=b$の右辺ベクトルb。
RNRM(3)	out	D	残差ベクトル$Ax-b$の1–ノルムおよび∞–ノルム(それぞれ、RNRM(1)、RNRM(3)に格納される)。
BNRM(3)	out	D	右辺ベクトルbの1–ノルムおよび∞–ノルム(それぞれ、BNRM(1)、BNRM(3)に格納される)。
ANRM(3)	out	D	係数行列Aの1–ノルムおよび∞–ノルム(それぞれ、AVNRD(1)、AVNRD(3)に格納される)。

変数／配列(次元)	I/O	型	意味
AVNRD(3)	out	D	係数行列の逆行列 A^{-1}の厳密な 1–ノルムおよび ∞–ノルム(それぞれ、AVNRD(1)、AVNRD(3)に格納される)。
AVNRX(3)	out	D	係数行列の逆行列 A^{-1}の近似的な 1–ノルムおよび ∞–ノルム(それぞれ、AVNRX(1)、AVNRX(3)に格納される)。
RCOND(3)	out	D	厳密な条件数(1–ノルム、∞–ノルム に基づくものが、それぞれ RCOND(1)、RCOND(3)に格納される)。
RCONX(3)	out	D	近似的な条件数(1–ノルム、∞–ノルム に基づくものが、それぞれ RCONX(1)、RCONX(3)に格納される)。
ERBF1D(3)	out	D	公式1で計算した誤差限界(厳密条件数に基づく) ($=\kappa_1(A)\epsilon$)。(1–ノルム、∞–ノルム に基づくものが、それぞれ ERBF1D(1)、ERBF1D(3)に格納される)。
ERBF1X(3)	out	D	公式1で計算した誤差限界(近似条件数に基づく) ($=\kappa_1(A)\epsilon$)。(1–ノルム、∞–ノルム に基づくものが、それぞれ ERBF1X(1)、ERBF1X(3)に格納される)。
ERBF2D(3)	out	D	公式2で計算した誤差限界(厳密条件数に基づく) ($=\kappa_1(A)\frac{\|Ax-b\|_1}{\|b\|_1}$)。(1–ノルム、$\infty$–ノルム に基づくものが、それぞれ ERBF2D(1)、ERBF2D(3)に格納される)。
ERBF2X(3)	out	D	公式2で計算した誤差限界(近似条件数に基づく) ($=\kappa_1(A)\frac{\|Ax-b\|_1}{\|b\|_1}$)。(1–ノルム、$\infty$–ノルム に基づくものが、それぞれ ERBF2X(1)、ERBF2X(3)に格納される)。
IWORK(LDA*2)	w	I	作業用配列
DWORK(LDA*2)	w	D	作業用配列
XWORK(LDA*2)	w	D	作業用配列
IRTN	out	I	リターンコード。 $=0$：正常終了。

データ構造に関する注意事項　本サブルーチンは配列データの並列処理分散メモリへのデータ配置として列に関するブロック・サイクリック分割を採用している。それ故、利用者はブロック・サイクリック列分割により行列等の配列データの分割を本サブルーチンの呼び出し前に行なっておく必要がある。

8.2 実数一般密行列誤差評価(MPI重複ローカル版) JAERI_dgdzlhiom

8.2.1 機能

実数密正方行列を係数行列とする連立一次方程式について係数行列の条件数および誤差限界を見積もる(MPI重複ローカル版)。

本サブルーチンは主要入力データとして、行列、計算された解、右辺ベクトルをとり、主要出力データとして係数行列のノルム、逆行列のノルム、条件数、誤差限界を返す。

ノルムは、1-ノルムと無限大ノルムの両方について求める。逆行列のノルムについては、逆行列を明示的に計算せずに近似的に求める Higham [4]の方法と、実際に逆行列を計算してそのノルムを計算する厳密な方法の2通りを行い、それぞれについて、1-ノルムと無限大ノルムを計算するので、組み合わせの数としては、4通りとなる。したがって、条件数と誤差限界も、この4通りに対応したものが計算される。

本サブルーチンは、分散メモリ型並列計算機での重複ローカル配置(すなわち各PEが行列全体を持つ)形態を想定する。つまり、本サブルーチンが呼ばれる前にMPIのイニシャライズ等は必要であるが、行列の分割配置は必要ない。具体的には本サブルーチンとともに供されるテストプログラムを参照されたい。

8.2.2 コーリングシーケンス

```
  call        JAERI_dgdzlhiom
 &                  (mpicw ,nnG
 &                  ,whohas,nclmL
 &                  ,EPSMCH
 &                  ,LDA          ,AL     ,XX     ,BB
 &                  ,RNRM         ,BNRM
 &                  ,ANRM         ,AVNRD  ,AVNRX  ,RCOND  ,RCONX
 &                  ,ERBF1D,ERBF1X,ERBF2D,ERBF2X
 &                  ,IWORK ,DWORK ,XWORK
 &                  ,IRTN   )
```

8.2.3 引数

用法: in　入力用。値は壊されない。

out　出力用。

i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。

w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)

D　倍精度実数型(double precision)

Z　倍精度複素数型(complex*16)

L　論理型(logical)

C　文字型(character * (*))

変数／配列(次元)	I/O	型	意味
mpicw	in	I	MPIコミュニケータ
nnG	in	I	連立一次方程式の実際の次元を示す。$nnG \leq LDA$
whohas(LDA)	in	I	元の行列であるグローバル行列の各列を処理担当するランク番号を格納する配列。グローバル行列を仮にAG(*,*)で表すとすると、そのj列目AG(*,j)を処理担当するランクの値($= 0, 1, \cdots, PE$数$- 1$)がwhohas(j)に格納されているものとする。whohasの内容は、全PE共通。
nclmL(0:PE数-1)	in	I	各ランク毎のローカル行列ALにおいて実際に処理対象となる列の数。ip($= 0, 1, \cdots, PE$数$- 1$)番目のランクが処理担当する列の数が、nclmL(ip)に格納されているものとする。nclmLの内容は、全PEについて共通。
EPSMCH	in	I	倍精度実数のマシン・イプシロン($= \epsilon$)。LAPACKに含まれる関数DLAMCH()などによりあらかじめ計算しておくとよい。
LDA	in	I	行列格納用配列aLの一次元目の宣言サイズ。
AL(LDA,LDA)	i/o	D	実密行列。グローバルな形で与える。すなわち、分割(縮小)した形で与える必要はない。。戻り値は、各PEが担当した部分に逆行列が格納される形となる(全体が逆行列になるとは限らない)。
XX(LDA)	in	I	$Ax = b$の解x(精度評価対象)。
BB(LDA)	in	I	$Ax = b$の右辺ベクトルb。
RNRM(3)	out	D	残差ベクトル$Ax - b$の1–ノルムおよび∞–ノルム(それぞれ、RNRM(1)、RNRM(3)に格納される)。
BNRM(3)	out	D	右辺ベクトルbの1–ノルムおよび∞–ノルム(それぞれ、BNRM(1)、BNRM(3)に格納される)。
ANRM(3)	out	D	係数行列Aの1–ノルムおよび∞–ノルム(それぞれ、AVNRD(1)、AVNRD(3)に格納される)。
AVNRD(3)	out	D	係数行列の逆行列A^{-1}の厳密な1–ノルムおよび∞–ノルム(それぞれ、AVNRD(1)、AVNRD(3)に格納される)。
AVNRX(3)	out	D	係数行列の逆行列A^{-1}の近似的な1–ノルムおよび∞–ノルム(それぞれ、AVNRX(1)、AVNRX(3)に格納される)。
RCOND(3)	out	D	厳密な条件数(1–ノルム、∞–ノルムに基づくものが、それぞれRCOND(1)、RCOND(3)に格納される)。
RCONX(3)	out	D	近似的な条件数(1–ノルム、∞–ノルムに基づくものが、それぞれRCONX(1)、RCONX(3)に格納される)。

変数／配列(次元)	I/O	型	意味
ERBF1D(3)	out	D	公式1で計算した誤差限界(厳密条件数に基づく) ($=\kappa_1(A)\epsilon$)。(1-ノルム、∞-ノルム に基づくものが、それぞれ ERBF1D(1)、ERBF1D(3) に格納される)。
ERBF1X(3)	out	D	公式1で計算した誤差限界(近似条件数に基づく) ($=\kappa_1(A)\epsilon$)。(1-ノルム、∞-ノルム に基づくものが、それぞれ ERBF1X(1)、ERBF1X(3) に格納される)。
ERBF2D(3)	out	D	公式2で計算した誤差限界(厳密条件数に基づく) ($=\kappa_1(A)\frac{\|Ax-b\|_1}{\|b\|_1}$)。(1-ノルム、∞-ノルム に基づくものが、それぞれ ERBF2D(1)、ERBF2D(3) に格納される)。
ERBF2X(3)	out	D	公式2で計算した誤差限界(近似条件数に基づく) ($=\kappa_1(A)\frac{\|Ax-b\|_1}{\|b\|_1}$)。(1-ノルム、∞-ノルム に基づくものが、それぞれ ERBF2X(1)、ERBF2X(3) に格納される)。
IWORK(LDA*2)	w	I	作業用配列
DWORK(LDA*2)	w	D	作業用配列
XWORK(LDA*2)	w	D	作業用配列
IRTN	out	I	リターンコード。 ＝0：正常終了。

データ構造に関する注意事項　本サブルーチンは配列データの並列処理分散メモリへのデータ配置として重複ローカルを採用している。それ故、利用者は行列グローバル配置(従来の単一メモリインターフェース)のまま本サブルーチンに渡せばよい。

8.3 実数一般行列誤差評価 (逐次版) JAERI_dgdzlhiss

8.3.1 機能

実数密正方行列を係数行列とする連立一次方程式について係数行列の条件数および誤差限界を見積もる(逐次版)。

本サブルーチンは主要入力データとして、行列、計算された解、右辺ベクトルをとり、主要出力データとして係数行列のノルム、逆行列のノルム、条件数、誤差限界を返す。

ノルムは、1-ノルムと無限大ノルムの両方について求める。

本サブルーチンは逐次処理計算機での使用を想定する (通常のベクトル型スーパーコンピューターを含む)。

8.3.2 コーリングシーケンス

```
      call JAERI_dgdzlhiss
     &     (         NN     ,EPSMCH,
     &      LDA             ,AA     ,XX     ,BB     ,
     &      RNRM1  ,RNRMI  ,BNRM1  ,BNRMI  ,
     &      ANRM1  ,ANRMI  ,AVNRD1,AVNRDI,AVNRX1,AVNRXI,
     &                      ,RCOND1,RCONDI,RCONX1,RCONXI,
     &      ERBDA1,ERBDAI,ERBDB1,ERBDBI,
     &      IWORK  ,DWORK  ,ZWORK  ,
     &      IRTN            )
```

8.3.3 引数

用法: in 入力用。値は壊されない。
out 出力用。
i/o 入出力用。入力として使用され、かつ戻り時に値が設定される。
w 作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I 整数型 (integer)
D 倍精度実数型 (double precision)
Z 倍精度複素数型 (complex*16)
L 論理型 (logical)
C 文字型 (character * (*))

変数／配列 (次元)	I/O	型	意味
NN	in	I	連立一次方程式の次元を示す。
EPSMCH	in	I	倍精度実数のマシン・イプシロン ($=\epsilon$)。
LDA	in	I	行列格納用配列 AL の一次元目の宣言サイズ。

変数／配列(次元)	I/O	型	意味
AA(LDA,*)	i/o	D	実密行列。逆行列が戻り値となる。
XX(LDA)	in	I	$Ax=b$ の解 x。
BB(LDA)	in	I	$Ax=b$ の右辺ベクトル b。
RNRM1	out	D	残差ベクトル $Ax-b$ の1–ノルム。
RNRMI	out	D	残差ベクトル $Ax-b$ の∞–ノルム。
BNRM1	out	D	右辺ベクトル b の 1–ノルム。
BNRMI	out	D	右辺ベクトル b の ∞–ノルム。
ANRM1	out	D	係数行列 A の 1–ノルム($=\|A\|_1$)。
ANRMI	out	D	係数行列 A の ∞–ノルム($=\|A\|_\infty$)。
AVNRD1	out	D	逆行列 A^{-1} の厳密 1–ノルム($=\|A^{-1}\|_1$)。
AVNRDI	out	D	逆行列 A^{-1} の厳密 ∞–ノルム($=\|A^{-1}\|_\infty$)。
AVNRX1	out	D	逆行列 A^{-1} の近似 1–ノルム($=\|A^{-1}\|_1$)。
AVNRXI	out	D	逆行列 A^{-1} の近似 ∞–ノルム($=\|A^{-1}\|_\infty$)。
RCOND1	out	D	1–ノルムによる厳密条件件数 ($\kappa_1(A)=\|A\|_1\|A^{-1}\|_1$)。
RCONDI	out	D	∞–ノルムによる厳密条件件数($\kappa_\infty(A)=\|A\|_\infty\|A^{-1}\|_\infty$)。
RCONX1	out	D	1–ノルムによる近似条件件数 ($\kappa_1(A)=\|A\|_1\|A^{-1}\|_1$)。
RCONXI	out	D	∞–ノルムによる近似条件件数($\kappa_\infty(A)=\|A\|_\infty\|A^{-1}\|_\infty$)。
ERBDA1	out	D	公式1、厳密 1–ノルムによる誤差限界 ($=\kappa_1(A)\ \epsilon$)。
ERBDAI	out	D	公式1、厳密 ∞–ノルムによる誤差限界 ($=\kappa_\infty(A)\ \epsilon$)。
ERBDB1	out	D	公式2、厳密 1–ノルムによる誤差限界 ($=\kappa_1(A)\frac{\|Ax-b\|_1}{\|b\|_1}$)。
ERBDBI	out	D	公式2、厳密 ∞–ノルムによる誤差限界 ($=\kappa_\infty(A)\frac{\|Ax-b\|_\infty}{\|b\|_\infty}$)。
ERBXA1	out	D	公式1、近似 1–ノルムによる誤差限界 ($=\kappa_1(A)\ \epsilon$)。
ERBXAI	out	D	公式1、近似 ∞–ノルムによる誤差限界 ($=\kappa_\infty(A)\ \epsilon$)。
ERBXB1	out	D	公式2、近似 1–ノルムによる誤差限界 ($=\kappa_1(A)\frac{\|Ax-b\|_1}{\|b\|_1}$)。
ERBXBI	out	D	公式2、近似 ∞–ノルムによる誤差限界 ($=\kappa_\infty(A)\frac{\|Ax-b\|_\infty}{\|b\|_\infty}$)。
IWORK(LDA)	w	I	作業用配列
DWORK(LDA*2)	w	D	作業用配列
ZWORK(LDA*2)	w	D	作業用配列
irtn	out	I	リターンコード。＝0 のとき、正常終了。

データ構造に関する注意事項　特になし。

8.4 複素数一般密行列誤差評価(MPI分割ローカル版) JAERI_zgdzlhipm

8.4.1 機能

複素数一般密行列の連立一次方程式に対して係数行列の条件数および誤差限界を見積もる(MPI分割ローカル版)。

本サブルーチンは主要入力データとして、行列、計算された解、右辺ベクトルをとり、主要出力データとして係数行列のノルム、逆行列のノルム、条件数、誤差限界を返す。

ノルムは、1-ノルムと無限大ノルムの両方について求める。条件数と誤差限界も、1-ノルムと無限大ノルムの2通りに対応したものを求める。

本サブルーチンは分散メモリ型並列計算機での分割ローカル配置、すなわち各PEが行列の部分行列を持つという稼働環境を想定する。

8.4.2 コーリングシーケンス

```
      call JAERI_zgdzlhipm
     &     (mpicw ,nnG    ,
     &      whohas,nclmL  ,mapG2L,mapL2G,
     &      EPSMCH,
     &      LDA    ,ncmaxL        ,aL     ,XX     ,BB     ,
     &      RNRM1 ,RNRMI  ,BNRM1 ,BNRMI  ,
     &      ANRM1 ,ANRMI  ,AIVNR1,AIVNRI,RCOND1,RCONDI,
     &      ERBDA1,ERBDAI,ERBDB1,ERBDBI,
     &      IWORK ,DWORK  ,ZWORK  ,
     &      IRTN  )
```

8.4.3 引数

用法: in　入力用。値は壊されない。
　　out　出力用。
　　i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。
　　w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)
　　D　倍精度実数型(double precision)
　　Z　倍精度複素数型(complex*16)
　　L　論理型(logical)
　　C　文字型(character $*$ ($*$))

変数／配列(次元)	I/O	型	意味
mpicw	in	I	MPIコミュニケータ
nnG	in	I	連立一次方程式の次元を示す。$nnG \le LDA$
whohas(LDA)	in	I	元の行列であるグローバル行列の各列を処理担当するランク番号を格納する配列。グローバル行列を仮にAG(*,*)で表すとすると、そのj列目AG(*,j)を処理担当するランクの値$(=0,1,\cdots,PE数-1)$がwhohas(j)に格納されているものとする。whohasの内容は、全PE共通。
nclmL(0:PE数-1)	in	I	各ランク毎のローカル行列aLにおいて実際に処理対象となる列の数。ip$(=0,1,\cdots,PE数-1)$番目のランクが処理担当する列の数が、nclmL(ip)に格納されているものとする。nclmLの内容は、全PEについて共通。
mapG2L(LDA)	in	I	元の行列であるグローバル行列からローカル行列aL(*,*)への列の対応を定義する。グローバル行列を仮にAG(*,*)で表すとするとAGのjG番目の列が、aLのjL番目の列に格納されているとき、jL = mapG2L(jG)の関係にある。AGのjG番目の列が、自ランクのaLに存在しない場合は、-1が格納されているものとする。mapG2Lの内容は、PE毎に異なる。
mapL2G(ncmaxL)	in	I	ローカル配列aL(*,*)から、元の行列であるグローバル配列への列の対応を定義する。グローバル行列を仮にAG(*,*)で表すとするとaLのjL番目の列が、AGのjG番目の列に格納されているとき、jG = mapL2G(jL)の関係にある。mapL2Gの内容は、PE毎に異なる。
EPSMCH	in	I	倍精度実数のマシン・イプシロン$(=\epsilon)$。LAPACKに含まれる関数DLAMCH()などにより計算できる。
LDA	in	I	行列格納用配列aLの一次元目の宣言サイズ。
ncmaxL	in	I	行列格納用配列aLの二次元目の宣言サイズ。
aL(LDA,ncmaxL)	i/o	Z	複素数密行列。元の行列をブロック・サイクリック列分割したものを縮小形で与える。逆行列が戻値となる。
XX(LDA)	in	I	$Ax=b$の解x。
BB(LDA)	in	I	$Ax=b$の右辺ベクトルb。
RNRM(3)	out	D	残差ベクトル$Ax-b$の1−ノルムおよび∞−ノルム(それぞれ、RNRM(1)、RNRM(3)に格納される)。
BNRM(3)	out	D	右辺ベクトルbの1−ノルムおよび∞−ノルム(それぞれ、BNRM(1)、BNRM(3)に格納される)。

変数／配列(次元)	I/O	型	意味
ANRM(3)	out	D	係数行列Aの 1–ノルムおよび ∞–ノルム(それぞれ、AVNRD(1)、AVNRD(3)に格納される)。
AVNRD(3)	out	D	係数行列の逆行列 A^{-1}の厳密な 1–ノルムおよび ∞–ノルム(それぞれ、AVNRD(1)、AVNRD(3)に格納される)。
AVNRX(3)	out	D	係数行列の逆行列 A^{-1}の近似的な 1–ノルムおよび ∞–ノルム(それぞれ、AVNRX(1)、AVNRX(3)に格納される)。
RCOND(3)	out	D	厳密な条件数(1–ノルム、∞–ノルム に基づくものが、それぞれ RCOND(1)、RCOND(3)に格納される)。
RCONX(3)	out	D	近似的な条件数(1–ノルム、∞–ノルム に基づくものが、それぞれ RCONX(1)、RCONX(3)に格納される)。
ERBF1D(3)	out	D	公式1で計算した誤差限界(厳密条件数に基づく) ($=\kappa_1(A)\epsilon$)。(1–ノルム、∞–ノルム に基づくものが、それぞれ ERBF1D(1)、ERBF1D(3)に格納される)。
ERBF1X(3)	out	D	公式1で計算した誤差限界(近似条件数に基づく) ($=\kappa_1(A)\epsilon$)。(1–ノルム、∞–ノルム に基づくものが、それぞれ ERBF1X(1)、ERBF1X(3)に格納される)。
ERBF2D(3)	out	D	公式2で計算した誤差限界(厳密条件数に基づく) ($=\kappa_1(A)\frac{\|Ax-b\|_1}{\|b\|_1}$)。(1–ノルム、$\infty$–ノルム に基づくものが、それぞれ ERBF2D(1)、ERBF2D(3)に格納される)。
ERBF2X(3)	out	D	公式2で計算した誤差限界(近似条件数に基づく) ($=\kappa_1(A)\frac{\|Ax-b\|_1}{\|b\|_1}$)。(1–ノルム、$\infty$–ノルム に基づくものが、それぞれ ERBF2X(1)、ERBF2X(3)に格納される)。
IWORK(LDA)	w	I	作業用配列
DWORK(LDA*2)	w	D	作業用配列
XWORK(LDA*2)	w	Z	作業用配列
IRTN	out	I	リターンコード。 ＝0：正常終了。

データ構造に関する注意事項　本サブルーチンは配列データの並列処理分散メモリへのデータ配置として列に関するブロック・サイクリック分割を採用している。それ故、利用者はブロック・サイクリック列分割により行列等の配列データの分割を本サブルーチンの呼び出し前に行なっておく必要がある。

8.5 複素数一般密行列誤差評価(MPI重複ローカル型) JAERI_zgdzlhiom

8.5.1 機能

複素数一般密行列の連立一次方程式に対して係数行列の条件数および誤差限界を見積もる(MPI重複ローカル版)。本サブルーチンは主要入力データとして、行列、計算された解、右辺ベクトルをとり、主要出力データとして係数行列のノルム、逆行列のノルム、条件数、誤差限界を返す。ノルムは、1-ノルムと無限大ノルムの両方について求める。条件数と誤差限界も、1-ノルムと無限大ノルムの2通りに対応したものを求める。

本サブルーチンは分散メモリ型並列計算機での分割ローカル配置、すなわち各PEが行列の部分行列を持つという稼働環境を想定する。

8.5.2 コーリングシーケンス

```
   call JAERI_zgdzlhiom
 &      (mpicw ,nnG
 &       whohas,nclmL ,mapG2L,mapL2G,
 &       EPSMCH,
 &       LDA    ,ncmaxL       ,aL    ,XX    ,BB    ,
 &       RNRM1 ,RNRMI ,BNRM1 ,BNRMI ,
 &       ANRM1 ,ANRMI ,AIVNR1,AIVNRI,RCOND1,RCONDI,
 &       ERBDA1,ERBDAI,ERBDB1,ERBDBI,
 &       IWORK ,DWORK ,ZWORK ,
 &       IRTN  )
```

8.5.3 引数

用法: in　入力用。値は壊されない。
　　out　出力用。
　　i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。
　　w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)
　　D　倍精度実数型(double precision)
　　Z　倍精度複素数型(complex*16)
　　L　論理型(logical)
　　C　文字型(character * (*))

変数／配列(次元)	I/O	型	意味
mpicw	in	I	MPIコミュニケータ
nnG	in	I	連立一次方程式の次元を示す。$nnG \leq LDA$

変数／配列(次元)	I/O	型	意味
whohas(LDA)	in	I	元の行列であるグローバル行列の各列を処理担当するランク番号を格納する配列。グローバル行列を仮にAG(*,*)で表すとすると、そのj列目AG(*,j)を処理担当するランクの値$(=0,1,\cdots,PE数-1)$がwhohas(j)に格納されているものとする。whohasの内容は、全PE共通。
nclmL(0:PE数-1)	in	I	各ランク毎のローカル行列aLにおいて実際に処理対象となる列の数。ip$(=0,1,\cdots,PE数-1)$番目のランクが処理担当する列の数が、nclmL(ip)に格納されているものとする。nclmLの内容は、全PEについて共通。
mapG2L(LDA)	in	I	元の行列であるグローバル行列からローカル行列aL(*,*)への列の対応を定義する。グローバル行列を仮にAG(*,*)で表すとするとAGのjG番目の列が、aLのjL番目の列に格納されているとき、jL = mapG2L(jG)の関係にある。AGのjG番目の列が、自ランクのaLに存在しない場合は、-1が格納されているものとする。mapG2Lの内容は、PE毎に異なる。
mapL2G(ncmaxL)	in	I	ローカル配列aL(*,*)から、元の行列であるグローバル配列への列の対応を定義する。グローバル行列を仮にAG(*,*)で表すとするとaLのjL番目の列が、AGのjG番目の列に格納されているとき、jG = mapL2G(jL)の関係にある。mapL2Gの内容は、PE毎に異なる。
EPSMCH	in	I	倍精度実数のマシン・イプシロン$(=\epsilon)$。LAPACKに含まれる関数DLAMCH()などにより計算できる。
LDA	in	I	行列格納用配列aLの一次元目の宣言サイズ。
ncmaxL	in	I	行列格納用配列aLの二次元目の宣言サイズ。
aL(LDA,ncmaxL)	i/o	Z	複素数密行列。元の行列をブロック・サイクリック列分割したものを縮小した形で与える。 逆行列が戻値となる。
XX(LDA)	in	I	$Ax=b$の解x。
BB(LDA)	in	I	$Ax=b$の右辺ベクトルb。
RNRM(3)	out	D	残差ベクトル$Ax-b$の1−ノルムおよび∞−ノルム(それぞれ、RNRM(1)、RNRM(3)に格納される)。
BNRM(3)	out	D	右辺ベクトルbの1−ノルムおよび∞−ノルム(それぞれ、BNRM(1)、BNRM(3)に格納される)。

変数／配列(次元)	I/O	型	意味
ANRM(3)	out	D	係数行列 A の 1−ノルムおよび ∞−ノルム(それぞれ、AVNRD(1)、AVNRD(3) に格納される)。
AVNRD(3)	out	D	係数行列の逆行列 A^{-1}の厳密な 1−ノルムおよび ∞−ノルム(それぞれ、AVNRD(1)、AVNRD(3) に格納される)。
AVNRX(3)	out	D	係数行列の逆行列 A^{-1}の近似的な 1−ノルムおよび ∞−ノルム(それぞれ、AVNRX(1)、AVNRX(3) に格納される)。
RCOND(3)	out	D	厳密な条件数(1−ノルム、∞−ノルム に基づくものが、それぞれ RCOND(1)、RCOND(3) に格納される)。
RCONX(3)	out	D	近似的な条件数(1−ノルム、∞−ノルム に基づくものが、それぞれ RCONX(1)、RCONX(3) に格納される)。
ERBF1D(3)	out	D	公式1で計算した誤差限界(厳密条件数に基づく) (= $\kappa_1(A)\epsilon$)。(1−ノルム、∞−ノルム に基づくものが、それぞれ ERBF1D(1)、ERBF1D(3) に格納される)。
ERBF1X(3)	out	D	公式1で計算した誤差限界(近似条件数に基づく) (= $\kappa_1(A)\epsilon$)。(1−ノルム、∞−ノルム に基づくものが、それぞれ ERBF1X(1)、ERBF1X(3) に格納される)。
ERBF2D(3)	out	D	公式2で計算した誤差限界(厳密条件数に基づく) (= $\kappa_1(A)\frac{\|Ax-b\|_1}{\|b\|_1}$)。(1−ノルム、∞−ノルム に基づくものが、それぞれ ERBF2D(1)、ERBF2D(3) に格納される)。
ERBF2X(3)	out	D	公式2で計算した誤差限界(近似条件数に基づく) (= $\kappa_1(A)\frac{\|Ax-b\|_1}{\|b\|_1}$)。(1−ノルム、∞−ノルム に基づくものが、それぞれ ERBF2X(1)、ERBF2X(3) に格納される)。
IWORK(LDA)	w	I	作業用配列
DWORK(LDA*2)	w	D	作業用配列
XWORK(LDA*2)	w	Z	作業用配列
IRTN	out	I	リターンコード。 = 0 : 正常終了。

データ構造に関する注意事項　本サブルーチンは配列データの並列処理分散メモリへのデータ配置として列に関するブロック・サイクリック分割を採用している。それ故、利用者はブロック・サイクリック列分割により行列等の配列データの分割を本サブルーチンの呼び出し前に行なっておく必要がある。

8.6 複素数一般密行列誤差評価(逐次版) JAERI_zgdzlhiss

8.6.1 機能

複素数一般密行列の連立一次方程式に対して係数行列の条件数および誤差限界を見積もる(逐次版)。

本サブルーチンは主要入力データとして、行列、計算された解、右辺ベクトルをとり、主要出力データとして係数行列のノルム、逆行列のノルム、条件数、誤差限界を返す。

ノルムは、1-ノルムと無限大ノルムの両方について求める。

本サブルーチンは逐次処理計算機での使用を想定する(通常のベクトル型スーパーコンピューターを含む)。

8.6.2 コーリングシーケンス

```
    call JAERI_zgdzlhiss
   &     (NN     ,EPSMCH,
   &      LDA            ,AA     ,XX     ,BB     ,
   &      RNRM1  ,RNRMI  ,BNRM1  ,BNRMI  ,
   &      ANRM1  ,ANRMI  ,AVNRD1,AVNRDI,AVNRX1,AVNRXI,
   &                     ,RCOND1,RCONDI,RCONX1,RCONXI,
   &      ERBDA1,ERBDAI,ERBDB1,ERBDBI,
   &      IWORK  ,DWORK  ,ZWORK  ,
   &      IRTN           )
```

8.6.3 引数

用法: in　入力用。値は壊されない。
　　　out　出力用。
　　　i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。
　　　w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)
　　　D　倍精度実数型(double precision)
　　　Z　倍精度複素数型(complex*16)
　　　L　論理型(logical)
　　　C　文字型(character $*$ ($*$))

変数/配列(次元)	I/O	型	意味
NN	in	I	連立一次方程式の次元を示す。
EPSMCH	in	I	倍精度実数のマシン・イプシロン(= ϵ)。
LDA	in	I	行列格納用配列ALの一次元目の宣言サイズ。

変数／配列(次元)	I/O	型	意味
AA(LDA,*)	i/o	Z	複素数密行列。LU 分解されたものが戻り値となる。
XX(LDA)	in	I	$Ax=b$ の解 x。
BB(LDA)	in	I	$Ax=b$ の右辺ベクトル b。
RNRM1	out	D	残差ベクトル $Ax-b$ の1–ノルム。
RNRMI	out	D	残差ベクトル $Ax-b$ の∞–ノルム。
BNRM1	out	D	右辺ベクトル b の 1–ノルム。
BNRMI	out	D	右辺ベクトル b の ∞–ノルム。
ANRM1	out	D	係数行列 A の 1–ノルム ($=\|A\|_1$)。
ANRMI	out	D	係数行列 A の ∞–ノルム ($=\|A\|_\infty$)。
AVNRD1	out	D	逆行列 A^{-1} の厳密 1–ノルム ($=\|A^{-1}\|_1$)。
AVNRDI	out	D	逆行列 A^{-1} の厳密 ∞–ノルム ($=\|A^{-1}\|_\infty$)。
AVNRX1	out	D	逆行列 A^{-1} の近似 1–ノルム ($=\|A^{-1}\|_1$)。
AVNRXI	out	D	逆行列 A^{-1} の近似 ∞–ノルム ($=\|A^{-1}\|_\infty$)。
RCOND1	out	D	1–ノルムによる厳密条件数 ($\kappa_1(A)=\|A\|_1\|A^{-1}\|_1$)。
RCONDI	out	D	∞–ノルムによる厳密条件数 ($\kappa_\infty(A)=\|A\|_\infty\|A^{-1}\|_\infty$)。
RCONX1	out	D	1–ノルムによる近似条件数 ($\kappa_1(A)=\|A\|_1\|A^{-1}\|_1$)。
RCONXI	out	D	∞–ノルムによる近似条件数 ($\kappa_\infty(A)=\|A\|_\infty\|A^{-1}\|_\infty$)。
ERBDA1	out	D	公式1、厳密 1–ノルムによる誤差限界 ($=\kappa_1(A)\ \epsilon$)。
ERBDAI	out	D	公式1、厳密 ∞–ノルムによる誤差限界 ($=\kappa_\infty(A)\ \epsilon$)。
ERBDB1	out	D	公式2、厳密 1–ノルムによる誤差限界 ($=\kappa_1(A)\frac{\|Ax-b\|_1}{\|b\|_1}$)。
ERBDBI	out	D	公式2、厳密 ∞–ノルムによる誤差限界 ($=\kappa_\infty(A)\frac{\|Ax-b\|_\infty}{\|b\|_\infty}$)。
ERBXA1	out	D	公式1、近似 1–ノルムによる誤差限界 ($=\kappa_1(A)\ \epsilon$)。
ERBXAI	out	D	公式1、近似 ∞–ノルムによる誤差限界 ($=\kappa_\infty(A)\ \epsilon$)。
ERBXB1	out	D	公式2、近似 1–ノルムによる誤差限界 ($=\kappa_1(A)\frac{\|Ax-b\|_1}{\|b\|_1}$)。
ERBXBI	out	D	公式2、近似 ∞–ノルムによる誤差限界 ($=\kappa_\infty(A)\frac{\|Ax-b\|_\infty}{\|b\|_\infty}$)。
IWORK(LDA)	w	I	作業用配列
DWORK(LDA*2)	w	D	作業用配列
ZWORK(LDA*2)	w	Z	作業用配列
irtn	out	I	リターンコード。$=0$ のとき、正常終了。

データ構造に関する注意事項　特になし。

9. 行列の固有値誤差評価

9.1 実対称密行列固有値誤差評価(MPI分割ローカル版) JAERI_dsdzekkpm

9.1.1 機能

実対称密行列に対して固有値・固有ベクトルの誤差を評価する(MPI分割ローカル版)。

本サブルーチンは主要入力データとして、行列、計算された固有値・固有ベクトルをとり、主要出力データとして真の固有値の存在範囲、計算固有値・計算固有ベクトルの残差ノルム、計算固有ベクトルと真の固有ベクトルの間の鋭角の正弦の誤差範囲を返す。

評価公式はKorn-Katoの公式に基づく。誤差評価は全てのモードの固有値に対して行なわれる。よって利用者は全モードの固有値・固有ベクトルを、予め求めておく必要がある。

本サブルーチンは分散メモリ型並列計算機での分割ローカル配置、すなわち各プロセッサが行列の部分行列を持つという稼働環境を想定する。

9.1.2 コーリングシーケンス

```
      CALL JAERI_dsdzekkpm
     &     (mpicw,ndim,nn,n1,n2,iblock,llrank,lrank2
     &     ,a,x2,eig,ray,errm,errp,rn,tht,eps,idbg,iopt
     &     ,wk,iup,idw,zwk1,zwk2,x,ierr)
```

9.1.3 引数

用法: in　入力用。値は壊されない。
- out　出力用。
- i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。
- w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)
- D　倍精度実数型(double precision)
- Z　倍精度複素数型(complex*16)
- L　論理型(logical)
- C　文字型(character $*$ ($*$))

変数／配列(次元)	I/O	型	意味
mpicw	in	I	MPIコミュニケータ
ndim	in	I	行列の次数。
nn	in	I	行列格納用配列aの第二次元目のサイズ。
n1	in	I	配列の下限用。 n1=1と指定する。
n2	in	I	配列の上限用。 n2=ndimと指定する。
iblock	in	I	ブロック・サイクリック列分割におけるブロック長。
llrank	in	I	各プロセスが処理を担当する列の数。 =配列aに実際に格納されている列の数。
lrank2 (0:PE数-1)	in	I	各プロセスが処理を担当する列の数を格納する配列。 lrank2(i)はランクiのプロセスが処理を担当する列の数を表わす。lrank2(i) $\leq$ nn.
a(n1:n2, 1:nn)	in	D	実対称密行列。行列をブロック・サイクリック列分割して与える。配列の縮小を行なう。
x2(n1:n2, 1:nn)	in	D	計算固有ベクトル。 第二次元目の添字について行列と同じ分割方法でプロセスに分散して与える。配列の縮小を行なう。
eig(ndim)	in	D	計算固有値。値の昇順に格納されているものとする。 eig(1) $\leq$ eig(2) $\leq \ldots \leq$ eig(ndim). 配列全体を与える。データ分割や配列縮小は行なわない。
ray(ndim)	out	D	計算固有ベクトルに対するRayleigh商。 $\rho = (x, Ax)/(x, x)$
errm(ndim)	out	D	計算固有値の下方誤差範囲。$\Delta\lambda^-$
errp(ndim)	out	D	計算固有値の上方誤差範囲。$\Delta\lambda^+$ モード毎に、真の固有値の存在範囲を$(\rho - \Delta\lambda^-, \rho + \Delta\lambda^+)$ の形で求める。
rn(ndim)	out	D	残差ノルム。
tht(ndim)	out	D	計算固有ベクトルと真の固有ベクトルの間の鋭角の正弦の誤差範囲。$\sin\theta \leq$ tht.
eps	in	D	固有値の縮退の判別を行なうための微小量。入力データにおける二つの計算固有値λ_i, λ_jが$\lvert\lambda_i - \lambda_i\rvert \leq$ eps を満たす場合、本サブルーチンは固有値λ_iとλ_jは縮退していると扱う。変数epsの値として、利用者は例えば機種精度(マシン・イプシロン)を与えても良い。
idbg	in	I	実行中の情報メッセージの出力制御用フラグ。 =0: 情報メッセージの出力は行なわず、stop文によりプログラムが終了したときのみエラーメッセージを出力する。 =1: 情報メッセージの出力を行なう。
iopt	in	I	出力データの持ち方についてのフラグ =0: ランク0のプロセスのみが配列の全体を持つ; =1: 全てのプロセスが配列の全体を持つ.

変数／配列(次元)	I/O	型	意味
wk(0:ndim+1)	w	D	作業用配列
iup(ndim)	w	I	作業用配列
idw(ndim)	w	I	作業用配列
zwk1(4,ndim)	w	D	作業用配列
zwk2(4,ndim)	w	D	作業用配列
x(n1:n2, 1:nn)	w	D	作業用配列
ierr	out	I	エラーコード。 =0: 正常終了。

データ構造に関する注意事項　本サブルーチンは配列データの並列処理分散メモリへのデータ配置として列に関するブロック・サイクリック分割を採用している。それ故、利用者はブロック・サイクリック列分割により行列等の配列データの分割を本サブルーチンの呼び出し前に行なっておく必要がある。また、実対称行列は下三角成分のみまたは上三角成分のみではなく、全体を与える必要がある。本ルーチン内では、対称性のチェックは行わない。

9.2 実対称密行列固有値誤差評価(MPI重複ローカル版) JAERI_dsdzekkom

9.2.1 機能

実対称密行列に対して固有値・固有ベクトルの誤差を評価する(MPI重複ローカル版)。

本サブルーチンは主要入力データとして、行列、計算された固有値・固有ベクトルをとり、主要出力データとして真の固有値の存在範囲、計算固有値・計算固有ベクトルの残差ノルム、計算固有ベクトルと真の固有ベクトルの間の鋭角の正弦の誤差範囲を返す。

評価公式はKorn-Katoの公式に基づく。誤差評価は全てのモードの固有値に対して行なわれる。よって利用者は全モードの固有値・固有ベクトルを、予め求めておく必要がある。

本サブルーチンは分散メモリ型並列計算機での重複ローカル配置、すなわち各プロセッサが同じ行列の全体を重複して持ち、処理のみ分割して行うという稼働環境を想定する。

9.2.2 コーリングシーケンス

```
      CALL JAERI_dsdzekkom
     &     (mpicw,ndim,n1,n2,iblock
     &     ,a,x,eig,ray,errm,errp,rn,tht,eps,idbg,iopt
     &     ,wk,iup,idw,zwk1,zwk2,ierr)
```

9.2.3 引数

用法: in 入力用。値は壊されない。

out 出力用。

i/o 入出力用。入力として使用され、かつ戻り時に値が設定される。

w 作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I 整数型(integer)

D 倍精度実数型(double precision)

Z 倍精度複素数型(complex*16)

L 論理型(logical)

C 文字型(character * (*))

変数／配列(次元)	I/O	型	意味
mpicw	in	I	MPIコミュニケータ
ndim	in	I	行列の次数。
n1	in	I	配列の下限用。 n1=1と指定する。
n2	in	I	配列の上限用。 n2=ndimと指定する。
iblock	in	I	ブロック・サイクリック列分割におけるブロック長。
a(ndim,ndim)	in	D	実対称密行列。a(i,j) $\longleftarrow a_{i,j}$. 行列全体を与える。データ分割や配列縮小は行なわない。
x(ndim,ndim)	in	D	計算固有ベクトル。x(i,mode) $\longleftarrow \tilde{x}^{\mathrm{mode}}$. 規格化されていてもいなくても良い。 配列全体を与える。データ分割や配列縮小は行なわない。
eig(ndim)	in	D	計算固有値。値の昇順に格納されているものとする。 eig(1) $\leq$ eig(2) $\leq \ldots \leq$ eig(ndim). 配列全体を与える。データ分割や配列縮小は行なわない。
ray(ndim)	out	D	計算固有ベクトルに対するRayleigh商。 $\rho = (x, Ax)/(x, x)$
errm(ndim)	out	D	計算固有値の下方誤差範囲。$\Delta\lambda^-$
errp(ndim)	out	D	計算固有値の上方誤差範囲。$\Delta\lambda^+$ モード毎に、真の固有値の存在範囲を $(\rho - \Delta\lambda^-, \rho + \Delta\lambda^+)$ の形で求める。
rn(ndim)	out	D	残差ノルム。
tht(ndim)	out	D	計算固有ベクトルと真の固有ベクトルの間の鋭角の正弦の誤差範囲。$\sin\theta \leq$ tht.
eps	in	D	固有値の縮退の判別を行なうための微小量。入力データにおける二つの計算固有値λ_i, λ_jが$\lvert\lambda_i - \lambda_i\rvert \leq$ eps を満たす場合、本サブルーチンは固有値λ_iとλ_jは縮退していると扱う。変数epsの値として、利用者は例えば機種精度(マシン・イプシロン)を与えても良い。
idbg	in	I	実行中の情報メッセージの出力制御用フラグ。 =0: 情報メッセージの出力は行なわず、stop文によりプログラムが終了したときのみエラーメッセージを出力する。 =1: 情報メッセージの出力を行なう。
iopt	in	I	出力データの持ち方についてのフラグ =0: ランク0のプロセスのみが配列の全体を持つ; =1: 全てのプロセスが配列の全体を持つ.
wk(0:ndim+1)	w	D	作業用配列
iup(ndim)	w	I	作業用配列
idw(ndim)	w	I	作業用配列
zwk1(4,ndim)	w	D	作業用配列
zwk2(4,ndim)	w	D	作業用配列
ierr	out	I	エラーコード。=0 のとき、正常終了。

データ構造に関する注意事項 本サブルーチンでの配列データは、全プロセッサが全体を重複して持ち、処理のみをブロック・サイクリック列分割で割り当てられた範囲を分担する方式をとっている。それゆえ、ユーザは配列データの分割を行う必要はない。なお、実対称行列は下三角成分のみまたは上三角成分のみではなく、全体を与える必要がある。本ルーチン内では、対称性のチェックは行わない。

9.3 実対称密行列の固有値誤差評価(逐次版) JAERI_dsdzekkss

9.3.1 機能

実対称密行列に対して固有値・固有ベクトルの誤差を評価する(逐次版)。本サブルーチンは主要入力データとして、行列、計算された固有値・固有ベクトルをとり、主要出力データとして真の固有値の存在範囲、計算固有値・計算固有ベクトルの残差ノルム、計算固有ベクトルと真の固有ベクトルの間の鋭角の正弦の誤差範囲を返す。評価公式はKorn-Katoの公式に基づく。誤差評価は全てのモードの固有値に対して行なわれる。よって利用者は全モードの固有値・固有ベクトルを、予め求めておく必要がある。

本サブルーチンは逐次処理計算機での使用を想定する(通常のベクトル型スーパーコンピューターを含む)。

9.3.2 コーリングシーケンス

```
      CALL JAERI_dsdzekkss
     &     (ndim,a,x,eig,ray,errm,errp,rn,tht,eps,idbg,wk,iup,idw,ierr)
```

9.3.3 引数

用法:	in	入力用。値は壊されない。
	out	出力用。
	i/o	入出力用。入力として使用され、かつ戻り時に値が設定される。
	w	作業用。作業領域として与えられる。戻り時の値は意味を持たない。
型 :	I	整数型(integer)
	D	倍精度実数型(double precision)
	Z	倍精度複素数型(complex*16)
	L	論理型(logical)
	C	文字型(character $*$ ($*$))

変数／配列(次元)	I/O	型	意味
ndim	in	I	行列の次数。
a(ndim,ndim)	in	D	実対称密行列。a(i,j) $\longleftarrow a_{i,j}$.
x(ndim,ndim)	in	D	計算固有ベクトル。x(i,mode) $\longleftarrow \bar{x}^{\mathrm{mode}}$. 規格化されていてもいなくても良い。
eig(ndim)	in	D	計算固有値。値の昇順に格納されているものとする。 eig(1) $\le$ eig(2) $\le \ldots \le$ eig(ndim).
ray(ndim)	out	D	計算固有ベクトルに対するRayleigh商。 $\rho = (x, Ax)/(x, x)$
errm(ndim)	out	D	計算固有値の下方誤差範囲。$\Delta\lambda^-$
errp(ndim)	out	D	計算固有値の上方誤差範囲。$\Delta\lambda^+$ モード毎に、真の固有値の存在範囲を $(\rho - \Delta\lambda^-, \rho + \Delta\lambda^+)$ の形で求める。
rn(ndim)	out	D	残差ノルム。
tht(ndim)	out	D	計算固有ベクトルと真の固有ベクトルの間の鋭角の正弦の誤差範囲。$\sin\theta \le$ tht.
eps	in	D	固有値の縮退の判別を行なうための微小量。入力データにおける二つの計算固有値λ_i, λ_jが $\lvert\lambda_i - \lambda_i\rvert \le$ eps を満たす場合、本サブルーチンは固有値λ_iとλ_jは縮退していると扱う。変数epsの値として、利用者は例えば機種精度(マシン・イプシロン)を与えても良い。
idbg	in	I	実行中の情報メッセージの出力制御用フラグ。 =0: 情報メッセージの出力は行なわず、stop文によりプログラムが終了したときのみエラーメッセージを出力する。 =1: 情報メッセージの出力を行なう。
wk(0:ndim+1)	w	D	作業用配列
iup(ndim)	w	I	作業用配列
idw(ndim)	w	I	作業用配列
ierr	out	I	エラーコード。 =0: 正常終了。

データ構造に関する注意事項　利用者は全ての行列要素を与える必要がある。つまり、本ルーチンでは対称性に基づいて、下三角要素または上三角要素のみを参照して処理するわけではない。本ルーチン内では、対称性のチェックは行わない。

9.4 実対称帯行列固有値誤差評価(MPI分割ローカル版) JAERI_dsbzekkpm

9.4.1 機能

実対称帯行列に対して固有値・固有ベクトルの誤差を評価する(MPI分割ローカル版)。

本サブルーチンは主要入力データとして、行列、計算された固有値・固有ベクトルをとり、主要出力データとして真の固有値の存在範囲、計算固有値・計算固有ベクトルの残差ノルム、計算固有ベクトルと真の固有ベクトルの間の鋭角の正弦の誤差範囲を返す。

評価公式はKorn-Katoの公式に基づく。誤差評価は全てのモードの固有値に対して行なわれる。よって利用者は全モードの固有値・固有ベクトルを、予め求めておく必要がある。

本サブルーチンは分散メモリ型並列計算機での分割ローカル配置、すなわち各プロセッサが行列の部分行列を持つという稼働環境を想定する。

9.4.2 コーリングシーケンス

```
      CALL JAERI_dsbzekkpm
     &     (mpicw,ndim,ndimb,nn,n1,n2,iblock,lrank2
     &     ,a,x2,eig,ray,errm,errp,rn,tht,iaux,eps,idbg,iopt
     &     ,wk,iup,idw,zwk1,zwk2,x,ierr)
```

9.4.3 引数

用法: in　入力用。値は壊されない。
　　out　出力用。
　　i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。
　　w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)
　　D　倍精度実数型(double precision)
　　Z　倍精度複素数型(complex*16)
　　L　論理型(logical)
　　C　文字型(character * (*))

変数／配列(次元)	I/O	型	意味
`mpicw`	in	I	MPIコミュニケータ
`ndim`	in	I	行列の次数。
`ndimb`	in	I	帯行列をLAPACK形式で格納するための配列の行の数。
`nn`	in	I	行列格納用配列aの第二次元目のサイズ。
`n1`	in	I	配列の下限用。 `n1=1`と指定する。
`n2`	in	I	配列の上限用。 `n2=ndim`と指定する。
`iblock`	in	I	ブロック・サイクリック列分割におけるブロック長。
`lrank2(0:PE数-1)`	in	I	各プロセスが処理を担当する列の数を格納する配列。 `lrank2(i)`はランク`i`のプロセスが処理を担当する列の数を表わす。`lrank2(i)` ≤ nn.
`a(1:ndimb, 1:nn)`	in in	D	実対称帯行列。LAPACK形式による格納方法。 行列をブロック・サイクリック列分割して与える。配列の縮小を行なう。
`x2(n1:n2, 1:nn)`	in	D	計算固有ベクトル。 第二次元目の添字について行列と同じ分割方法でプロセスに分散して与える。配列の縮小を行なう。
`eig(ndim)`	in	D	計算固有値。値の昇順に格納されているものとする。 eig(1) ≤ eig(2) ≤ ... ≤ eig(ndim). 配列全体を与える。データ分割や配列縮小は行なわない。
`ray(ndim)`	out	D	計算固有ベクトルに対するRayleigh商。 $\rho = (x, Ax)/(x, x)$
`errm(ndim)`	out	D	計算固有値の下方誤差範囲。$\Delta\lambda^-$
`errp(ndim)`	out	D	計算固有値の上方誤差範囲。$\Delta\lambda^+$ モード毎に、真の固有値の存在範囲を $(\rho - \Delta\lambda^-, \rho + \Delta\lambda^+)$ の形で求める。
`rn(ndim)`	out	D	残差ノルム。
`tht(ndim)`	out	D	計算固有ベクトルと真の固有ベクトルの間の鋭角の正弦の誤差範囲。$\sin\theta \leq$ tht.
`iaux(10)`	in	I	iaux(1)=ku: 上帯幅 iaux(2)=kl: 下帯幅
`eps`	in	D	固有値の縮退の判別を行なうための微小量。入力データにおける二つの計算固有値λ_i, λ_jが$\lvert\lambda_i - \lambda_i\rvert \leq$ eps を満たす場合、本サブルーチンは固有値λ_iとλ_jは縮退していると扱う。変数epsの値として、利用者は例えば機種精度(マシン・イプシロン)を与えても良い。
`idbg`	in	I	実行中の情報メッセージの出力制御用フラグ。 =0: 情報メッセージの出力は行なわず、stop文によりプログラムが終了したときのみエラーメッセージを出力する。 =1: 情報メッセージの出力を行なう。

変数／配列(次元)	I/O	型	意味
iopt	in	I	出力データの持ち方についてのフラグ =0: ランク0のプロセスのみが配列の全体を持つ; =1: 全てのプロセスが配列の全体を持つ.
wk(0:ndim+1)	out	D	作業用配列
iup(ndim)	out	I	作業用配列
idw(ndim)	out	I	作業用配列
zwk1(4,ndim)	out	D	作業用配列
zwk2(4,ndim)	out	D	作業用配列
x(n1:n2, 1:nn)	out	D	作業用配列
ierr	out	I	エラーコード。 =0: 正常終了。

データ構造に関する注意事項　本サブルーチンの入力に備えて、利用者はLAPACKの形式での帯行列の格納および下帯幅k_l, 上帯幅k_uの値設定を行なう必要がある。次ではLAPACKの形式について説明する。([1] p.98)

帯幅k_lの対角下帯と帯幅k_uの対角上帯をもつ帯行列は$k_l + k_u + 1$個の行とn個の列をもつ二次元配列に詰めて格納する。行列の次数をnとしてより具体的にあらわすと、$\max(1, j - k_u) \leq i \leq \min(n, j + k_u)$なる$a_{ij}$を二次元配列$\mathtt{AB}(k_u + 1 + i - j, j)$に格納する。ここで二次元配列をABとした。

利用者は上記以外の配列要素を設定する必要はなく、また本サブルーチンでは参照されない。

本サブルーチンは帯幅k_l,k_uと行列の次数nとのあいだの整合性の検証はおこなわない。

利用者は入力データの行列にたいしてこのLAPACK形式の配列にブロック・サイクリック列分割を行ない、さらに配列の縮小を行なう必要がある。

9.5 実対称帯行列固有値誤差評価(MPI重複ローカル版) JAERI_dsbzekkom

9.5.1 機能

実対称帯行列に対して固有値・固有ベクトルの誤差を評価する(MPI重複ローカル版)。

本サブルーチンは主要入力データとして、行列、計算された固有値・固有ベクトルをとり、主要出力データとして真の固有値の存在範囲、計算固有値・計算固有ベクトルの残差ノルム、計算固有ベクトルと真の固有ベクトルの間の鋭角の正弦の誤差範囲を返す。

評価公式はKorn-Katoの公式に基づく。誤差評価は全てのモードの固有値に対して行なわれる。よって利用者は全モードの固有値・固有ベクトルを、予め求めておく必要がある。

本サブルーチンは分散メモリ型並列計算機での重複ローカル配置、すなわち各プロセッサが同じ行列の全体を重複して持ち、処理のみ分割して行うという稼働環境を想定する。

9.5.2 コーリングシーケンス

```
    CALL JAERI_dsbzekkom
  &      (mpicw,ndim,ndimb,n1,n2,iblock
  &      ,a,x,eig,ray,errm,errp,rn,tht,iaux,eps,idbg,iopt
  &      ,wk,iup,idw,zwk1,zwk2,ierr)
```

9.5.3 引数

用法: in 入力用。値は壊されない。

out 出力用。

i/o 入出力用。入力として使用され、かつ戻り時に値が設定される。

w 作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I 整数型(integer)

D 倍精度実数型(double precision)

Z 倍精度複素数型(complex*16)

L 論理型(logical)

C 文字型(character * (*))

変数／配列(次元)	I/O	型	意味
mpicw	in	I	MPIコミュニケータ
ndim	in	I	行列の次数。
ndimb	in	I	帯行列をLAPACK形式で格納するための配列の行の数。
n1	in	I	配列の下限用。 n1=1と指定する。
n2	in	I	配列の上限用。 n2=ndimと指定する。
iblock	in	I	ブロック・サイクリック列分割におけるブロック長。
a(ndim,ndim)	in	D	実対称帯行列。LAPACK形式による格納方法。 行列全体を与える。データ分割や配列縮小は行なわない。
x(ndim,ndim)	in	D	計算固有ベクトル。x(i,mode) $\longleftarrow \vec{x}^{\mathrm{mode}}$. 規格化されていてもいなくても良い。 配列全体を与える。データ分割や配列縮小は行なわない。
eig(ndim)	in	D	計算固有値。値の昇順に格納されているものとする。 eig(1) $\leq$ eig(2) $\leq \ldots \leq$ eig(ndim). 配列全体を与える。データ分割や配列縮小は行なわない。
ray(ndim)	out	D	計算固有ベクトルに対するRayleigh商。 $\rho = (x, Ax)/(x, x)$
errm(ndim)	out	D	計算固有値の下方誤差範囲。$\Delta\lambda^-$
errp(ndim)	out	D	計算固有値の上方誤差範囲。$\Delta\lambda^+$ モード毎に、真の固有値の存在範囲を $(\rho - \Delta\lambda^-, \rho + \Delta\lambda^+)$ の形で求める。
rn(ndim)	out	D	残差ノルム。
tht(ndim)	out	D	計算固有ベクトルと真の固有ベクトルの間の鋭角の正弦の誤差範囲。$\sin\theta \leq$ tht.
iaux(10)	in	I	iaux(1)=ku: 上帯幅 iaux(2)=kl: 下帯幅
eps	in	D	固有値の縮退の判別を行なうための微小量。入力データにおける二つの計算固有値λ_i, λ_jが$\lvert\lambda_i - \lambda_i\rvert \leq$ eps を満たす場合、本サブルーチンは固有値λ_iとλ_jは縮退していると扱う。変数epsの値として、利用者は例えば機種精度(マシン・イプシロン)を与えても良い。
idbg	in	I	実行中の情報メッセージの出力制御用フラグ。 =0: 情報メッセージの出力は行なわず、stop文によりプログラムが終了したときのみエラーメッセージを出力する。 =1: 情報メッセージの出力を行なう。

変数／配列(次元)	I/O	型	意味
`iopt`	in	I	出力データの持ち方についてのフラグ =0: ランク0のプロセスのみが配列の全体を持つ; =1: 全てのプロセスが配列の全体を持つ.
`wk(0:ndim+1)`	w	D	作業用配列
`iup(ndim)`	w	I	作業用配列
`idw(ndim)`	w	I	作業用配列
`zwk1(4,ndim)`	w	D	作業用配列
`zwk2(4,ndim)`	w	D	作業用配列
`ierr`	out	I	エラーコード。 =0: 正常終了。

データ構造に関する注意事項　利用者はLAPACKの形式での帯行列の格納および下帯幅k_l, 上帯幅k_uの値設定を行なう必要がある([1] p.98)

帯幅k_lの対角下帯と帯幅k_uの対角上帯をもつ帯行列は$k_l + k_u + 1$個の行とn個の列をもつ二次元配列に詰めて格納する。行列の次数をnとしてより具体的にあらわすと、$\max(1, j - k_u) \le i \le \min(n, j + k_u)$なる$a_{ij}$を二次元配列$\mathtt{AB}(k_u + 1 + i - j, j)$に格納する。ここで二次元配列をABとした。

利用者は上記以外の配列要素を設定する必要はなく、また本サブルーチンでは参照されない。

本サブルーチンは帯幅k_l,k_uと行列の次数nとのあいだの整合性の検証はおこなわない。

9.6 実対称帯行列の固有値誤差評価(逐次版) JAERI_dsbzekkss

9.6.1 機能

実対称帯行列に対して固有値・固有ベクトルの誤差を評価する(逐次版)。本サブルーチンは主要入力データとして、行列、計算された固有値・固有ベクトルをとり、主要出力データとして真の固有値の存在範囲、計算固有値・計算固有ベクトルの残差ノルム、計算固有ベクトルと真の固有ベクトルの間の鋭角の正弦の誤差範囲を返す。評価公式はKorn-Katoの公式に基づく。誤差評価は全てのモードの固有値に対して行なわれる。よって利用者は全モードの固有値・固有ベクトルを、予め求めておく必要がある。

本サブルーチンは逐次処理計算機での使用を想定する(通常のベクトル型スーパーコンピューターを含む)。

9.6.2 コーリングシーケンス

```
CALL JAERI_dsbzekkss
&(ndim,a,x,eig,ray,errm,errp,rn,tht,iaux,eps,idbg,wk,iup,idw,ierr)
```

9.6.3 引数

用法: in　入力用。値は壊されない。
　　out　出力用。
　　i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。
　　w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)
　　D　倍精度実数型(double precision)
　　Z　倍精度複素数型(complex*16)
　　L　論理型(logical)
　　C　文字型(character ∗ (∗))

変数／配列(次元)	I/O	型	意味
ndim	in	I	行列の次数。
a(ndim,ndim)	in	D	実対称帯行列。LAPACK形式による格納方法。
x(ndim,ndim)	in	D	計算固有ベクトル。x(i,mode) $\longleftarrow \vec{x}^{\mathrm{mode}}$. 規格化されていてもいなくても良い。
eig(ndim)	in	D	計算固有値。値の昇順に格納されているものとする。 eig(1) $\leq$ eig(2) $\leq \ldots \leq$ eig(ndim).
ray(ndim)	out	D	計算固有ベクトルに対するRayleigh商。 $\rho = (x, Ax)/(x, x)$
errm(ndim)	out	D	計算固有値の下方誤差範囲。$\Delta\lambda^-$
errp(ndim)	out	D	計算固有値の上方誤差範囲。$\Delta\lambda^+$ モード毎に、真の固有値の存在範囲を $(\rho - \Delta\lambda^-, \rho + \Delta\lambda^+)$ の形で求める。
rn(ndim)	out	D	残差ノルム。
tht(ndim)	out	D	計算固有ベクトルと真の固有ベクトルの間の鋭角の正弦の誤差範囲。$\sin\theta \leq$ tht.
iaux(10)	in	I	iaux(1)=ku: 上帯幅
	in		iaux(2)=kl: 下帯幅
eps	in	D	固有値の縮退の判別を行なうための微小量。入力データにおける二つの計算固有値λ_i, λ_jが$\lvert\lambda_i - \lambda_i\rvert \leq$ eps を満たす場合、本サブルーチンは固有値λ_iとλ_jは縮退していると扱う。変数epsの値として、利用者は例えば機種精度(マシン・イプシロン)を与えても良い。
idbg	in	I	実行中の情報メッセージの出力制御用フラグ。 =0: 情報メッセージの出力は行なわず、stop文によりプログラムが終了したときのみエラーメッセージを出力する。 =1: 情報メッセージの出力を行なう。
wk(0:ndim+1)	w	D	作業用配列
iup(ndim)	w	I	作業用配列
idw(ndim)	w	I	作業用配列
ierr	out	I	エラーコード。 =0 のとき、正常終了。

データ構造に関する注意事項　本サブルーチンの入力に備えて、利用者はLAPACKの形式での帯行列の格納および下帯幅k_l, 上帯幅k_uの値設定を行なう必要がある。次ではLAPACKの形式について説明する。([1] p.98)

帯幅k_lの対角下帯と帯幅k_uの対角上帯をもつ帯行列は$k_l + k_u + 1$個の行とn個の列をもつ二次元配列に詰めて格納する。行列の次数をnとしてより具体的にあらわすと、$\max(1, j - k_u) \leq i \leq \min(n, j + k_u)$ なるa_{ij}を二次元配列AB$(k_u + 1 + i - j, j)$ に格納する。ここで二次元配列をABとした。

利用者は上記以外の配列要素を設定する必要はなく、また本サブルーチンでは参照されない。

本サブルーチンは帯幅k_l,k_uと行列の次数nとのあいだの整合性の検証はおこなわない。

9.7 実対称疎行列固有値誤差評価 (MPI 分割ローカル版) JAERI_ds3zekkpm

9.7.1 機能

実対称疎行列に対して固有値・固有ベクトルの誤差を評価する (MPI 分割ローカル版)。

本サブルーチンは主要入力データとして、行列、計算された固有値・固有ベクトルをとり、主要出力データとして真の固有値の存在範囲、計算固有値・計算固有ベクトルの残差ノルム、計算固有ベクトルと真の固有ベクトルの間の鋭角の正弦の誤差範囲を返す。

評価公式は Korn-Kato の公式に基づく。誤差評価は全てのモードの固有値に対して行なわれる。よって利用者は全モードの固有値・固有ベクトルを、予め求めておく必要がある。

本サブルーチンは分散メモリ型並列計算機での分割ローカル配置、すなわち各プロセッサが行列の部分行列を持つという稼働環境を想定する。

9.7.2 コーリングシーケンス

```
      CALL JAERI_ds3zekkpm
     &     (mpicw,ndim,nc,ncmx
     &     ,n1,n2
     &     ,m1,m2,m3,a1,a2,a3,a4,a5,a6,a7,x2,eig
     &     ,ray,errm,errp,rn,tht,eps,idbg,iopt
     &     ,wk,wk2,iup,idw,zwk1,zwk2,istax,iendx,x,ierr)
```

9.7.3 引数

用法: in　入力用。値は壊されない。

　　　out　出力用。

　　　i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。

　　　w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型 (integer)

　　　D　倍精度実数型 (double precision)

　　　Z　倍精度複素数型 (complex*16)

　　　L　論理型 (logical)

　　　C　文字型 (character * (*))

変数／配列(次元)	I/O	型	意味
mpicw	in	I	MPIコミュニケータ
ndim	in	I	行列の次数。
nc(0:PE数-1)	in	I	各プロセスが処理する行の数を保持する配列。
ncmx	in	I	nc(0),...,nc(PE数 − 1) に関する最大値。
n1	in	I	配列の下限用。 n1=1 と指定する。
n2	in	I	配列の上限用。 n2=ndim と指定する。
m1	in	I	x 軸方向の節点数
m2	in	I	y軸方向の節点数
m3	in	I	z軸方向の節点数

以下のa1からa7の一次元配列は3次元7点差分公式による離散化から得られる実対称規則的疎行列を格納する配列である。

a1(1-m1*m2:*)	in	D	a1(i)
a2(1-m1*m2:*)	in	D	a2(i)
a3(1-m1*m2:*)	in	D	a3(i)
a4(1-m1*m2:*)	in	D	a4(i)
a5(1-m1*m2:*)	in	D	a5(i)
a6(1-m1*m2:*)	in	D	a6(i)
a7(1-m1*m2:*)	in	D	a7(i) m1*m2を前後に袖として持った幅nc(i)でブロック分割する。配列の縮小を行なう。添え字の取り得る値は、1-m1*m2 から nc(myrank)+m1*m2 までとする。

変数／配列(次元)	I/O	型	意味
x2(n1:n2, 1:ncmx)	in	D	計算固有ベクトル。 第二次元目の添字について行列と同じ分割方法でプロセスに分散して与える。ただし前後に幅m1*m2の袖領域は導入しない。配列の縮小を行なう。
eig(ndim)	in	D	計算固有値。値の昇順に格納されているものとする。 eig(1) $\leq$ eig(2) $\leq \ldots \leq$ eig(ndim). 配列全体を与える。データ分割や配列縮小は行なわない。
ray(ndim)	out	D	計算固有ベクトルに対するRayleigh商。 $\rho = (x, Ax)/(x, x)$
errm(ndim)	out	D	計算固有値の下方誤差範囲。$\Delta\lambda^-$
errp(ndim)	out	D	計算固有値の上方誤差範囲。$\Delta\lambda^+$ モード毎に、真の固有値の存在範囲を $(\rho - \Delta\lambda^-, \rho + \Delta\lambda^+)$ の形で求める。
rn(ndim)	out	D	残差ノルム。
tht(ndim)	out	D	計算固有ベクトルと真の固有ベクトルの間の鋭角の正弦の誤差範囲。$\sin\theta \leq$ tht.
eps	in	D	固有値の縮退の判別を行なうための微小量。入力データにおける二つの計算固有値λ_i, λ_jが$\|\lambda_i - \lambda_i\| \leq$ eps を満たす場合、本サブルーチンは固有値λ_iとλ_jは縮退していると扱う。変数epsの値として、利用者は例えば機種精度(マシン・イプシロン)を与えても良い。
idbg	in	I	実行中の情報メッセージの出力制御用フラグ。 =0: 情報メッセージの出力は行なわず、stop文によりプログラムが終了したときのみエラーメッセージを出力する。 =1: 情報メッセージの出力を行なう。
iopt	in	I	出力データの持ち方についてのフラグ
	in		=0: ランク0のプロセスのみが配列の全体を持つ;
	in		=1: 全てのプロセスが配列の全体を持つ.
wk(0:ndim+1)	w	D	作業用配列
wk2(1-m1*m2:ndim+m1*m2)	w	D	作業用配列
iup(ndim)	w	I	作業用配列
idw(ndim)	w	I	作業用配列
zwk1(4,ndim)	w	D	作業用配列
zwk2(4,ndim)	w	D	作業用配列
istax(0:PE数-1)	w	I	作業用配列
iendx(0:PE数-1)	w	I	作業用配列
x(n1:n2, 1:ncmx)	w	D	作業用配列
ierr	out	I	エラーコード。 =0: 正常終了。

データ構造に関する注意事項　本サブルーチンが入力とする実対称疎行列として、3次元7点差分公式による離散化から得られる規則的疎行列を想定する。ここではこの型の行列の格納方法を記述する。格納のための配列の宣言方法として、ベクトル計算の分野において効率的ベクトル計算を目的として用いられるような、行列の範囲外まで領域を確保して配列を宣言する方法を導入する。まず簡単のため、データを分割せず従って配列を縮小しない場合を述べる。これは逐次版・並列重複ローカル版の場合と同じである。

x, y, z方向の固定境界を除いた節点数をそれぞれm_1, m_2, m_3とする。節点はx, y, zの方向の順に通し番号で番号付けられていなければならない。この時の疎行列の次数は$n = m_1 \cdot m_2 \cdot m_3$である。非零要素は対角、および対角から$\pm 1$, $\pm m_1$, $\pm m_1 \cdot m_2$ 離れたところにある。

対角要素のベクトルをa4で表わす。対角から-1, $+1$離れた要素をそれぞれa3, a5で表わす。対角から$-m_1$, $+m_1$離れた要素をそれぞれa2, a6で表わす。対角から$-m_1 \cdot m_2$, $+m_1 \cdot m_2$離れた要素をそれぞれa1, a7で表わす。これらa1からa7は全て下限: $1 - m_1 \cdot m_2$, 上限: $n + m_1 \cdot m_2$ の一次元配列として宣言されなければならない。元行列のi行j列要素を$a_{i,j}$とすると、各要素の対応は次のようになる。

$$
\begin{aligned}
\mathtt{a1}(i) &\longleftarrow a_{i,i-m_1 \cdot m_2}, \\
\mathtt{a2}(i) &\longleftarrow a_{i,i-m_1}, \\
\mathtt{a3}(i) &\longleftarrow a_{i,i-1}, \\
\mathtt{a4}(i) &\longleftarrow a_{i,i}, \\
\mathtt{a5}(i) &\longleftarrow a_{i,i+1}, \\
\mathtt{a6}(i) &\longleftarrow a_{i,i+m_1}, \\
\mathtt{a7}(i) &\longleftarrow a_{i,i+m_1 \cdot m_2}.
\end{aligned}
\tag{12}
$$

次に、データを分割し、さらに配列を縮小する場合を述べる。$n = m_1 \cdot m_2 \cdot m_3$個をnprocs個のプロセスに割り振るとして、その個数を配列要素nc(0), nc(1), ..., nc(nprocs-1)に格納する。差分の隣接領域を袖として前後の持つことにすると、各プロセスでの処理対象となる縮小された配列a1等の領域は、ランクiのプロセスにたいして、下限: 1-m1*m2, 上限: nc(i)+m1*m2 となる。配列要素のうち、行列の範囲外の部分すなわち余分に確保された部分は本サブルーチン内でゼロクリアされる。

9.8 実対称疎行列固有値誤差評価(MPI重複ローカル版) JAERI_ds3zekkom

9.8.1 機能

実対称疎行列に対して固有値・固有ベクトルの誤差を評価する(MPI重複ローカル版)。

本サブルーチンは主要入力データとして、行列、計算された固有値・固有ベクトルをとり、主要出力データとして真の固有値の存在範囲、計算固有値・計算固有ベクトルの残差ノルム、計算固有ベクトルと真の固有ベクトルの間の鋭角の正弦の誤差範囲を返す。

評価公式はKorn-Katoの公式に基づく。誤差評価は全てのモードの固有値に対して行なわれる。よって利用者は全モードの固有値・固有ベクトルを、予め求めておく必要がある。

本サブルーチンは分散メモリ型並列計算機での重複ローカル配置、すなわち各プロセッサが同じ行列の全体を重複して持ち、処理のみ分割して行うという稼働環境を想定する。

9.8.2 コーリングシーケンス

```
      CALL JAERI_ds3zekkom
     &     (mpicw,ndim,nc
     &     ,m1,m2,m3,a1,a2,a3,a4,a5,a6,a7,x,eig
     &     ,ray,errm,errp,rn,tht,eps,idbg,iopt
     &     ,wk,wk2,iup,idw,zwk1,zwk2,istax,iendx,ierr)
```

9.8.3 引数

用法: in　入力用。値は壊されない。
out　出力用。
i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。
w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)
D　倍精度実数型(double precision)
Z　倍精度複素数型(complex*16)
L　論理型(logical)
C　文字型(character * (*))

変数／配列(次元)	I/O	型	意味
mpicw	in	I	MPIコミュニケータ
ndim	in	I	行列の次数。
nc(0:PE数-1)	in	I	各プロセスが処理する行の数を保持する配列。
m1	in	I	x軸方向の節点数
m2	in	I	y軸方向の節点数
m3	in	I	z軸方向の節点数

以下のa1からa7の一次元配列は3次元7点差分公式による離散化から得られる実対称規則的疎行列を格納する配列である。

a1(1-m1*m2:ndim+m1*m2)	in	D	a1$(i) \longleftarrow a_{i,i-m_1 \cdot m_2}$
a2(1-m1*m2:ndim+m1*m2)	in	D	a2$(i) \longleftarrow a_{i,i-m_1}$
a3(1-m1*m2:ndim+m1*m2)	in	D	a3$(i) \longleftarrow a_{i,i-1}$
a4(1-m1*m2:ndim+m1*m2)	in	D	a4$(i) \longleftarrow a_{i,i}$
a5(1-m1*m2:ndim+m1*m2)	in	D	a5$(i) \longleftarrow a_{i,i+1}$
a6(1-m1*m2:ndim+m1*m2)	in	D	a6$(i) \longleftarrow a_{i,i+m_1}$
a7(1-m1*m2:ndim+m1*m2)	in	D	a7$(i) \longleftarrow a_{i,i+m_1 \cdot m_2}$ 配列全体を与える。データ分割や配列縮小は行なわない。
x(ndim,ndim)	in	D	計算固有ベクトル。x(i,mode) $\longleftarrow \tilde{x}^{\mathrm{mode}}$. 規格化されていてもいなくても良い。 配列全体を与える。データ分割や配列縮小は行なわない。
eig(ndim)	in	D	計算固有値。値の昇順に格納されているものとする。 eig(1) $\leq$ eig(2) $\leq \ldots \leq$ eig(ndim). 配列全体を与える。データ分割や配列縮小は行なわない。
ray(ndim)	out	D	計算固有ベクトルに対するRayleigh商。 $\rho = (x, Ax)/(x, x)$
errm(ndim)	out	D	計算固有値の下方誤差範囲。$\Delta\lambda^-$
errp(ndim)	out	D	計算固有値の上方誤差範囲。$\Delta\lambda^+$ モード毎に、真の固有値の存在範囲を $(\rho - \Delta\lambda^-, \rho + \Delta\lambda^+)$ の形で求める。
rn(ndim)	out	D	残差ノルム。
tht(ndim)	out	D	計算固有ベクトルと真の固有ベクトルの間の鋭角の正弦の誤差範囲。$\sin\theta \leq$ tht.
eps	in	D	固有値の縮退の判別を行なうための微小量。入力データにおける二つの計算固有値λ_i, λ_jが$\lvert\lambda_i - \lambda_i\rvert \leq$ eps を満たす場合、本サブルーチンは固有値λ_iとλ_jは縮退していると扱う。変数epsの値として、利用者は例えば機種精度(マシン・イプシロン)を与えても良い。
idbg	in	I	実行中の情報メッセージの出力制御用フラグ。 =0: 情報メッセージの出力は行なわず、stop文によりプログラムが終了したときのみエラーメッセージを出力する。 =1: 情報メッセージの出力を行なう。

変数／配列(次元)	I/O	型	意味
iopt	in	I	出力データの持ち方についてのフラグ =0: ランク0のプロセスのみが配列の全体を持つ; =1: 全てのプロセスが配列の全体を持つ.
wk(0:ndim+1)	w	D	作業用配列
wk2(1-m1*m2:ndim+m1*m2)	w	D	作業用配列
iup(ndim)	w	I	作業用配列
idw(ndim)	w	I	作業用配列
zwk1(4,ndim)	w	D	作業用配列
zwk2(4,ndim)	w	D	作業用配列
istax(0:PE数-1)	w	I	作業用配列
iendx(0:PE数-1)	w	I	作業用配列
ierr	out	I	エラーコード。 =0: 正常終了。

データ構造に関する注意事項　本サブルーチンでの配列データは、全プロセッサが全体を重複して持ち、処理のみをブロック・サイクリック列分割で割り当てられた範囲を分担する方式をとっている。それゆえ、ユーザは配列データの分割を行う必要はない。なお、実対称疎行列は上半分のみまたは下半分のみではなく、全体を与える必要がある。本ルーチン内では、対称性のチェックは行わない。

9.9 実対称疎行列の固有値誤差評価(逐次版) JAERI_ds3zekkss

9.9.1 機能

実対称疎行列に対して固有値・固有ベクトルの誤差を評価する(逐次版)。本サブルーチンは主要入力データとして、行列、計算された固有値・固有ベクトルをとり、主要出力データとして真の固有値の存在範囲、計算固有値・計算固有ベクトルの残差ノルム、計算固有ベクトルと真の固有ベクトルの間の鋭角の正弦の誤差範囲を返す。評価公式はKorn-Katoの公式に基づく。誤差評価は全てのモードの固有値に対して行なわれる。よって利用者は全モードの固有値・固有ベクトルを、予め求めておく必要がある。

本サブルーチンは逐次処理計算機での使用を想定する(通常のベクトル型スーパーコンピューターを含む)。

9.9.2 コーリングシーケンス

```
      CALL JAERI_ds3zekkss
     &      (ndim,m1,m2,m3,a1,a2,a3,a4,a5,a6,a7,x,eig
     &      ,ray,errm,errp,rn,tht,eps,idbg
     &      ,wk,wk2,iup,idw,ierr)
```

9.9.3 引数

用法: in 入力用。値は壊されない。

out 出力用。

i/o 入出力用。入力として使用され、かつ戻り時に値が設定される。

w 作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I 整数型(integer)

D 倍精度実数型(double precision)

Z 倍精度複素数型(complex*16)

L 論理型(logical)

C 文字型(character * (*))

変数／配列(次元)	I/O	型	意味
ndim	in	I	行列の次数。

以下のa1からa7の一次元配列は3次元7点差分公式による離散化から得られる実対称規則的疎行列を格納する配列である。

m1	in	I	x軸方向の節点数
m2	in	I	y軸方向の節点数
m3	in	I	z軸方向の節点数
a1(1-m1*m2:ndim+m1*m2)	in	D	a1(i) $\longleftarrow a_{i,i-m_1\cdot m_2}$
a2(1-m1*m2:ndim+m1*m2)	in	D	a2(i) $\longleftarrow a_{i,i-m_1}$
a3(1-m1*m2:ndim+m1*m2)	in	D	a3(i) $\longleftarrow a_{i,i-1}$
a4(1-m1*m2:ndim+m1*m2)	in	D	a4(i) $\longleftarrow a_{i,i}$
a5(1-m1*m2:ndim+m1*m2)	in	D	a5(i) $\longleftarrow a_{i,i+1}$
a6(1-m1*m2:ndim+m1*m2)	in	D	a6(i) $\longleftarrow a_{i,i+m_1}$
a7(1-m1*m2:ndim+m1*m2)	in	D	a7(i) $\longleftarrow a_{i,i+m_1\cdot m_2}$
x(ndim,ndim)	in	D	計算固有ベクトル。x(i,mode) $\longleftarrow \tilde{x}^{\mathrm{mode}}$. 規格化されていてもいなくても良い。
eig(ndim)	in	D	計算固有値。値の昇順に格納されているものとする。 eig(1) $\leq$ eig(2) $\leq \ldots \leq$ eig(ndim).
ray(ndim)	out	D	計算固有ベクトルに対するRayleigh商。 $\rho = (x, Ax)/(x, x)$
errm(ndim)	out	D	計算固有値の下方誤差範囲。$\Delta\lambda^-$
errp(ndim)	out	D	計算固有値の上方誤差範囲。$\Delta\lambda^+$ モード毎に、真の固有値の存在範囲を $(\rho - \Delta\lambda^-, \rho + \Delta\lambda^+)$ の形で求める。
rn(ndim)	out	D	残差ノルム。
tht(ndim)	out	D	計算固有ベクトルと真の固有ベクトルの間の鋭角の正弦の誤差範囲。$\sin\theta \leq$ tht.
eps	in	D	固有値の縮退の判別を行なうための微小量。入力データにおける二つの計算固有値λ_i, λ_jが$\lvert\lambda_i - \lambda_i\rvert \leq$ eps を満たす場合、本サブルーチンは固有値λ_iとλ_jは縮退していると扱う。変数epsの値として、利用者は例えば機種精度(マシン・イプシロン)を与えても良い。
idbg	in	I	実行中の情報メッセージの出力制御用フラグ。 =0: 情報メッセージの出力は行なわず、stop文によりプログラムが終了したときのみエラーメッセージを出力する。 =1: 情報メッセージの出力を行なう。
wk(0:ndim+1)	w	D	作業用配列
wk2(1-m1:ndim+m1)	w	D	作業用配列
iup(ndim)	w	I	作業用配列
idw(ndim)	w	I	作業用配列
ierr	out	I	エラーコード。=0 のとき、正常終了。

データ構造に関する注意事項　利用者は疎行列における全ての有効な要素を与える必要がある。つまり、本ルーチンでは対称性に基づいて、上半分または下半分のみを参照して処理するわけではない。本ルーチン内では、対称性のチェックは行わない。

9.10 エルミート密行列固有値誤差評価(MPI分割ローカル版) JAERI_zhdzekkpm

9.10.1 機能

エルミート密行列に対して固有値・固有ベクトルの誤差を評価する。

本サブルーチンは主要入力データとして、行列、計算された固有値・固有ベクトルをとり、主要出力データとして真の固有値の存在範囲、計算固有値・計算固有ベクトルの残差ノルム、計算固有ベクトルと真の固有ベクトルの間の鋭角の正弦の誤差範囲を返す。

評価公式はKorn-Katoの公式に基づく。誤差評価は全てのモードの固有値に対して行なわれる。よって利用者は全モードの固有値・固有ベクトルを、予め求めておく必要がある。

本サブルーチンは分散メモリ型並列計算機での分割ローカル配置、すなわち各プロセッサが行列の部分行列を持つという稼働環境を想定する。

9.10.2 コーリングシーケンス

```
      CALL JAERI_dhdzekkpm
     &      (mpicw,NDIM,LDA,ncmaxL,iblock,llrank,lrank2,
     &       aL,evecL,EIG,RAY,ERRM,ERRP,RN,THT,EPS,IDBG,iopt,
     &       WK,IUP,IDW,WK1,WK2,evecLw,ierr)
```

9.10.3 引数

用法: in　入力用。値は壊されない。
　　out　出力用。
　　i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。
　　w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)
　　D　倍精度実数型(double precision)
　　Z　倍精度複素数型(complex*16)
　　L　論理型(logical)
　　C　文字型(character * (*))

変数／配列(次元)	I/O	型	意味
mpicw	in	I	MPIコミュニケータ
NDIM	in	I	行列の次数。
LDA	in	I	行列格納用配列aLの一次元目の宣言サイズ。
ncmaxL	in	I	行列格納用配列aLの二次元目の宣言サイズ。
iblock	in	I	ブロック・サイクリック列分割におけるブロック長(列幅)。
llrank	in	I	各プロセスが処理を担当する列の数。 すなわち、配列aLに実際に格納されている列の数。
lrank2(0:PE数-1)	in	I	各プロセスが処理を担当する列の数を格納する配列。 lrank2(i)＝ ランクiのプロセスが処理を担当する列の数 lrank2(i) $\leq$ nn.
aL(LDA,ncmaxL)	in	Z	エルミート密行列。元の行列をブロック・サイクリック列分割した上で、縮小して与える。
EVECL(LDA,ncmaxL)	in	D	計算固有ベクトル。第二次元目の添字について行列と同じ分割方法でプロセスに分散して与える。配列の縮小を行なう。
EIG(LDA)	in	D	計算固有値。値の昇順に格納されているものとする。 EIG(1) $\leq$ EIG(2) $\leq \ldots \leq$ EIG(NDIM). 配列全体を与える。データ分割や配列縮小は行なわない。
RAY(LDA)	out	D	計算固有ベクトルに対するRayleigh商。 $\rho = (x, Ax)/(x, x)$
ERRM(LDA)	out	D	計算固有値の下方誤差範囲 $\Delta\lambda^-$ 。
ERRP(LDA)	out	D	計算固有値の上方誤差範囲 $\Delta\lambda^+$ 。モード毎に、真の固有値の存在範囲を $(\rho - \Delta\lambda^-, \rho + \Delta\lambda^+)$ の形で求める。
RN (LDA)	out	D	残差ノルム。
THT(LDA)	out	D	固有ベクトルの計算値と真値のなす鋭角の正弦の誤差範囲。$\sin\theta \leq$ THT.
EPS	in	D	固有値の縮退の判別を行なうための微小量。入力データにおける二つの計算固有値λ_i, λ_j が $\lvert\lambda_i - \lambda_i\rvert \leq$ eps を満たす場合、本サブルーチンは固有値λ_iとλ_jは縮退しているものとみなす。具体的には、使用する機種のマシン・イプシロンの値を与えればよい。

変数／配列(次元)	I/O	型	意味
IDBG	in	I	実行中の情報メッセージの出力制御用フラグ。 =0: 情報メッセージの出力は行なわず、stop 文によりプログラムが終了したときのみエラーメッセージを出力する。 =1: 情報メッセージの出力を行なう。
iopt	in	I	出力データの持ち方についてのフラグ =0: ランク0のプロセスのみが配列の全体を持つ; =1: 全てのプロセスが配列の全体を持つ.
WK(0:LDA+1)	w	D	作業用配列
IUP(LDA)	w	I	作業用配列
IDW(LDA)	w	I	作業用配列
WK1(4,LDA)	w	D	作業用配列
WK2(4,LDA)	w	D	作業用配列
evecLw(LDA,ncmaxL)	w	Z	作業用配列
ierr	out	I	エラーコード。 =0: 正常終了。

データ構造に関する注意事項　本サブルーチンは配列データの並列処理分散メモリへのデータ配置として列に関するブロック・サイクリック分割を採用している。それ故、利用者はブロック・サイクリック列分割により行列等の配列データの分割を本サブルーチンの呼び出し前に行なっておく必要がある。また、エルミート行列は下三角成分のみまたは上三角成分のみではなく、全体を与える必要がある。本ルーチン内では、エルミート性のチェックは行わない。

9.11 エルミート密行列の固有値誤差評価 (逐次版) JAERI_zhdzekkss

9.11.1 機能

エルミート密行列に対して固有値・固有ベクトルの誤差を評価する。本サブルーチンは主要入力データとして、行列、計算された固有値・固有ベクトルをとり、主要出力データとして真の固有値の存在範囲、計算固有値・計算固有ベクトルの残差ノルム、計算固有ベクトルと真の固有ベクトルの間の鋭角の正弦の誤差範囲を返す。評価公式はKorn-Katoの公式に基づく。誤差評価は全てのモードの固有値に対して行なわれる。よって利用者は全モードの固有値・固有ベクトルを、予め求めておく必要がある。

本サブルーチンは逐次処理計算機での使用を想定する (通常のベクトル型スーパーコンピューターを含む)。

9.11.2 コーリングシーケンス

```
      CALL JAERI_zhdzekkss
     &     (        NN     ,LDA    ,AA     ,EVAL   ,EVEC   ,
     &      ZWORK ,
     &      RAY    ,EVALLW,EVALHI,RN     ,THT    ,IRTNCD)
```

9.11.3 引数

用法: in　入力用。値は壊されない。

out　出力用。

i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。

w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型 (integer)

D　倍精度実数型 (double precision)

Z　倍精度複素数型 (complex*16)

L　論理型 (logical)

C　文字型 (character * (*))

変数／配列(次元)	I/O	型	意味
NN	in	I	行列の次数。
LDA	in	I	行列格納用配列AAの宣言サイズ。
AA(LDA,LDA)	in	Z	エルミート密行列。 元の行列をブロック・サイクリック列分割したものを縮小して与えたもの。
EVAL(LDA,LDA)	in	D	計算固有ベクトル。 EVAL(i,mode) $\longleftarrow \vec{x}^{\mathrm{mode}}$. 正規化の有無は問わない。
EVEC(LDA)	in	D	計算固有値。値の昇順に格納されているものとする。 $\mathtt{EIG}(1) \leq \mathtt{EIG}(2) \leq \ldots \leq \mathtt{EIG}(\mathtt{NN})$.
ZWORK(LDA)	w	Z	作業用配列
RAY(LDA)	out	D	計算固有ベクトルに対するRayleigh商。 $\rho = (x, Ax)/(x, x)$
EVALLW(LDA)	out	D	計算固有値の下方誤差範囲$\Delta\lambda^-$ 。
EVALHI(LDA)	out	D	計算固有値の上方誤差範囲$\Delta\lambda^+$ 。モード毎に、 真の固有値の存在範囲を$(\rho - \Delta\lambda^-, \rho + \Delta\lambda^+)$ の形で求める。
RN(LDA)	out	D	残差ノルム$(= \max\|Ax - \lambda x\|)$。
THT(LDA)	out	D	固有ベクトルの真値と計算値のなす鋭角の正弦の誤差範囲 $\sin\theta \leq \mathtt{THT}$.
IRTNCD	out	I	エラーコード。 ＝ 0 : 正常終了。

データ構造に関する注意事項　利用者は全ての行列要素を与える必要がある。つまり、本ルーチンではエルミート性に基づいて、下三角行列または上三角要素のみを参照して処理するわけではない。本ルーチン内では、エルミート性のチェックは行わない。

10. 数学研究に現れる行列生成ライブラリ

10.1 ヒルベルト行列の生成　JAERI_dgdzqHLss

10.1.1 機能

本サブルーチンはヒルベルト行列とそのヒルベルト行列の逆行列を返す。

ヒルベルト行列とは行列成分が以下のように表される行列である。

$$A_{i,j} = \frac{1}{i+j-1}$$

10.1.2 コーリングシーケンス

```
   call JAERI_dgdzqHLss
  &     (n,n1,  x,nx,y,ny,org,inv,eig,A,Ainv,Eigen,wk,INFO)
```

10.1.3 引数

用法: in　入力用。値は壊されない。

out　出力用。

i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。

w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)

D　倍精度実数型(double precision)

Z　倍精度複素数型(complex*16)

L　論理型(logical)

C　文字型(character $*$ ($*$))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数 (n1 ≧ n ≧ 1)
n1	in	I	配列A,Ainvの配列宣言での行数 (n1 ≧ 1)
x	in	I	最初の行。A(1,*)はHI(x,*)をあらわす ただし、HI(*,*)は、次数nのヒルベルト行列
nx	in	I	行列の行の数 (n ≧ x + nx - 1)
y	in	I	最初の列。A(*,1)はHI(*,y)をあらわす ただし、HI(*,*)は、次数nのヒルベルト行列
ny	in	I	行列の列の数 (n ≧ y + ny - 1)
org	in	L	TRUE ： ヒルベルト行列中の指定された部分を返す FALSE ： ヒルベルト行列中の指定された部分を返さない
inv	in	L	TRUE ： 逆行列中の指定された部分を返す FALSE ： 逆行列中の指定された部分を返さない
eig	in	L	FALSE ： 固有値を返さない
A(n1,*)	out	D	ヒルベルト行列中の指定された部分
Ainv(n1,*)	out	D	逆行列中の指定された部分
Eigen(*)	out	D	使用しない (* ≧ n)
wk(*)	w	D	作業用配列 (* ≧ n + 1)
INFO	out	I	＝0：正常終了 ＝1：引数が不適当である。

10.2 パスカル行列の生成　JAERI_dgdzqPPss

10.2.1 機能

本サブルーチンはパスカル行列とそのパスカル行列の逆行列と固有値を返す。

パスカル行列とは行列成分が以下のように表される行列である。

$$P_{i,j} = \begin{cases} {}_{i-1}C_{j-1} & (i \geq j) \\ 0 & (i < j) \end{cases}$$

10.2.2 コーリングシーケンス

```
      call JAERI_dgdzqPPss
     &     (n,n1,  x,nx,y,ny,org,inv,eig,A,Ainv,Eigen,wk,INFO)
```

10.2.3 引数

用法: in　入力用。値は壊されない。

out　出力用。

i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。

w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)

D　倍精度実数型(double precision)

Z　倍精度複素数型(complex*16)

L　論理型(logical)

C　文字型(character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数 (n1 ≧ n ≧ 1)
n1	in	I	配列A,Ainvの配列宣言での行数 (n1 ≧ 1)
x	in	I	最初の行。A(1,*)はPS(x,*)をあらわす ただし、PS(*,*)は、次数nのパスカル行列
nx	in	I	行列の行の数 (n ≧ x + nx - 1)
y	in	I	最初の列。A(*,1)はPS(*,y)をあらわす ただし、PS(*,*)は、次数nのパスカル行列
ny	in	I	行列の列の数 (n ≧ y + ny - 1)
org	in	L	TRUE ：パスカル行列中の指定された部分を返す FALSE ：パスカル行列中の指定された部分を返さない
inv	in	L	TRUE ：逆行列中の指定された部分を返す FALSE ：逆行列中の指定された部分を返さない
eig	in	L	TRUE ：固有値を返す FALSE ：固有値を返さない
A(n1,*)	out	D	パスカル行列中の指定された部分
Ainv(n1,*)	out	D	逆行列中の指定された部分
Eigen(*)	out	D	行列の固有値 (* ≧ n)
wk(*)	w	D	作業用配列 (* ≧ n)
INFO	out	I	＝0：正常終了 ＝1：引数が不適当である。

10.3 パスカルのQ行列の生成 JAERI_dgdzpqss

10.3.1 機能

本サブルーチンはパスカルのQ行列とそのパスカルのQ行列の逆行列を返す。
パスカルのQ行列とは行列成分が以下のように表される行列である。

$$Q_{i,j} = \frac{(i+j-2)!}{(i-1)!(j-1)!}$$

10.3.2 コーリングシーケンス

```
      call JAERI_dgdzpqss
     &      (n,n1,  x,nx,y,ny,org,inv,eig,A,Ainv,Eigen,wk,INFO)
```

10.3.3 引数

用法: in 入力用。値は壊されない。
out 出力用。
i/o 入出力用。入力として使用され、かつ戻り時に値が設定される。
w 作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I 整数型(integer)
D 倍精度実数型(double precision)
Z 倍精度複素数型(complex*16)
L 論理型(logical)
C 文字型(character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数 (n1 ≧ n ≧ 1)
n1	in	I	配列A,Ainvの配列宣言での行数 (n1 ≧ 1)
x	in	I	最初の行。A(1,*)はPq(x,*)をあらわす ただし、Pq(*,*)は、次数nのパスカルのQ行列
nx	in	I	行列の行の数 (n ≧ x + nx - 1)
y	in	I	最初の列。A(*,1)はPq(*,y)をあらわす ただし、Pq(*,*)は、次数nのパスカルのQ行列
ny	in	I	行列の列の数 (n ≧ y + ny - 1)
org	in	L	TRUE ：パスカルのQ行列中の指定された部分を返す FALSE ：パスカルのQ行列中の指定された部分を返さない
inv	in	L	TRUE ：逆行列中の指定された部分を返す FALSE ：逆行列中の指定された部分を返さない
eig	in	L	FALSE ：固有値を返さない
A(n1,*)	out	D	パスカルのQ行列中の指定された部分
Ainv(n1,*)	out	D	逆行列中の指定された部分
Eigen(*)	out	D	使用しない (* ≧ n)
wk(*)	w	D	作業用配列 (* ≧ n)
INFO	out	I	＝0：正常終了 ＝1：引数が不適当である。

10.4 フランク行列の生成 JAERI_dgdzqFRss

10.4.1 機能

本サブルーチンはフランク行列とそのフランク行列の固有値を返す。
フランク行列行列とは行列成分が以下のように表される行列である。

$A_{i,j} = n + 1 - \max(i, j)$

10.4.2 コーリングシーケンス

```
    call JAERI_dgdzqFRss
   &     (n,n1,  x,nx,y,ny,org,inv,eig,A,Ainv,Eigen,INFO)
```

10.4.3 引数

用法: in 入力用。値は壊されない。
out 出力用。
i/o 入出力用。入力として使用され、かつ戻り時に値が設定される。
w 作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I 整数型 (integer)
D 倍精度実数型 (double precision)
Z 倍精度複素数型 (complex*16)
L 論理型 (logical)
C 文字型 (character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数 (n1 ≧ n ≧ 1)
n1	in	I	配列A,Ainvの配列宣言での行数 (n1 ≧ 1)
x	in	I	最初の行。A(1,*)はHI(x,*)をあらわす ただし、FR(*,*)は、次数nのフランク行列
nx	in	I	行列の行の数 (n ≧ x + nx - 1)
y	in	I	最初の列。A(*,1)はHI(*,y)をあらわす ただし、FR(*,*)は、次数nのフランク行列
ny	in	I	行列の列の数 (n ≧ y + ny - 1)
org	in	L	TRUE ： フランク行列中の指定された部分を返す FALSE ： フランク行列中の指定された部分を返さない
inv	in	L	FALSE ： 逆行列中の指定された部分を返さない
eig	in	L	TRUE ： 行列の固有値を返す FALSE ： 行列の固有値を返さない
A(n1,*)	out	D	フランク行列中の指定された部分
Ainv(n1,*)	out	D	使用しない
Eigen(*)	out	D	行列の固有値 (* ≧ n)
INFO	out	I	＝0：正常終了 ＝1：引数が不適当である。

10.5 ペイ行列の生成 JAERI_dgdzqPEss

10.5.1 機能

本サブルーチンはペイ行列とそのペイ行列の逆行列と固有値を返す。
ペイ行列とは行列成分が以下のように表される行列である。

$$A_{i,j} = \begin{cases} d & \text{(対角成分)} \\ 1 & \text{(非対角成分)} \end{cases}$$

10.5.2 コーリングシーケンス

```
      call JAERI_dgdzqPEss
     &     (n,n1,d,x,nx,y,ny,org,inv,eig,A,Ainv,Eigen,INFO)
```

10.5.3 引数

用法: in　入力用。値は壊されない。
　　out　出力用。
　　i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。
　　w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型 (integer)
　　D　倍精度実数型 (double precision)
　　Z　倍精度複素数型 (complex*16)
　　L　論理型 (logical)
　　C　文字型 (character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数（n1 ≧ n ≧ 1）
n1	in	I	配列A,Ainvの配列宣言での行数（n1 ≧ 1）
d	in	D	対角要素の値（d ＞ 1.0）
x	in	I	最初の行。A(1,*)はPI(x,*)をあらわす ただし、PI(*,*)は、次数nのペイ行列
nx	in	I	行列の行の数（n ≧ x + nx - 1）
y	in	I	最初の列。A(*,1)はPI(*,y)をあらわす ただし、PI(*,*)は、次数nのペイ行列
ny	in	I	行列の列の数（n ≧ y + ny - 1）
org	in	L	TRUE ：ペイ行列中の指定された部分を返す FALSE ：ペイ行列中の指定された部分を返さない
inv	in	L	TRUE ：逆行列中の指定された部分を返す FALSE ：逆行列中の指定された部分を返さない
eig	in	L	TRUE ：固有値を返す FALSE ：固有値を返さない
A(n1,*)	out	D	ペイ行列中の指定された部分
Ainv(n1,*)	out	D	逆行列中の指定された部分
Eigen(*)	out	D	行列の固有値（* ≧ n）
INFO	out	I	＝0：正常終了 ＝1：引数が不適当である。

10.6 レーマー行列の生成 JAERI_dgdzqLMss

10.6.1 機能

本サブルーチンはレーマー行列とそのレーマー行列の逆行列を返す。

レーマー行列とは行列成分が以下のように表される行列である。

$$A_{i,j} = \begin{cases} i \;/\; j & (i \leq j) \\ j \;/\; i & (i > j) \end{cases}$$

10.6.2 コーリングシーケンス

```
      call JAERI_dgdzqLMss
     &      (n,n1,  x,nx,y,ny,org,inv,eig,A,Ainv,Eigen,INFO)
```

10.6.3 引数

用法: in　入力用。値は壊されない。

out　出力用。

i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。

w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型 (integer)

D　倍精度実数型 (double precision)

Z　倍精度複素数型 (complex*16)

L　論理型 (logical)

C　文字型 (character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数 (n1 ≧ n ≧ 1)
n1	in	I	配列A,Ainvの配列宣言での行数 (n1 ≧ 1)
x	in	I	最初の行。A(1,*)はHI(x,*)をあらわす ただし、ML(*,*)は、次数nのレーマー行列
nx	in	I	行列の行の数 (n ≧ x + nx - 1)
y	in	I	最初の列。A(*,1)はHI(*,y)をあらわす ただし、LM(*,*)は、次数nのレーマー行列
ny	in	I	行列の列の数 (n ≧ y + ny - 1)
org	in	L	TRUE ：レーマー行列中の指定された部分を返す FALSE ：レーマー行列中の指定された部分を返さない
inv	in	L	TRUE ：逆行列中の指定された部分を返す FALSE ：逆行列中の指定された部分を返さない
eig	in	L	FALSE ：固有値を返さない
A(n1,*)	out	D	レーマー行列中の指定された部分
Ainv(n1,*)	out	D	逆行列中の指定された部分
Eigen(*)	out	D	使用しない (* ≧ n)
INFO	out	I	＝0：正常終了 ＝1：引数が不適当である。

10.7 特別な３重対角行列の生成　JAERI_dgdzqTRss

10.7.1 機能

本サブルーチンは特別な３重対角行列とその特別な３重対角行列の逆行列と固有値を返す。

特別な３重対角行列とは行列成分が以下のように表される行列である。

$$A_{i,j} = \begin{cases} -2 & (i = j) \\ 1 & (|i - j| = 1) \\ 0 & (|i - j| > 1) \end{cases}$$

10.7.2 コーリングシーケンス

```
      call JAERI_dgdzqTRss
     &      (n,n1,  x,nx,y,ny,org,inv,eig,A,Ainv,Eigen,    INFO)
```

10.7.3 引数

用法: in　入力用。値は壊されない。

out　出力用。

i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。

w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)

D　倍精度実数型(double precision)

Z　倍精度複素数型(complex*16)

L　論理型(logical)

C　文字型(character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数 (n1 ≧ n ≧ 1)
n1	in	I	配列A,Ainvの配列宣言での行数 (n1 ≧ 1)
x	in	I	最初の行。A(1,*)はTD(x,*)をあらわす ただし、TD(*,*)は、次数nの特別な３重対角行列
nx	in	I	行列の行の数 (n ≧ x + nx - 1)
y	in	I	最初の列。A(*,1)はTD(*,y)をあらわす ただし、TD(*,*)は、次数nの特別な３重対角行列
ny	in	I	行列の列の数 (n ≧ y + ny - 1)
org	in	L	TRUE ： 特別な３重対角行列中の指定された部分を返す FALSE ： 特別な３重対角行列中の指定された部分を返さない
inv	in	L	TRUE ： 逆行列中の指定された部分を返す FALSE ： 逆行列中の指定された部分を返さない
eig	in	L	TRUE ： 固有値を返す FALSE ： 固有値を返さない
A(n1,*)	out	D	特別な３重対角行列中の指定された部分
Ainv(n1,*)	out	D	行列の逆行列中の指定された部分
Eigen(*)	out	D	行列の固有値 (* ≧ n)
INFO	out	I	＝0：正常終了 ＝1：引数が不適当である。

11. 乱数行列生成ライブラリ

11.1 実数一般行列の生成 JAERI_dgdzrgdss

11.1.1 機能

実数一般正方行列となる乱数行列を生成する。

本サブルーチンは、主要入力データとして対角要素に関する指定、乱数の形式・範囲の指定などをとり、主要出力データとして乱数行列を返す。

11.1.2 コーリングシーケンス

```
    call JAERI_dgdzrgdss
   &     (n,n1,optdg,normal,u,v,method,seed,T,INFO)
```

11.1.3 引数

用法: in 入力用。値は壊されない。

out 出力用。

i/o 入出力用。入力として使用され、かつ戻り時に値が設定される。

w 作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I 整数型 (integer)

D 倍精度実数型 (double precision)

Z 倍精度複素数型 (complex*16)

L 論理型 (logical)

C 文字型 (character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数（n1 ≧ n ≧ 1）
n1	in	I	配列Tの配列宣言での行数（n1 ≧ 1）
optdg	in	I	対角要素に関する指定 ＝ 0 ：対角要素の符号は任意、対角優位に関する指定無し ＜ 0 ：対角要素の少なくとも1つは正の実数以外 ＞ 0 ：対角要素は全て正の実数 |optdg| = 1 ：行の対角優位を指定 = 2 ：列の対角優位を指定 = 3 ：行と列の対角優位を指定 = 10：行の対角優位を指定しない = 20：列の対角優位を指定しない = 30：行と列の対角優位を指定しない
normal	in	L	TRUE ：正規分布 FALSE ：一様分布
u	in	D	normal = TRUE ならば、非対角要素の平均 normal = FALSE ならば、非対角要素の最小値
v	in	D	normal = TRUE ならば、非対角要素の標準偏差（v > 0） normal = FALSE ならば、非対角要素の最大値（v > u）
method	in	I	1 ：乗算合同法によって乱数を生成する 2 ：混合合同法によって乱数を生成する
seed	i/o	I	入力：乱数の種 seed ≦ 0 ならば、デフォルト値を使用する method ＝ 1 かつ seed ＞ 0 ならば、偶数でなければならない 出力：次のコールのための乱数の種
T(n1,*)	out	D	生成された一般行列
INFO	out	I	＝ 0：正常終了 ＝ 1：引数が不適当である。

11.2 実数対称行列の生成 JAERI_dgdzrsdss

11.2.1 機能

実数対称行列となる乱数行列を生成する。

本サブルーチンは、主要入力データとして対角要素に関する指定、乱数の形式・範囲の指定などをとり、主要出力データとして乱数行列を返す。

11.2.2 コーリングシーケンス

```
      call JAERI_dgdzrsdss
     &     (n,n1,optdg,normal,u,v,method,seed,T,INFO)
```

11.2.3 引数

用法: in 入力用。値は壊されない。
out 出力用。
i/o 入出力用。入力として使用され、かつ戻り時に値が設定される。
w 作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I 整数型 (integer)
D 倍精度実数型 (double precision)
Z 倍精度複素数型 (complex*16)
L 論理型 (logical)
C 文字型 (character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数（n1 ≧ n ≧ 1)
n1	in	I	配列Tの配列宣言での行数（n1 ≧ 1)
optdg	in	I	対角要素に関する指定 ＝ 0 ： 対角要素の符号は任意、対角優位に関する指定無し ＜ 0 ： 対角要素の少なくとも1つは正の実数以外 ＞ 0 ： 対角要素は全て正の実数 |optdg| = 1, 2, 3 ： 対角優位を指定 = 10,20,30: 対角優位を指定しない
normal	in	L	TRUE ： 正規分布 FALSE ： 一様分布
u	in	D	normal = TRUE ならば、非対角要素の平均 normal = FALSE ならば、非対角要素の最小値
v	in	D	normal = TRUE ならば、非対角要素の標準偏差（v > 0) normal = FALSE ならば、非対角要素の最大値（v > u)
method	in	I	1 ： 乗算合同法によって乱数を生成する 2 ： 混合合同法によって乱数を生成する
seed	i/o	I	入力：乱数の種 seed ≦ 0 ならば、デフォルト値を使用する method = 1 かつ seed ＞ 0 ならば、偶数でなければならない 出力：次のコールのための乱数の種
T(n1,*)	out	D	生成された対称行列
INFO	out	I	＝0：正常終了 ＝1：引数が不適当である。

11.3 実数非対称帯行列の生成（圧縮）JAERI_dgbzrgbss

11.3.1 機能

実数非対称帯行列となる乱数行列を生成する。

本サブルーチンは、主要入力データとして対角要素に関する指定、乱数の形式・範囲の指定などをとり、主要出力データとして圧縮形式で格納された乱数行列を返す。

11.3.2 コーリングシーケンス

```
      call JAERI_dgbzrgbss
     &      (n,nb1,kl,ku,npadd,optdg,normal,u,v,method,seed,T,INFO)
```

11.3.3 引数

用法: in　入力用。値は壊されない。

out　出力用。

i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。

w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)

D　倍精度実数型(double precision)

Z　倍精度複素数型(complex*16)

L　論理型(logical)

C　文字型(character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数 (n ≧ 1)
nb1	in	I	配列Tの配列宣言での行数 (nb1 ≧ kl + ku + 1)
kl	in	I	下側バンド幅
ku	in	I	上側バンド幅
npadd	in	I	圧縮形式で余分に用意する領域の大きさ (npadd ≧ 0)
optdg	in	I	対角要素に関する指定 ＝ 0 ：対角要素の符号は任意、対角優位に関する指定無し ＜ 0 ：対角要素の少なくとも１つは正の実数以外 ＞ 0 ：対角要素は全て正の実数 |optdg| = 1 ：行の対角優位を指定 = 2 ：列の対角優位を指定 = 3 ：行と列の対角優位を指定 = 10：行の対角優位を指定しない = 20：列の対角優位を指定しない = 30：行と列の対角優位を指定しない
normal	in	L	TRUE ：正規分布 FALSE ：一様分布
u	in	D	normal = TRUE ならば、非対角要素の平均 normal = FALSE ならば、非対角要素の最小値
v	in	D	normal = TRUE ならば、非対角要素の標準偏差 (v > 0) normal = FALSE ならば、非対角要素の最大値 (v > u)
method	in	I	1 ：乗算合同法によって乱数を生成する 2 ：混合合同法によって乱数を生成する
seed	i/o	I	入力：乱数の種 seed ≦ 0 ならば、デフォルト値を使用する method ＝ 1 かつ seed ＞ 0 ならば、偶数でなければならない 出力：次のコールのための乱数の種
T(nb1,*)	out	D	生成された非対称帯行列
INFO	out	I	＝ 0：正常終了 ＝ 1：引数が不適当である。

11.4 実数非対称帯行列の生成（非圧縮）JAERI_dgdzrgbss

11.4.1 機能

実数非対称帯行列となる乱数行列を生成する。

本サブルーチンは、主要入力データとして対角要素に関する指定、乱数の形式・範囲の指定などをとり、主要出力データとして非圧縮形式で格納された乱数行列を返す。

11.4.2 コーリングシーケンス

```
      call JAERI_dgdzrgbss
     &     (n,n1, kl,ku,optdg,normal,u,v,method,seed,T,INFO)
```

11.4.3 引数

用法: in　入力用。値は壊されない。

out　出力用。

i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。

w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型 (integer)

D　倍精度実数型 (double precision)

Z　倍精度複素数型 (complex*16)

L　論理型 (logical)

C　文字型 (character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数（n1 ≧ n ≧ 1）
n1	in	I	配列Tの配列宣言での行数（n1 ≧ 1）
kl	in	I	下側バンド幅
ku	in	I	上側バンド幅
optdg	in	I	対角要素に関する指定 ＝ 0 ： 対角要素の符号は任意、対角優位に関する指定無し ＜ 0 ： 対角要素の少なくとも1つは正の実数以外 ＞ 0 ： 対角要素は全て正の実数 |optdg| = 1 ： 行の対角優位を指定 = 2 ： 列の対角優位を指定 = 3 ： 行と列の対角優位を指定 = 10： 行の対角優位を指定しない = 20： 列の対角優位を指定しない = 30： 行と列の対角優位を指定しない
normal	in	L	TRUE ： 正規分布 FALSE ： 一様分布
u	in	D	normal = TRUE ならば、非対角要素の平均 normal = FALSE ならば、非対角要素の最小値
v	in	D	normal = TRUE ならば、非対角要素の標準偏差（v > 0） normal = FALSE ならば、非対角要素の最大値（v > u）
method	in	I	1 ： 乗算合同法によって乱数を生成する 2 ： 混合合同法によって乱数を生成する
seed	i/o	I	入力：乱数の種 seed ≦ 0 ならば、デフォルト値を使用する method ＝ 1 かつ seed ＞ 0 ならば、偶数でなければならない 出力：次のコールのための乱数の種
T(n1,*)	out	D	生成された非対称帯行列
INFO	out	I	＝0：正常終了 ＝1：引数が不適当である。

11.5 実数対称帯行列の生成（圧縮）JAERI_dsbzrsbss

11.5.1 機能

実数対称帯行列となる乱数行列を生成する。

本サブルーチンは、主要入力データとして対角要素に関する指定、乱数の形式・範囲の指定などをとり、主要出力データとして圧縮形式で格納された乱数行列を返す。

11.5.2 コーリングシーケンス

```
call JAERI_dsbzrsbss
&     (n,nc1,k,npadd,optdg,normal,u,v,method,seed,T,INFO)
```

11.5.3 引数

用法: in　入力用。値は壊されない。
　out　出力用。
　i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。
　w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)
　D　倍精度実数型(double precision)
　Z　倍精度複素数型(complex*16)
　L　論理型(logical)
　C　文字型(character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数（n ≧ 1）
nc1	in	I	配列Tの配列宣言での行数（nc1 ≧ k + 1）
k	in	I	バンド幅
npadd	in	I	圧縮形式で余分に用意する領域の大きさ（npadd ≧ 0）
optdg	in	I	対角要素に関する指定 ＝ 0 ： 対角要素の符号は任意、対角優位に関する指定無し ＜ 0 ： 対角要素の少なくとも１つは正の実数以外 ＞ 0 ： 対角要素は全て正の実数 |optdg| = 1, 2, 3 ： 対角優位を指定 = 10,20,30: 対角優位を指定しない
normal	in	L	TRUE ： 正規分布 FALSE ： 一様分布
u	in	D	normal = TRUE ならば、非対角要素の平均 normal = FALSE ならば、非対角要素の最小値
v	in	D	normal = TRUE ならば、非対角要素の標準偏差（v > 0） normal = FALSE ならば、非対角要素の最大値（v > u）
method	in	I	1 ： 乗算合同法によって乱数を生成する 2 ： 混合合同法によって乱数を生成する
seed	i/o	I	入力：乱数の種 seed≦0 ならば、デフォルト値を使用する method＝1かつseed＞0 ならば、偶数でなければならない 出力：次のコールのための乱数の種
T(nc1,*)	out	D	生成された対称帯行列
INFO	out	I	＝0：正常終了 ＝1：引数が不適当である。

11.6 実数対称帯行列の生成（非圧縮）JAERI_dgdzrsbss

11.6.1 機能

実数対称帯行列となる乱数行列を生成する。

本サブルーチンは、主要入力データとして対角要素に関する指定、乱数の形式・範囲の指定などをとり、主要出力データとして非圧縮形式で格納された乱数行列を返す。

11.6.2 コーリングシーケンス

```
    call JAERI_dgdzrsbss
  &     (n,n1, k,optdg,normal,u,v,method,seed,T,INFO)
```

11.6.3 引数

用法: in　入力用。値は壊されない。

out　出力用。

i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。

w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)

D　倍精度実数型(double precision)

Z　倍精度複素数型(complex*16)

L　論理型(logical)

C　文字型(character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数（n1 ≧ n ≧ 1）
n1	in	I	配列Tの配列宣言での行数（n1 ≧ 1）
k	in	I	バンド幅
optdg	in	I	対角要素に関する指定 ＝ 0 ： 対角要素の符号は任意、対角優位に関する指定無し ＜ 0 ： 対角要素の少なくとも1つは正の実数以外 ＞ 0 ： 対角要素は全て正の実数 |optdg| = 1, 2, 3 ： 対角優位を指定 = 10,20,30: 対角優位を指定しない
normal	in	L	TRUE ： 正規分布 FALSE ： 一様分布
u	in	D	normal = TRUE ならば、非対角要素の平均 normal = FALSE ならば、非対角要素の最小値
v	in	D	normal = TRUE ならば、非対角要素の標準偏差（v > 0） normal = FALSE ならば、非対角要素の最大値（v > u）
method	in	I	1 ： 乗算合同法によって乱数を生成する 2 ： 混合合同法によって乱数を生成する
seed	i/o	I	入力：乱数の種 seed≦0 ならば、デフォルト値を使用する method＝1かつseed＞0 ならば、偶数でなければならない 出力：次のコールのための乱数の種
T(n1,*)	out	D	生成された対称帯行列
INFO	out	I	＝0：正常終了 ＝1：引数が不適当である。

11.7 実数n重対角非対称行列の生成（圧縮）JAERI_ds3zrs3ss

11.7.1 機能

実数n重対角非対称行列となる乱数行列を生成する。

本サブルーチンは、主要入力データとして対角要素に関する指定、乱数の形式・範囲の指定などをとり、主要出力データとして圧縮形式で格納された乱数行列を返す。

11.7.2 コーリングシーケンス

```
call JAERI_ds3zrs3ss
&      (n,m1,m2,npadd,optdg,normal,u,v,method,seed,
&                          a1,a2,a3,a4,a5,a6,a7, INFO)
```

11.7.3 引数

用法: in　入力用。値は壊されない。
out　出力用。
i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。
w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)
D　倍精度実数型(double precision)
Z　倍精度複素数型(complex*16)
L　論理型(logical)
C　文字型(character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数 (n ≧ 1)
m1	in	I	x軸方向の節点数 (m1 ≧ 2)
m2	in	I	y軸方向の節点数 (m2 ≧ 2)
npadd	in	I	圧縮形式で余分に用意する領域の大きさ (npadd ≧ 0)
optdg	in	I	対角要素に関する指定 ＝ 0 ： 対角要素の符号は任意、対角優位に関する指定無し ＜ 0 ： 対角要素の少なくとも1つは正の実数以外 ＞ 0 ： 対角要素は全て正の実数 |optdg| = 1 ： 行の対角優位を指定 = 2 ： 列の対角優位を指定 = 3 ： 行と列の対角優位を指定 = 10： 行の対角優位を指定しない = 20： 列の対角優位を指定しない = 30： 行と列の対角優位を指定しない
normal	in	L	TRUE ： 正規分布 FALSE ： 一様分布
u	in	D	normal = TRUE ならば、非対角要素の平均 normal = FALSE ならば、非対角要素の最小値
v	in	D	normal = TRUE ならば、非対角要素の標準偏差 (v > 0) normal = FALSE ならば、非対角要素の最大値 (v > u)
method	in	I	1 ： 乗算合同法によって乱数を生成する 2 ： 混合合同法によって乱数を生成する
seed	i/o	I	入力：乱数の種 seed ≦ 0 ならば、デフォルト値を使用する method＝1かつseed＞0 ならば、偶数でなければならない 出力：次のコールのための乱数の種
a1(*)	out	D	行列を構成する対角ベクトル (* ≧ n + 2 * npadd)
a2(*)	out	D	行列を構成する対角ベクトル (* ≧ n + 2 * npadd)
a3(*)	out	D	行列を構成する対角ベクトル (* ≧ n + 2 * npadd)
a4(*)	out	D	行列を構成する対角ベクトル (* ≧ n + 2 * npadd)
a5(*)	out	D	行列を構成する対角ベクトル (* ≧ n + 2 * npadd)
a6(*)	out	D	行列を構成する対角ベクトル (* ≧ n + 2 * npadd)
a7(*)	out	D	行列を構成する対角ベクトル (* ≧ n + 2 * npadd)
INFO	out	I	＝0：正常終了 ＝1：引数が不適当である。

11.8 実数n重対角非対称行列の生成（非圧縮）JAERI_dgdzrg3ss

11.8.1 機能

実数n重対角非対称行列となる乱数行列を生成する。

本サブルーチンは、主要入力データとして対角要素に関する指定、乱数の形式・範囲の指定などをとり、主要出力データとして非圧縮形式で格納された乱数行列を返す。

11.8.2 コーリングシーケンス

```
    call JAERI_dgdzrg3ss
  &      (n,n1, m1,m2,optdg,normal,u,v,method,seed,T,INFO)
```

11.8.3 引数

用法: in　入力用。値は壊されない。

out　出力用。

i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。

w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)

D　倍精度実数型(double precision)

Z　倍精度複素数型(complex*16)

L　論理型(logical)

C　文字型(character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数 (n1 ≧ n ≧ 1)
n1	in	I	配列Tの配列宣言での行数 (n1 ≧ 1)
ml	in	I	x軸方向の節点数 (m1 ≧ 2)
m2	in	I	y軸方向の節点数 (m2 ≧ 2)
optdg	in	I	対角要素に関する指定 ＝ 0 ： 対角要素の符号は任意、対角優位に関する指定無し ＜ 0 ： 対角要素の少なくとも1つは正の実数以外 ＞ 0 ： 対角要素は全て正の実数 |optdg| = 1 ： 行の対角優位を指定 = 2 ： 列の対角優位を指定 = 3 ： 行と列の対角優位を指定 = 10： 行の対角優位を指定しない = 20： 列の対角優位を指定しない = 30： 行と列の対角優位を指定しない
normal	in	L	TRUE ： 正規分布 FALSE ： 一様分布
u	in	D	normal = TRUE ならば、非対角要素の平均 normal = FALSE ならば、非対角要素の最小値
v	in	D	normal = TRUE ならば、非対角要素の標準偏差 (v > 0) normal = FALSE ならば、非対角要素の最大値 (v > u)
method	in	I	1 ： 乗算合同法によって乱数を生成する 2 ： 混合合同法によって乱数を生成する
seed	i/o	I	入力：乱数の種 seed≦0 ならば、デフォルト値を使用する method＝1かつseed＞0 ならば、偶数でなければならない 出力：次のコールのための乱数の種
T(n1,*)	out	D	生成されたn重対角非対称行列
INFO	out	I	＝0：正常終了 ＝1：引数が不適当である。

11.9 実数n重対角対称行列の生成（圧縮）JAERI_dg3zrg3ss

11.9.1 機能

実数n重対角対称行列となる乱数行列を生成する。

本サブルーチンは、主要入力データとして対角要素に関する指定、乱数の形式・範囲の指定などをとり、主要出力データとして圧縮形式で格納された乱数行列を返す。

11.9.2 コーリングシーケンス

```
     call JAERI_dg3zrg3ss
    &     (n,m1,m2,npadd,optdg,normal,u,v,method,seed,
    &                          a1,a2,a3,a4, INFO)
```

11.9.3 引数

用法: in　入力用。値は壊されない。

out　出力用。

i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。

w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)

D　倍精度実数型(double precision)

Z　倍精度複素数型(complex*16)

L　論理型(logical)

C　文字型(character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数 (n ≧ 1)
m1	in	I	x軸方向の節点数 (m1 ≧ 2)
m2	in	I	y軸方向の節点数 (m2 ≧ 2)
npadd	in	I	圧縮形式で余分に用意する領域の大きさ (npadd ≧ 0)
optdg	in	I	対角要素に関する指定 ＝ 0 ： 対角要素の符号は任意、対角優位に関する指定無し ＜ 0 ： 対角要素の少なくとも1つは正の実数以外 ＞ 0 ： 対角要素は全て正の実数 |optdg| = 1, 2, 3 ： 対角優位を指定 = 10,20,30: 対角優位を指定しない
normal	in	L	TRUE ： 正規分布 FALSE ： 一様分布
u	in	D	normal = TRUE ならば、非対角要素の平均 normal = FALSE ならば、非対角要素の最小値
v	in	D	normal = TRUE ならば、非対角要素の標準偏差 (v > 0) normal = FALSE ならば、非対角要素の最大値 (v > u)
method	in	I	1 ： 乗算合同法によって乱数を生成する 2 ： 混合合同法によって乱数を生成する
seed	i/o	I	入力：乱数の種 seed ≦ 0 ならば、デフォルト値を使用する method ＝ 1 かつ seed ＞ 0 ならば、偶数でなければならない 出力：次のコールのための乱数の種
a1(*)	out	D	行列を構成する対角ベクトル (* ≧ n + 2 * npadd)
a2(*)	out	D	行列を構成する対角ベクトル (* ≧ n + 2 * npadd)
a3(*)	out	D	行列を構成する対角ベクトル (* ≧ n + 2 * npadd)
a4(*)	out	D	行列を構成する対角ベクトル (* ≧ n + 2 * npadd)
INFO	out	I	＝ 0：正常終了 ＝ 1：引数が不適当である。

11.10 実数n重対角対称行列の生成（非圧縮）JAERI_dgdzrs3ss

11.10.1 機能

実数n重対角対称行列となる乱数行列を生成する。

本サブルーチンは、主要入力データとして対角要素に関する指定、乱数の形式・範囲の指定などをとり、主要出力データとして非圧縮形式で格納された乱数行列を返す。

11.10.2 コーリングシーケンス

```
    call JAERI_dgdzrs3ss
  &      (n,n1, m1,m2,optdg,normal,u,v,method,seed,T,INFO)
```

11.10.3 引数

用法: in　入力用。値は壊されない。

out　出力用。

i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。

w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)

D　倍精度実数型(double precision)

Z　倍精度複素数型(complex*16)

L　論理型(logical)

C　文字型(character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数 (n1 ≧ n ≧ 1)
n1	in	I	配列Tの配列宣言での行数 (n1 ≧ 1)
m1	in	I	x軸方向の節点数 (m1 ≧ 2)
m2	in	I	y軸方向の節点数 (m2 ≧ 2)
optdg	in	I	対角要素に関する指定 ＝ 0 ： 対角要素の符号は任意、対角優位に関する指定無し ＜ 0 ： 対角要素の少なくとも1つは正の実数以外 ＞ 0 ： 対角要素は全て正の実数 |optdg| = 1, 2, 3 ： 対角優位を指定 = 10,20,30: 対角優位を指定しない
normal	in	L	TRUE ： 正規分布 FALSE ： 一様分布
u	in	D	normal = TRUE ならば、非対角要素の平均 normal = FALSE ならば、非対角要素の最小値
v	in	D	normal = TRUE ならば、非対角要素の標準偏差 (v ＞ 0) normal = FALSE ならば、非対角要素の最大値 (v ＞ u)
method	in	I	1 ： 乗算合同法によって乱数を生成する 2 ： 混合合同法によって乱数を生成する
seed	i/o	I	入力：乱数の種 seed ≦ 0 ならば、デフォルト値を使用する method ＝ 1 かつ seed ＞ 0 ならば、偶数でなければならない 出力：次のコールのための乱数の種
T(n1,*)	out	D	生成されたn重対角対称行列
INFO	out	I	＝0：正常終了 ＝1：引数が不適当である。

11.11 実数三角行列の生成（圧縮）JAERI_dgtzrgtss

11.11.1 機能

実数三角行列となる乱数行列を生成する。

本サブルーチンは、主要入力データとして対角要素に関する指定、乱数の形式・範囲の指定などをとり、主要出力データとして圧縮形式で格納された乱数行列を返す。

11.11.2 コーリングシーケンス

```
call JAERI_dgtzrgtss
&      (n,upper,optdg,normal,u,v,method,seed,T,INFO)
```

11.11.3 引数

用法: in　入力用。値は壊されない。
　out　出力用。
　i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。
　w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)
　D　倍精度実数型(double precision)
　Z　倍精度複素数型(complex*16)
　L　論理型(logical)
　C　文字型(character * (*))

変数／配列(次元)	I/O	型	意味
n	I	行列の次数	(n ≧ 1)
upper	in	L	TRUE　: 上三角部分を出力する FALSE : 下三角部分を出力する
optdg	in	I	対角要素に関する指定 ＝ 0 : 対角要素の符号は任意、対角優位に関する指定無し ＜ 0 : 対角要素の少なくとも1つは正の実数以外 ＞ 0 : 対角要素は全て正の実数 |optdg| = 1 : 行の対角優位を指定 = 2 : 列の対角優位を指定 = 3 : 行と列の対角優位を指定 = 10: 行の対角優位を指定しない = 20: 列の対角優位を指定しない = 30: 行と列の対角優位を指定しない
normal	in	L	TRUE　: 正規分布 FALSE : 一様分布
u	in	D	normal = TRUE　ならば、非対角要素の平均 normal = FALSE ならば、非対角要素の最小値
v	in	D	normal = TRUE　ならば、非対角要素の標準偏差 (v > 0) normal = FALSE ならば、非対角要素の最大値 (v > u)
method	in	I	1 : 乗算合同法によって乱数を生成する 2 : 混合合同法によって乱数を生成する
seed	i/o	I	入力：乱数の種 seed≦0 ならば、デフォルト値を使用する method＝1かつseed＞0 ならば、偶数でなければならない 出力：次のコールのための乱数の種
T(*)	out	D	生成された三角行列 (* ≧ n × (n + 1) / 2)
INFO	out	I	＝0：正常終了 ＝1：引数が不適当である。

11.12 実数三角行列の生成（非圧縮）JAERI_dgdzrgtss

11.12.1 機能

実数三角行列となる乱数行列を生成する。

本サブルーチンは、主要入力データとして対角要素に関する指定、乱数の形式・範囲の指定などをとり、主要出力データとして非圧縮形式で格納された乱数行列を返す。

11.12.2 コーリングシーケンス

```
    call JAERI_dgdzrgtss
   &     (n,n1, upper,       optdg,normal,u,v,method,seed,T,INFO)
```

11.12.3 引数

用法: in　入力用。値は壊されない。

out　出力用。

i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。

w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)

D　倍精度実数型(double precision)

Z　倍精度複素数型(complex*16)

L　論理型(logical)

C　文字型(character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数 (n1 ≧ n ≧ 1)
n1	in	I	配列Tの配列宣言での行数 (n1 ≧ 1)
upper	in	L	TRUE ： 上三角部分を出力する FALSE ： 下三角部分を出力する
optdg	in	I	対角要素に関する指定 ＝ 0 ： 対角要素の符号は任意、対角優位に関する指定無し ＜ 0 ： 対角要素の少なくとも1つは正の実数以外 ＞ 0 ： 対角要素は全て正の実数 |optdg| = 1 ： 行の対角優位を指定 = 2 ： 列の対角優位を指定 = 3 ： 行と列の対角優位を指定 = 10： 行の対角優位を指定しない = 20： 列の対角優位を指定しない = 30： 行と列の対角優位を指定しない
normal	in	L	TRUE ： 正規分布 FALSE ： 一様分布
u	in	D	normal = TRUE ならば、非対角要素の平均 normal = FALSE ならば、非対角要素の最小値
v	in	D	normal = TRUE ならば、非対角要素の標準偏差 (v > 0) normal = FALSE ならば、非対角要素の最大値 (v > u)
method	in	I	1 ： 乗算合同法によって乱数を生成する 2 ： 混合合同法によって乱数を生成する
seed	i/o	I	入力：乱数の種 seed≦0 ならば、デフォルト値を使用する method＝1かつseed＞0 ならば、偶数でなければならない 出力：次のコールのための乱数の種
T(n1,*)	out	D	生成された三角行列
INFO	out	I	＝0： 正常終了 ＝1： 引数が不適当である。

11.13 複素数一般行列の生成 JAERI_zgdzrgdss

11.13.1 機能

複素数一般正方行列となる複素数を要素とする乱数行列を生成する。

本サブルーチンは、主要入力データとして対角要素に関する指定、乱数の形式・範囲の指定などをとり、主要出力データとして複素行列を返す。

11.13.2 コーリングシーケンス

```
call JAERI_zgdzrgdss
&      (n,n1,optdg,normal,u,v,method,seed,T,INFO)
```

11.13.3 引数

用法:	in	入力用。値は壊されない。
	out	出力用。
	i/o	入出力用。入力として使用され、かつ戻り時に値が設定される。
	w	作業用。作業領域として与えられる。戻り時の値は意味を持たない。
型 :	I	整数型 (integer)
	D	倍精度実数型 (double precision)
	Z	倍精度複素数型 (complex*16)
	L	論理型 (logical)
	C	文字型 (character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数（n1 ≧ n ≧ 1）
n1	in	I	配列Tの配列宣言での行数（n1 ≧ 1）
optdg	in	I	対角要素に関する指定 ＝ 0 ：対角要素の符号は任意、対角優位に関する指定無し ＜ 0 ：対角要素の少なくとも１つは正の実数以外 ＞ 0 ：対角要素は全て正の実数 |optdg| = 1 ：行の対角優位を指定 = 2 ：列の対角優位を指定 = 3 ：行と列の対角優位を指定 = 10：行の対角優位を指定しない = 20：列の対角優位を指定しない = 30：行と列の対角優位を指定しない
normal	in	L	TRUE ：正規分布 FALSE ：一様分布
u	in	D	normal = TRUE ならば、非対角要素の平均 normal = FALSE ならば、非対角要素の最小値
v	in	D	normal = TRUE ならば、非対角要素の標準偏差（v > 0） normal = FALSE ならば、非対角要素の最大値（v > u）
method	in	I	1 ：乗算合同法によって乱数を生成する 2 ：混合合同法によって乱数を生成する
seed	i/o	I	入力：乱数の種 seed≦0 ならば、デフォルト値を使用する method＝1かつseed＞0 ならば、偶数でなければならない 出力：次のコールのための乱数の種
T(n1,*)	out	Z	生成された一般複素行列
INFO	out	I	＝0：正常終了 ＝1：引数が不適当である。

11.14 エルミート行列の生成 JAERI_zgdzrsdss

11.14.1 機能

エルミート行列となる複素数を要素とする乱数行列を生成する。

本サブルーチンは、主要入力データとして対角要素に関する指定、乱数の形式・範囲の指定などをとり、主要出力データとしてエルミート行列を返す。

11.14.2 コーリングシーケンス

```
    call JAERI_zgdzrsdss
   &     (n,n1,optdg,normal,u,v,method,seed,T,INFO)
```

11.14.3 引数

用法: in 入力用。値は壊されない。
out 出力用。
i/o 入出力用。入力として使用され、かつ戻り時に値が設定される。
w 作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I 整数型(integer)
D 倍精度実数型(double precision)
Z 倍精度複素数型(complex*16)
L 論理型(logical)
C 文字型(character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数（n1 ≧ n ≧ 1）
n1	in	I	配列Tの配列宣言での行数（n1 ≧ 1）
optdg	in	I	対角要素に関する指定 ＝ 0 ： 対角要素の符号は任意、対角優位に関する指定無し ＜ 0 ： 対角要素の少なくとも１つは正の実数以外 ＞ 0 ： 対角要素は全て正の実数 |optdg| = 1, 2, 3 ： 対角優位を指定 = 10,20,30: 対角優位を指定しない
normal	in	L	TRUE ： 正規分布 FALSE ： 一様分布
u	in	D	normal = TRUE ならば、非対角要素の平均 normal = FALSE ならば、非対角要素の最小値
v	in	D	normal = TRUE ならば、非対角要素の標準偏差（v ＞ 0） normal = FALSE ならば、非対角要素の最大値（v ＞ u）
method	in	I	１ ： 乗算合同法によって乱数を生成する ２ ： 混合合同法によって乱数を生成する
seed	i/o	I	入力：乱数の種 seed ≦ 0 ならば、デフォルト値を使用する method ＝ 1 かつ seed ＞ 0 ならば、偶数でなければならない 出力：次のコールのための乱数の種
T(n1,*)	out	Z	生成されたエルミート行列
INFO	out	I	＝0：正常終了 ＝1：引数が不適当である。

11.15 非エルミート帯行列の生成（圧縮）JAERI_zgbzrgbss

11.15.1 機能

非エルミート帯行列となる複素数を要素とする乱数行列を生成する。

本サブルーチンは、主要入力データとして対角要素に関する指定、乱数の形式・範囲の指定などをとり、主要出力データとして圧縮形式で格納された非エルミート帯行列を返す。

11.15.2 コーリングシーケンス

```
      call JAERI_zgbzrgbss
     &     (n,nb1,kl,ku,npadd,optdg,normal,u,v,method,seed,T,INFO)
```

11.15.3 引数

用法: in　入力用。値は壊されない。

out　出力用。

i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。

w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型 (integer)

D　倍精度実数型 (double precision)

Z　倍精度複素数型 (complex*16)

L　論理型 (logical)

C　文字型 (character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数（n ≧ 1）
nb1	in	I	配列Tの配列宣言での行数（nb1 ≧ kl + ku + 1）
kl	in	I	下側バンド幅
ku	in	I	上側バンド幅
npadd	in	I	圧縮形式で余分に用意する領域の大きさ（npadd ≧ 0）
optdg	in	I	対角要素に関する指定 ＝ 0 ：対角要素の符号は任意、対角優位に関する指定無し ＜ 0 ：対角要素の少なくとも1つは正の実数以外 ＞ 0 ：対角要素は全て正の実数 |optdg| = 1 ：行の対角優位を指定 = 2 ：列の対角優位を指定 = 3 ：行と列の対角優位を指定 = 10：行の対角優位を指定しない = 20：列の対角優位を指定しない = 30：行と列の対角優位を指定しない
normal	in	L	TRUE ：正規分布 FALSE ：一様分布
u	in	D	normal = TRUE ならば、非対角要素の平均 normal = FALSE ならば、非対角要素の最小値
v	in	D	normal = TRUE ならば、非対角要素の標準偏差（v > 0） normal = FALSE ならば、非対角要素の最大値（v > u）
method	in	I	1 ：乗算合同法によって乱数を生成する 2 ：混合合同法によって乱数を生成する
seed	i/o	I	入力：乱数の種 seed≦0 ならば、デフォルト値を使用する method＝1かつseed＞0 ならば、偶数でなければならない 出力：次のコールのための乱数の種
T(nb1,*)	out	Z	生成された非エルミート帯行列
INFO	out	I	＝0：正常終了 ＝1：引数が不適当である。

11.16 非エルミート帯行列の生成（非圧縮）JAERI_zgdzrgbss

11.16.1 機能

非エルミート帯行列となる複素数を要素とする乱数行列を生成する。

本サブルーチンは、主要入力データとして対角要素に関する指定、乱数の形式・範囲の指定などをとり、主要出力データとして非圧縮形式で格納された非エルミート帯行列を返す。

11.16.2 コーリングシーケンス

```
      call JAERI_zgdzrgbss
     &     (n,n1, kl,ku,optdg,normal,u,v,method,seed,T,INFO)
```

11.16.3 引数

用法: in　入力用。値は壊されない。

out　出力用。

i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。

w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)

D　倍精度実数型(double precision)

Z　倍精度複素数型(complex*16)

L　論理型(logical)

C　文字型(character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数（n1 ≧ n ≧ 1）
n1	in	I	配列Tの配列宣言での行数（n1 ≧ 1）
kl	in	I	下側バンド幅
ku	in	I	上側バンド幅
optdg	in	I	対角要素に関する指定 ＝ 0 ： 対角要素の符号は任意、対角優位に関する指定無し ＜ 0 ： 対角要素の少なくとも1つは正の実数以外 ＞ 0 ： 対角要素は全て正の実数 |optdg| = 1 ： 行の対角優位を指定 = 2 ： 列の対角優位を指定 = 3 ： 行と列の対角優位を指定 = 10： 行の対角優位を指定しない = 20： 列の対角優位を指定しない = 30： 行と列の対角優位を指定しない
normal	in	L	TRUE ： 正規分布 FALSE ： 一様分布
u	in	D	normal = TRUE ならば、非対角要素の平均 normal = FALSE ならば、非対角要素の最小値
v	in	D	normal = TRUE ならば、非対角要素の標準偏差（v > 0） normal = FALSE ならば、非対角要素の最大値（v > u）
method	in	I	1 ： 乗算合同法によって乱数を生成する 2 ： 混合合同法によって乱数を生成する
seed	i/o	I	入力：乱数の種 seed ≦ 0 ならば、デフォルト値を使用する method＝1かつseed＞0 ならば、偶数でなければならない 出力：次のコールのための乱数の種
T(n1,*)	out	Z	生成された非エルミート帯行列
INFO	out	I	＝0：正常終了 ＝1：引数が不適当である。

11.17 エルミート帯行列の生成（圧縮）JAERI_zhbzrhbss

11.17.1 機能

エルミート帯行列となる複素数を要素とする乱数行列を生成する。

本サブルーチンは、主要入力データとして対角要素に関する指定、乱数の形式・範囲の指定などをとり、主要出力データとして圧縮形式で格納されたエルミート帯行列を返す。

11.17.2 コーリングシーケンス

```
    call JAERI_zhbzrhbss
   &     (n,nc1,k,     npadd,optdg,normal,u,v,method,seed,T,INFO)
```

11.17.3 引数

用法: in　入力用。値は壊されない。

out　出力用。

i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。

w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型 (integer)

D　倍精度実数型 (double precision)

Z　倍精度複素数型 (complex*16)

L　論理型 (logical)

C　文字型 (character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数 (n ≧ 1)
nc1	in	I	配列Tの配列宣言での行数 (nc1 ≧ k + 1)
k	in	I	バンド幅
npadd	in	I	圧縮形式で余分に用意する領域の大きさ (npadd ≧ 0)
optdg	in	I	対角要素に関する指定 ＝ 0 ： 対角要素の符号は任意、対角優位に関する指定無し ＜ 0 ： 対角要素の少なくとも1つは正の実数以外 ＞ 0 ： 対角要素は全て正の実数 |optdg| = 1, 2, 3 ： 対角優位を指定 = 10,20,30: 対角優位を指定しない
normal	in	L	TRUE ： 正規分布 FALSE ： 一様分布
u	in	D	normal = TRUE ならば、非対角要素の平均 normal = FALSE ならば、非対角要素の最小値
v	in	D	normal = TRUE ならば、非対角要素の標準偏差 (v > 0) normal = FALSE ならば、非対角要素の最大値 (v > u)
method	in	I	1 ： 乗算合同法によって乱数を生成する 2 ： 混合合同法によって乱数を生成する
seed	i/o	I	入力：乱数の種 seed≦0 ならば、デフォルト値を使用する method＝1かつseed＞0 ならば、偶数でなければならない 出力：次のコールのための乱数の種
T(nc1,*)	out	Z	生成されたエルミート帯行列
INFO	out	I	＝0：正常終了 ＝1：引数が不適当である。

11.18 エルミート帯行列の生成（非圧縮）JAERI_zgdzrhbss

11.18.1 機能

エルミート帯行列となる乱数行列を生成する。

本サブルーチンは、主要入力データとして対角要素に関する指定、乱数の形式・範囲の指定などをとり、主要出力データとして非圧縮形式で格納されたエルミート帯行列を返す。

11.18.2 コーリングシーケンス

```
    call JAERI_zgdzrhbss
   &     (n,n1, k,optdg,normal,u,v,method,seed,T,INFO)
```

11.18.3 引数

用法: in　入力用。値は壊されない。

out　出力用。

i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。

w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)

D　倍精度実数型(double precision)

Z　倍精度複素数型(complex*16)

L　論理型(logical)

C　文字型(character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数 (n1 ≧ n ≧ 1)
n1	in	I	配列Tの配列宣言での行数 (n1 ≧ 1)
k	in	I	バンド幅
optdg	in	I	対角要素に関する指定 ＝ 0 ： 対角要素の符号は任意、対角優位に関する指定無し ＜ 0 ： 対角要素の少なくとも1つは正の実数以外 ＞ 0 ： 対角要素は全て正の実数 |optdg| = 1, 2, 3 ： 対角優位を指定 = 10,20,30: 対角優位を指定しない
normal	in	L	TRUE ： 正規分布 FALSE ： 一様分布
u	in	D	normal = TRUE ならば、非対角要素の平均 normal = FALSE ならば、非対角要素の最小値
v	in	D	normal = TRUE ならば、非対角要素の標準偏差 (v > 0) normal = FALSE ならば、非対角要素の最大値 (v > u)
method	in	I	1 ： 乗算合同法によって乱数を生成する 2 ： 混合合同法によって乱数を生成する
seed	i/o	I	入力：乱数の種 seed ≦ 0 ならば、デフォルト値を使用する method＝1かつseed＞0 ならば、偶数でなければならない 出力：次のコールのための乱数の種
T(n1,*)	out	Z	生成されたエルミート帯行列
INFO	out	I	＝0：正常終了 ＝1：引数が不適当である。

11.19 n重対角非エルミート行列の生成（圧縮）JAERI_zh3zrh3ss

11.19.1 機能

n重対角非エルミート行列となる複素数を要素とする乱数行列を生成する。

本サブルーチンは、主要入力データとして対角要素に関する指定、乱数の形式・範囲の指定などをとり、主要出力データとして圧縮形式で格納されたn重対角非エルミート行列を返す。

11.19.2 コーリングシーケンス

```
   call JAERI_zh3zrh3ss
  &     (n,m1,m2,npadd,optdg,normal,u,v,method,seed,
  &                   a1,a2,a3,a4,a5,a6,a7, INFO)
```

11.19.3 引数

用法: in　入力用。値は壊されない。

out　出力用。

i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。

w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)

D　倍精度実数型(double precision)

Z　倍精度複素数型(complex*16)

L　論理型(logical)

C　文字型(character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数（n ≧ 1）
m1	in	I	x軸方向の節点数 (m1 ≧ 2)
m2	in	I	y軸方向の節点数 (m2 ≧ 2)
npadd	in	I	圧縮形式で余分に用意する領域の大きさ（npadd ≧ 0）
optdg	in	I	対角要素に関する指定 ＝ 0 ： 対角要素の符号は任意、対角優位に関する指定無し ＜ 0 ： 対角要素の少なくとも1つは正の実数以外 ＞ 0 ： 対角要素は全て正の実数 |optdg| = 1 ： 行の対角優位を指定 = 2 ： 列の対角優位を指定 = 3 ： 行と列の対角優位を指定 = 10： 行の対角優位を指定しない = 20： 列の対角優位を指定しない = 30： 行と列の対角優位を指定しない
normal	in	L	TRUE ： 正規分布 FALSE ： 一様分布
u	in	D	normal = TRUE ならば、非対角要素の平均 normal = FALSE ならば、非対角要素の最小値
v	in	D	normal = TRUE ならば、非対角要素の標準偏差（v > 0） normal = FALSE ならば、非対角要素の最大値（v > u）
method	in	I	1 ： 乗算合同法によって乱数を生成する 2 ： 混合合同法によって乱数を生成する
seed	i/o	I	入力：乱数の種 seed ≦ 0 ならば、デフォルト値を使用する method＝1かつseed＞0 ならば、偶数でなければならない 出力：次のコールのための乱数の種
a1(*)	out	Z	行列を構成する対角ベクトル（* ≧ n + 2 * npadd）
a2(*)	out	Z	行列を構成する対角ベクトル（* ≧ n + 2 * npadd）
a3(*)	out	Z	行列を構成する対角ベクトル（* ≧ n + 2 * npadd）
a4(*)	out	Z	行列を構成する対角ベクトル（* ≧ n + 2 * npadd）
a5(*)	out	Z	行列を構成する対角ベクトル（* ≧ n + 2 * npadd）
a6(*)	out	Z	行列を構成する対角ベクトル（* ≧ n + 2 * npadd）
a7(*)	out	Z	行列を構成する対角ベクトル（* ≧ n + 2 * npadd）
INFO	out	I	＝0：正常終了 ＝1：引数が不適当である。

11.20 n重対角非エルミート行列の生成（非圧縮）JAERI_zgdzrg3ss

11.20.1 機能

n重対角非エルミート行列となる複素数を要素とする乱数行列を生成する。

本サブルーチンは、主要入力データとして対角要素に関する指定、乱数の形式・範囲の指定などをとり、主要出力データとして非圧縮形式で格納されたn重対角非エルミート行列を返す。

11.20.2 コーリングシーケンス

```
    call JAERI_zgdzrg3ss
  &      (n,n1, m1,m2,optdg,normal,u,v,method,seed,T,INFO)
```

11.20.3 引数

用法: in　入力用。値は壊されない。

out　出力用。

i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。

w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)

D　倍精度実数型(double precision)

Z　倍精度複素数型(complex*16)

L　論理型(logical)

C　文字型(character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数（n1 ≧ n ≧ 1）
n1	in	I	配列Tの配列宣言での行数（n1 ≧ 1）
m1	in	I	x軸方向の節点数 (m1 ≧ 2)
m2	in	I	y軸方向の節点数 (m2 ≧ 2)
optdg	in	I	対角要素に関する指定 ＝ 0 ： 対角要素の符号は任意、対角優位に関する指定無し ＜ 0 ： 対角要素の少なくとも1つは正の実数以外 ＞ 0 ： 対角要素は全て正の実数 |optdg| = 1 ： 行の対角優位を指定 = 2 ： 列の対角優位を指定 = 3 ： 行と列の対角優位を指定 = 10： 行の対角優位を指定しない = 20： 列の対角優位を指定しない = 30： 行と列の対角優位を指定しない
normal	in	L	TRUE　： 正規分布 FALSE ： 一様分布
u	in	D	normal = TRUE　ならば、非対角要素の平均 normal = FALSE ならば、非対角要素の最小値
v	in	D	normal = TRUE　ならば、非対角要素の標準偏差（v > 0） normal = FALSE ならば、非対角要素の最大値（v > u）
method	in	I	1 ： 乗算合同法によって乱数を生成する 2 ： 混合合同法によって乱数を生成する
seed	i/o	I	入力：乱数の種 seed ≦ 0　ならば、デフォルト値を使用する method ＝ 1 かつ seed ＞ 0　ならば、偶数でなければならない 出力：次のコールのための乱数の種
T(n1,*)	out	Z	生成されたn重対角非エルミート行列
INFO	out	I	＝0：正常終了 ＝1：引数が不適当である。

11.21 n重対角エルミート行列の生成（圧縮）JAERI_zg3zrg3ss

11.21.1 機能

n重対角エルミート行列となる複素数を要素とする乱数行列を生成する。

本サブルーチンは、主要入力データとして対角要素に関する指定、乱数の形式・範囲の指定などをとり、主要出力データとして圧縮形式で格納されたn重対角エルミート行列を返す。

11.21.2 コーリングシーケンス

```
      call JAERI_zg3zrg3ss
     &     (n,m1,m2,npadd,optdg,normal,u,v,method,seed,
     &                      a1,a2,a3,a4, INFO)
```

11.21.3 引数

用法: in 入力用。値は壊されない。

out 出力用。

i/o 入出力用。入力として使用され、かつ戻り時に値が設定される。

w 作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I 整数型(integer)

D 倍精度実数型(double precision)

Z 倍精度複素数型(complex*16)

L 論理型(logical)

C 文字型(character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数 (n ≧ 1)
m1	in	I	x軸方向の節点数 (m1 ≧ 2)
m2	in	I	y軸方向の節点数 (m2 ≧ 2)
npadd	in	I	圧縮形式で余分に用意する領域の大きさ (npadd ≧ 0)
optdg	in	I	対角要素に関する指定 ＝ 0 ： 対角要素の符号は任意、対角優位に関する指定無し ＜ 0 ： 対角要素の少なくとも１つは正の実数以外 ＞ 0 ： 対角要素は全て正の実数 |optdg| = 1, 2, 3 ： 対角優位を指定 = 10,20,30: 対角優位を指定しない
normal	in	L	TRUE ： 正規分布 FALSE ： 一様分布
u	in	D	normal = TRUE ならば、非対角要素の平均 normal = FALSE ならば、非対角要素の最小値
v	in	D	normal = TRUE ならば、非対角要素の標準偏差 (v > 0) normal = FALSE ならば、非対角要素の最大値 (v > u)
method	in	I	1 ： 乗算合同法によって乱数を生成する 2 ： 混合合同法によって乱数を生成する
seed	i/o	I	入力：乱数の種 seed ≦ 0 ならば、デフォルト値を使用する method ＝ 1 かつ seed ＞ 0 ならば、偶数でなければならない 出力：次のコールのための乱数の種
a1(*)	out	Z	行列を構成する対角ベクトル (* ≧ n + 2 * npadd)
a2(*)	out	Z	行列を構成する対角ベクトル (* ≧ n + 2 * npadd)
a3(*)	out	Z	行列を構成する対角ベクトル (* ≧ n + 2 * npadd)
a4(*)	out	Z	行列を構成する対角ベクトル (* ≧ n + 2 * npadd)
INFO	out	I	＝0：正常終了 ＝1：引数が不適当である。

11.22 n重対角エルミート行列の生成（非圧縮）JAERI_zgdzrh3ss

11.22.1 機能

n重対角エルミート行列となる複素数を要素とする乱数行列を生成する。

本サブルーチンは、主要入力データとして対角要素に関する指定、乱数の形式・範囲の指定などをとり、主要出力データとして非圧縮形式で格納されたn重対角エルミート行列を返す。

11.22.2 コーリングシーケンス

```
      call JAERI_zgdzrh3ss
     &     (n,n1, m1,m2,optdg,normal,u,v,method,seed,T,INFO)
```

11.22.3 引数

用法: in　入力用。値は壊されない。

out　出力用。

i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。

w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)

D　倍精度実数型(double precision)

Z　倍精度複素数型(complex*16)

L　論理型(logical)

C　文字型(character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数（n1 ≧ n ≧ 1）
n1	in	I	配列Tの配列宣言での行数（n1 ≧ 1）
m1	in	I	x軸方向の節点数(m1 ≧ 2)
m2	in	I	y軸方向の節点数(m2 ≧ 2)
optdg	in	I	対角要素に関する指定 ＝ 0 ： 対角要素の符号は任意、対角優位に関する指定無し ＜ 0 ： 対角要素の少なくとも1つは正の実数以外 ＞ 0 ： 対角要素は全て正の実数 |optdg| = 1, 2, 3 ： 対角優位を指定 = 10,20,30: 対角優位を指定しない
normal	in	L	TRUE ： 正規分布 FALSE ： 一様分布
u	in	D	normal = TRUE ならば、非対角要素の平均 normal = FALSE ならば、非対角要素の最小値
v	in	D	normal = TRUE ならば、非対角要素の標準偏差（v > 0） normal = FALSE ならば、非対角要素の最大値（v > u）
method	in	I	1 ： 乗算合同法によって乱数を生成する 2 ： 混合合同法によって乱数を生成する
seed	i/o	I	入力：乱数の種 seed≦0 ならば、デフォルト値を使用する method＝1かつseed＞0 ならば、偶数でなければならない 出力：次のコールのための乱数の種
T(n1,*)	out	Z	生成されたn重対角エルミート行列
INFO	out	I	＝0：正常終了 ＝1：引数が不適当である。

11.23 複素数三角行列の生成（圧縮）JAERI_zgtzrgtss

11.23.1 機能

三角行列となる複素数を要素とする乱数行列を生成する。

本サブルーチンは、主要入力データとして対角要素に関する指定、乱数の形式・範囲の指定などをとり、主要出力データとして圧縮形式で格納された複素三角行列を返す。

11.23.2 コーリングシーケンス

```
      call JAERI_zgtzrgtss
     &      (n,upper,optdg,normal,u,v,method,seed,T,INFO)
```

11.23.3 引数

用法: in　入力用。値は壊されない。
out　出力用。
i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。
w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)
D　倍精度実数型(double precision)
Z　倍精度複素数型(complex*16)
L　論理型(logical)
C　文字型(character *(*))

変数／配列(次元)	I/O	型	意味
n	I	行列の次数	(n ≧ 1)
upper	in	L	TRUE ：上三角部分を出力する FALSE ：下三角部分を出力する
optdg	in	I	対角要素に関する指定 ＝ 0 ：対角要素の符号は任意、対角優位に関する指定無し ＜ 0 ：対角要素の少なくとも１つは正の実数以外 ＞ 0 ：対角要素は全て正の実数 |optdg| = 1 ：行の対角優位を指定 = 2 ：列の対角優位を指定 = 3 ：行と列の対角優位を指定 = 10：行の対角優位を指定しない = 20：列の対角優位を指定しない = 30：行と列の対角優位を指定しない
normal	in	L	TRUE ：正規分布 FALSE ：一様分布
u	in	D	normal = TRUE ならば、非対角要素の平均 normal = FALSE ならば、非対角要素の最小値
v	in	D	normal = TRUE ならば、非対角要素の標準偏差 (v > 0) normal = FALSE ならば、非対角要素の最大値 (v > u)
method	in	I	1 ：乗算合同法によって乱数を生成する 2 ：混合合同法によって乱数を生成する
seed	i/o	I	入力：乱数の種 seed ≦ 0 ならば、デフォルト値を使用する method＝1かつseed＞0 ならば、偶数でなければならない 出力：次のコールのための乱数の種
T(*)	out	Z	生成された三角行列 (* ≧ n × (n + 1) / 2)
INFO	out	I	＝0：正常終了 ＝1：引数が不適当である。

11.24 複素三角行列の生成（非圧縮）JAERI_zdgdzrgtss

11.24.1 機能

三角行列となる複素数を要素とする乱数行列を生成する。

本サブルーチンは、主要入力データとして対角要素に関する指定、乱数の形式・範囲の指定などをとり、主要出力データとして非圧縮形式で格納された複素三角行列を返す。

11.24.2 コーリングシーケンス

```
    call JAERI_zdgdzrgtss
   &     (n,n1, upper,optdg,normal,u,v,method,seed,T,INFO)
```

11.24.3 引数

用法: in　入力用。値は壊されない。

out　出力用。

i/o　入出力用。入力として使用され、かつ戻り時に値が設定される。

w　作業用。作業領域として与えられる。戻り時の値は意味を持たない。

型 : I　整数型(integer)

D　倍精度実数型(double precision)

Z　倍精度複素数型(complex*16)

L　論理型(logical)

C　文字型(character * (*))

変数／配列(次元)	I/O	型	意味
n	in	I	行列の次数 (n1 ≧ n ≧ 1)
n1	in	I	配列Tの配列宣言での行数 (n1 ≧ 1)
upper	in	L	TRUE ： 上三角部分を出力する FALSE ： 下三角部分を出力する
optdg	in	I	対角要素に関する指定 ＝ 0 ： 対角要素の符号は任意、対角優位に関する指定無し ＜ 0 ： 対角要素の少なくとも１つは正の実数以外 ＞ 0 ： 対角要素は全て正の実数 |optdg| = 1 ： 行の対角優位を指定 = 2 ： 列の対角優位を指定 = 3 ： 行と列の対角優位を指定 = 10： 行の対角優位を指定しない = 20： 列の対角優位を指定しない = 30： 行と列の対角優位を指定しない
normal	in	L	TRUE ： 正規分布 FALSE ： 一様分布
u	in	D	normal = TRUE ならば、非対角要素の平均 normal = FALSE ならば、非対角要素の最小値
v	in	D	normal = TRUE ならば、非対角要素の標準偏差 (v > 0) normal = FALSE ならば、非対角要素の最大値 (v > u)
method	in	I	1 ： 乗算合同法によって乱数を生成する 2 ： 混合合同法によって乱数を生成する
seed	i/o	I	入力：乱数の種 seed ≦ 0 ならば、デフォルト値を使用する method ＝ 1 かつ seed ＞ 0 ならば、偶数でなければならない 出力：次のコールのための乱数の種
T(n1,*)	out	Z	生成された複素三角行列
INFO	out	I	＝ 0：正常終了 ＝ 1：引数が不適当である。

12. 試験行列：アプリケーションベース

12.1 3次元流体7点差分疎行列生成プログラム JAERI_dg3zpfdss

12.1.1 概要

3次元非圧縮流体解析に現れる圧力方程式の係数行列と右辺ベクトルおよび数値解をファイルに格納する。物理的境界条件として、細い直方体領域において、

- 入口で Poiseille flow の流速プロファイル
- 上下面ですべりなし、
- 左右面および出口で加速度なし、

を与える。

この条件のもとに3次元 Navier–Stokes 方程式をスタッガードメッシュに基づく7点差分で離散化し、速度場と圧力の交互反復計算によって数値解を得る。時間項は陽的差分によって扱われる。

収束時の流速分布は、計算領域の至る所でほとんど Poiseille flow となり、圧力は入口から出口に向かうx座標についてほとんど線形の関数となる。出力対象が、流速分布ではなく、圧力方程式である点に注意されたい。

12.1.2 プログラム利用方法

(1) プログラムの起動

実行形式のプログラムは a.out である。入力プロンプト% に対し、a.out を入力すればよい。

```
[起動コマンド]
% a.out   ←  タイプ後、リターン
```

(2) 格子分割数

格子分割数を指定する。下記の例では、x、y、z方向について、それぞれ51、3、21個の圧力格子点を生成することを指定している。

```
[格子分割数]
Enter grid counts L,M,N in X,Y,Z direction recpectively.
   Upper limits: L_max= 101, M_max= 5, N_max= 21
   For example, you can enter
                 101, 5, 21
51,3,21  ←  タイプ後、リターン
```

(3) 出力先ファイル名

試験行列を格納するためのファイル名を指定する。

```
[出力先ファイル名]
Enter a filename for saving a generated matrix and so on !
     (ex.) test.d
sample.d  ←  タイプ後、リターン
```

(4) 計算経過表示

以下のように、内部変数の一部、続いて計算経過が表示される。

```
[計算経過表示]

    ==========
     dg3zpfdss(Ver 1.0)
    ==========
    Number of lattice : 51 * 3 * 21
    .....
    = BiCG - STAB2 =   200    0.826843e-03
    = BiCG - STAB2 =   200    0.368667e-03
    = BiCG - STAB2 =   200    0.437487e-03
    = BiCG - STAB2 =   200    0.302890e-03
    = BiCG - STAB2 =   200    0.264785e-03
    = BiCG - STAB2 =   110    0.559254e-04
    = BiCG - STAB2 =    49    0.969006e-04
    = BiCG - STAB2 =    43    0.950204e-04
    .....
```

ここで、「= BiCG-STAB2 =」の右側の整数が、1タイムステップで反復に要した回数である。各タイムステップにおいて、最大200回のループで次のタイムステップに進む。その右側の浮動小数点表示が、残差ベクトル $Ax-b$ の絶対値最大値 ε を表し、この値が小さいほど当該タイムステップにおいて収束状況が良好であることを表す。本プログラムは、各タイムステップの収束状況の如何にかかわらず、最大で1000タイムステップ進んだ時点で、終了する。したがって、プログラム終了時の最後の ε が十分小さい値であることを確認しておく必要がある。なお、「最大200回のループ」、「1000タイムステップ」などの値は、ユーザがカスタマイズ可能である(12.1.5)。

(5) プログラムの終了

以下のようなメッセージが出力された時点で、試験行列がファイルに格納されたことになる。上記(4)に述べたように、ε の値が十分小さくない場合は、x および b の値は精度が低いことに注意されたい。試験行列が具体的にどのようにファイルに書き出されたかについては、12.1.3 を参照されたい。

```
[終了時メッセージ]
(dg2zpfdss) A test matrix and so on are generated.
     Matrix size (exclude dummy areas) = NNNNNNN1
     File                              = test.d
```

Matrix size	:	試験行列の格納される長さを示す(ダミーエリアを除いた長さ)。 NNNNNNN1 は8桁以下の整数である。
File	:	ユーザが指定した試験行列の格納先ファイルを示す。

12.1.3 試験行列の利用

本プログラムでは3次元7点差分疎行列、右辺ベクトル、数値解が、行単位にこの順番で、フォーマット無しの内部形式でファイルに書き出される。以下に、本プログラムが試験行列等を出力している部分のソースと変数の説明を参考までに示す。

```
      OPEN(16,FILE=FILENM,FORM='UNFORMATTED')
      WRITE(16) L,M,N
      DO 50 I =1,LMN
        IP = I + LM
        WRITE(16) (COEF(I,J),J=1,7), B(I), X(IP)
50    CONTINUE
      CLOSE(16)
```

(1) L、M、N：整数
　x、y、z 方向圧力格子数。

(2) COEF(∗ , 1:7)：倍精度実数2次元配列。
　圧力方程式の係数行列(3次元7点差分疎行列)。COEF(∗,1)、COEF(∗,2)、...、COEF(∗,7)が、A1(∗)、A2(∗)、...、A7(∗)に対応する。但し、本プログラム内のCOEFは、ダミー領域を持たない。

(3) B(∗)：倍精度実数1次元配列。
　圧力方程式の右辺ベクトル。本プログラム内のbは、ダミー領域を持たない。

(4) X(∗)：倍精度実数1次元配列。
　圧力の数値解。領域は、前後に l ∗ m の長さのダミー領域を持つが、出力対象は、非ダミー領域のみである。

なお、上記のソースプログラムで文字変数FILENMには、試験行列等を格納するファイル名が代入済みとする。この出力においては、疎行列、右辺ベクトル、解ベクトルのいずれもダミーでない部分のみがファイルに出力されている点に注意されたい。(疎行列、右辺ベクトルについては、もともとダミー領域を用意していない)。

ユーザが、ファイルから試験行列を読み込む場合のプログラム例をFig.12.1.3に示す。この例では、ダミー領域の長さをL ∗ M個としている。但し、疎行列の構造は2次元としている。処理例として、残差ベクトル $Ax-b$ の無限大ノルムを計算している。

Fig.12.1.3　3次元7点差分疎行列読込みプログラム例

```
      PROGRAM SAMP37
CC
CC A SAMPLE PROGRAM OF READING 3-D,7-POINTS SPARSE MATRIX.
CC
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (LMNMX =10000, IOFMX=500, LMNMX2=LMNMX + 2*IOFMX)
      REAL*8 AA(LMNMX2,1:7), BB(LMNMX2), XX(LMNMX2)
C/*** BEGIN ***/
      OPEN (15,FILE='test.d',FORM='UNFORMATTED')
      READ (15) LL,MM,NN
      LLMMNN = LL*MM*NN
      LLMM = LL*MM
      IF (LLMMNN+2*LLMM .GT. LMNMX2) THEN
        WRITE(*,*) 'ERROR '
        STOP
      ENDIF
CC    /****** BEGIN: ZERO CLEAR ******/
      DO   10 JJ=1,7
        DO   10 II=1,LLMMNN+2*LLMM
          AA      (II,JJ) = 0.0D0
   10 CONTINUE
      DO 12 II=1,LLMMNN+2*LLMM
        BB      (II) = 0.0D0
        XX      (II) = 0.0D0
   12 CONTINUE
CC    /****** END : ZERO CLEAR ******/
      DO 50 II =1,LLMMNN
        READ (15) (AA(LLMM+II,JJ),JJ=1,7),BB(LLMM+II),XX(LLMM+II)
   50 CONTINUE
      RESINF = 0.0D0
      DO 100 II=1,LLMMNN
        IP = LLMM + II
        RES = BB(IP      ) - (AA(IP ,1)*XX(IP   - LLMM   )
     &                        +AA(IP ,2)*XX(IP   - LL     )
     &                        +AA(IP ,3)*XX(IP   -       1)
     &                        +AA(IP ,4)*XX(IP            )
     &                        +AA(IP ,5)*XX(IP   +       1)
     &                        +AA(IP ,6)*XX(IP   + LL     )
     &                        +AA(IP ,7)*XX(IP   + LLMM   ) )
        RESINF = MAX ( RESINF , DABS(RES) )
  100 CONTINUE
      WRITE(*,600) LL    ,MM    ,NN    ,LLMMNN,LLMM ,RESINF
  600 FORMAT(/' ','/*** SAMP37 ***/',
     &       ' ',' LL,MM,NN          = ',I5, ',' ,I5, ',' ,I5,
     &      /' ',' LL*MM*NN & LL*MM = ',I5, ',' ,I5
     &      /' ',' INF-NORM OF AX-B = ',1PE14.7 )
      END
```

12.1.4 トラブルシューティング

典型的なエラーの原因と対処方法について述べる。

(1) 格子数が多すぎる。

```
(dg3zpfdss-E01) Your value exceeds the upper limits:
L_max= NNN1 , M_max= NNN2 , N_max= NNN3
  Reenter L, M and N !
```

意味 : 分割格子数が多すぎる。

原因 : 指定した分割格子数が、プログラム内の配列の大きさを越えてしまった。

L_max = x 方向最大（圧力）格子数。

M_max = y 方向最大（圧力）格子数。

N_max = z 方向最大（圧力）格子数。

対策 : 1) 最大格子数を超えないように、再度、L、 M、 N を入力しなおす。

2) プログラム内の配列の大きさを変更してコンパイルし直す。

変更箇所は、Include/cmvars というファイル内の、ll、 mm、 nn の値である (12.1.5)。

"make clearn" の後に、"make" を投入すればよい。

```
[Include/cmvars の中の修正個所]
parameter ( ll=101, mm=5, nn=21 )
```

(2) x、y 方向の格子分割数が奇数でない。

```
(dg3zpfdss-E02) L and M must be odd number for red-black/4-colored SOR.
  Reenter L, M and N !
```

意味 : 連立一次方程式の解法に red-black SOR または、4-colored SOR を採用しているにもかかわらず、L、M の値が奇数でない。

原因 : red-black SOR または、4-colored SOR の場合、L、M の両方が奇数でなければならない。

対策 : L、M が奇数になるように再度 L、M、N の値を入力しなおす。なお、N の値は偶数／奇数のいずれでもよい。

備考 : このエラーメッセージは、インストール時のソースファイルをそのままコンパイルして利用している時は、出力されない。

もし、ユーザが、サブルーチン datain 内の ilsolv の値の設定を'2' または'5' に変更し、連立一次方程式の解法として red-black SOR または、4-colored SOR を選択した場合で、L、M の値が奇数でないときに、本メッセージが現れる (Cf.12.1.5)。その理由は、通し番号 I が奇数の時に、その格子点の情報更新時に参照される格子点の通し番号 $I\pm1$、$I\pm l$、$I\pm l*m$ がいずれも偶数でなければならないからである。

(3) 収束していない。

```
(dg3zpfdss-W01) Perhaps NO CONVERGENCE !
```

意味　：最終タイムステップにおいて、収束せずに計算が終了した。

原因　：連立一次方程式の解法選択および問題の自由度数の大きさに対して、プログラム内で設定されているタイムステップ数が小さすぎる。

あるいは、反復計算に係わる諸パラメタの値が適当でない。

対策　：以下に要点を記す。サブルーチン／変数の意味等は12.1.5 を参照のこと。

1) タイムステップ数を増やす。

サブルーチン datain 内で設定している

lpend または linner

の値を変更する。

2) 1タイムステップ内での反復回数を増やす。

サブルーチン datain 内で設定している

maxitr

の値を変更する。

3) 連立一次方程式の解法ルーチンを変更する。

サブルーチン datain 内で設定している

ilsolv

の値を変更する。

備考　：本プログラムでは反復解法を用いているため、収束しない場合の万全の対策は一般に存在しない。場合によっては、プログラム自体の改良が必要になることがある。

12.1.5 プログラム仕様 dg3zpfdss

モジュール構成

dg3zpfdss は、以下のモジュールで構成される。

```
dg3zpfdss.f  :メインプログラム
+- clearv.f  :ゼロクリア
+- datain.f  :パラメタ設定
+- initvr.f  :よく使われる値ををあらかじめ設定しておく
+- mkcoef.f  :coef(*,7) 等の設定
+- ilvpcg.f  :ハイパープレーンのリストベクトルdd(*) の設定
+- setbnd.f  :境界条件の設定(u,v,w)
+- calcu1.f  :u1 の計算
+- calcv1.f  :v1 の計算
+- calcw1.f  :w1 の計算
+- setbcv.f  :境界条件の設定(u1,v1,w1)
+- caluvw.f  :圧力Poisson 方程式の計算
   +- linsor.f  :SOR natural ordering
   +- lsorrb.f  :SOR red-Black
   +- linpcg.f  :PCG natural ordering
   +- lpcghp.f  :PCG hyperplane
   +- lsor4c.f  :SOR 4-color
   +- lbcgs2.f  :BiCG-STAB, GP-BiCG, GP-BiCG(ω), BiCG-STAB2
```

インクルードファイル

共通パラメタ、共通変数は、以下のインクルードファイルに定義されている。

(1) ./Include/cmvars
格子分割数、配列の大きさなどのパラメタ、流速、圧力などの主要配列やレイノルズ数などの共通変数を宣言する。

(2) ./Include/cmbicg
連立一次方程式のサブルーチンが使用する作業領域を宣言する。

主要パラメタ／変数

型
- C　文字型 (character * (*))
- I　整数型 (integer)
- D　倍精度実数型 (double precision)

変数名／パラメタ名	型	意味
filenm	C	試験行列格納先ファイル名 (ユーザ入力、20 バイト以内)。
ll	I	x 方向圧力最大格子点数パラメタ (default:101)。
mm	I	y 方向圧力最大格子点数パラメタ (default: 5)。
nn	I	z 方向圧力最大格子点数パラメタ (default: 21)。
llmmnn	I	= ll * mm * nn 。
llmmnn2	I	= llmmnn + 2 * ll * mm
l	I	x 方向圧力格子点数 (ユーザ入力)。
m	I	y 方向圧力格子点数 (ユーザ入力)。
n	I	z 方向圧力格子点数 (ユーザ入力)。
lmn	I	= l * m * n 。
uinit	D	一様流の流速 (default:1.0)。
xlen	D	直方体流れ方向長さ (default:15.0)。
ylen	D	直方体幅 (default:1.0)。
zlen	D	直方体高さ (default:1.0)。
dx	D	x 方向格子間隔 (= xlenl)。
dy	D	y 方向格子間隔 (= ylenm)。
dz	D	z 方向格子間隔 (= zlenn)。
dt	D	時間刻み幅。dt は以下の安定条件を満たすように決める。 $\Delta t \leq \frac{1}{\frac{u}{\Delta x}+\frac{2\alpha}{\Delta x^2}}$ ここで、動粘性係数 $\alpha = \frac{uL}{Re}$。また、代表長さ L として"ylen" または "zlen" のうち、大きい方の値を採用している。さらに、"dt" の仮数部が、1、 2、5 のいずれかになるように調整している。
omega	D	連立一次方程式の解法に SOR タイプのもの (ilsolv=1,2,5) を使用しているときの緩和係数 ω の値 (default:1.8)。
eps	D	1 タイムステップ内の反復計算での解の変動 $\frac{\|x_{new}-x_{old}\|_\infty}{\|b\|_\infty}$ の収束判定ための ε 。 なお、x_{old}、x_{new} は、それぞれ反復前、反復後の解の値である。

ilsolv	I	連立一次方程式の解法選択(default:24= BiCG-STAB 2)。 1: SOR 2: red-black SOR 3: PCG (natural ordering) 4: PCG (hyperplane ordering) 5: 4-colored SOR 21: BiCG-STAB 22: GP-BiCG(ω) 23: GP-BiCG 24: BiCG-STAB 2
lpbgn	I	外側ループ開始値(default:1)。
lpend	I	外側ループ終了値(default:20)。
linner	I	内側ループ回数(default:50)。 備考: 外側ループの最後では流速プロファイルの中央の値 (理論解では 1.5 * uinit) を計算経過情報として出力するが、内側ループでは出力しない。 本プログラムでのタイムステップは、(lpend-lpbgn+1)*linner となる。
maxitr	I	1タイムステップ内での最大反復回数(default:200)。
itrcnt	I	1タイムステップ内での反復回数。
itrtot	I	上記 itrcnt 累積値(合計値)。
u (ll+1,mm+2,nn+2)	D	流速の x 成分。
v (ll+2,mm+1,nn+2)	D	流速の y 成分。
w (ll+2,mm+2,nn+1)	D	流速の z 成分。
u1(ll+1,mm+2,nn+2)	D	流速予測値の x 成分。
v1(ll+2,mm+1,nn+2)	D	流速予測値の y 成分。
w1(ll+2,mm+2,nn+1)	D	流速予測値の z 成分。
p (ll+2,mm+2,nn+2)	D	圧力(u1,v1,w1 を計算するための作業領域)。
coef(llmmnn,7)	D	圧力方程式の係数行列。 なお、本プログラム内ではダミー領域を設けていない。
b(llmmnn)	D	圧力方程式の右辺行列。 なお、本プログラム内ではダミー領域を設けていない。
x(llmmnn2)	D	圧力方程式を解いた解。ダミー領域有り。

12.2 2次元流体5点差分疎行列生成プログラム JAERI_dg2zpfdss

12.2.1 概要

2次元非圧縮流体解析に現れる圧力方程式の係数行列と右辺ベクトルおよび数値解をファイルに格納する。物理的境界条件として、横に細長い2次元長方形領域において、

- 入口で Poiseille flow の流速プロファイル
- 上下面ですべりなし、
- 出口で加速度なし、

を与える。座標軸は流れ方向をx軸、x軸と垂直方向をz軸とする(3次元xyz座標から、xz面を切り出してきたと考える)。

この条件のもとに2次元 Navier–Stokes 方程式をスタッガードメッシュに基づく5点差分で離散化し、速度場と圧力の交互反復計算によって数値解を得る。時間項は陽的差分によって扱われる。

収束時の流速分布は、計算領域の至る所でほとんど Poiseille flow となり、圧力は入口から出口に向かうx座標についてほとんど線形の関数となる。出力対象が、流速分布ではなく、圧力方程式である点に注意されたい。

12.2.2 プログラム利用方法

(1) プログラムの起動

実行形式のプログラムは a.out である。入力プロンプト% に対し、a.out を入力すればよい。

```
[起動コマンド]
% a.out   ←  タイプ後、リターン
```

(2) 格子分割数

格子分割数を指定する。下記の例では、x、 z 方向について、それぞれ51 、21 個の圧力格子点を生成することを指定している。

```
[格子分割数]
Enter grid counts L, N in X, Z direction recpectively.
   Upper limits: L_max= 101, N_max= 21
   For example, you can enter
                 101, 21
51, 21  ←  タイプ後、リターン
```

(3) 出力先ファイル名

試験行列を格納するためのファイル名を指定する。

```
［出力先ファイル名］
Enter a filename for saving a generated matrix and so on !
      (ex.) test.d
sample.d  ←  タイプ後、リターン
```

(4) 計算経過表示

以下のように、内部変数の一部、続いて計算経過が表示される。

```
［計算経過表示］


    ==========
      dg2zpfdss(Ver 1.0)
    ==========
    Number of lattice : 51 * 21
    .....
    = BiCG - STAB2 =   200     0.458034e-03
    = BiCG - STAB2 =   200     0.463688e-03
    = BiCG - STAB2 =   200     0.174947e-03
    = BiCG - STAB2 =   200     0.321351e-03
    = BiCG - STAB2 =   200     0.142303e-03
    = BiCG - STAB2 =   124     0.946801e-04
    = BiCG - STAB2 =   200     0.102339e-03
    = BiCG - STAB2 =    64     0.935495e-04
    .....
```

ここで、「= BiCG-STAB2 =」の右側の整数が、1タイムステップで反復に要した回数である。各タイムステップにおいて、最大200回のループで次のタイムステップに進む。その右側の浮動小数点表示が、残差ベクトル $Ax - b$ の絶対値最大値 ε を表し、この値が小さいほど当該タイムステップにおいて収束状況が良好であることを表す。本プログラムは、各タイムステップの収束状況の如何にかかわらず、最大で1000タイムステップ進んだ時点で、終了する。したがって、プログラム終了時の最後の ε が十分小さい値であることを確認しておく必要がある。なお、「最大200回のループ」、「1000タイムステップ」などの値は、ユーザがカスタマイズ可能である(12.2.5)。

(5) プログラムの終了

以下のようなメッセージが出力された時点で、試験行列がファイルに格納されたことになる。上記(4)に述べたように、ε の値が十分小さくない場合は、x および b の値は精度が低いことに注意されたい。試験行列が具体的にどのようにファイルに書き出されたかについては、12.2.3 を参照されたい。

```
［終了時メッセージ］
(dg2zpfdss) A test matrix and so on are generated.
      Matrix size (exclude dummy areas) = NNNNNNN1
      File                              = test.d
```

Matrix size	:	試験行列の格納される長さを示す(ダミーエリアを除いた長さ)。 NNNNNNN1 は8桁以下の整数である。
File	:	ユーザが指定した試験行列の格納先ファイルを示す。

12.2.3 試験行列の利用

本プログラムでは2次元5点差分疎行列、右辺ベクトル、数値解が、行単位にこの順番で、フォーマット無しの内部形式でファイルに書き出される。以下に、本プログラムが試験行列等を出力している部分のソースと変数の説明を参考までに示す。

```
      OPEN(16,FILE=FILENM,FORM='UNFORMATTED')
      WRITE(16) LL,NN
      DO 50 I =1,LN
        IP = I + LL
        WRITE(16) (COEF(I,J),J=2,6), B(I), X(IP)
50    CONTINUE
      CLOSE(16)
```

(1) LL、NN : 整数
　x、z 方向圧力格子数。

(2) COEF(* , 2:6) : 倍精度実数2次元配列。
　圧力方程式の係数行列(2次元5点差分疎行列)。COEF(*,2)、COEF(*,3)、...、COEF(*,6) が、A2(*)、A3(*)、...、A6(*) に対応する。但し、本プログラム内のCOEF は、ダミー領域を持たない。

(3) B(*) : 倍精度実数1次元配列。
　圧力方程式の右辺ベクトル。本プログラム内の B は、ダミー領域を持たない。

(4) X(*) : 倍精度実数1次元配列。
　圧力の数値解。領域は前後に LL の長さのダミー領域を持つが、出力対象は、非ダミー領域のみである。

なお、上記のソースプログラムで文字変数 FILENM には、試験行列等を格納するファイル名が代入済みとする。

この出力においては、疎行列、右辺ベクトル、解ベクトルのいずれもダミーでない部分のみがファイルに出力されている点に注意されたい。(疎行列、右辺ベクトルについては、もともとダミー領域を用意していない)。

ユーザが、ファイルから試験行列を読み込む場合のプログラム例をFig.12.2.3 に示す。この例では、?? の記述に従い、ダミー領域の長さをLL 個としている。但し、疎行列の構造は2次元としている。処理例として、残差ベクトル $Ax-b$ の無限大ノルムを計算している。

Fig.12.2.3　2次元5点差分疎行列読込みプログラム例

```
      PROGRAM SAMP25
CC
CC A SAMPLE PROGRAM OF READING 2-D,5-POINTS SPARSE MATRIX.
CC
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (LNMX =10000, IOFMX=500, LNMX2=LNMX + 2*IOFMX)
      REAL*8 AA(LNMX2,2:6), BB(LNMX2), XX(LNMX2)
C/*** BEGIN ***/
      OPEN (15,FILE='test.d',FORM='UNFORMATTED')
      READ (15) LL,NN
      LLNN = LL*NN
      IF (LLNN+2*LL .GT. LNMX2) THEN
        WRITE(*,*) 'ERROR '
        STOP
      ENDIF
CC    /****** BEGIN: ZERO CLEAR ******/
      DO   10 JJ=2,6
        DO   10 II=1,LLNN+2*LL
          AA      (II,JJ) = 0.0D0
   10 CONTINUE
      DO 12 II=1,LLNN+2*LL
        BB      (II) = 0.0D0
        XX      (II) = 0.0D0
   12 CONTINUE
CC    /****** END : ZERO CLEAR ******/
      DO 50 II =1,LLNN
        READ (15) (AA(LL+II,JJ),JJ=2,6),BB(LL+II),XX(LL+II)
   50 CONTINUE
      RESINF = 0.0D0
      DO 100 II=1,LLNN
        IP = LL + II
        RES = BB(IP      ) - (
     &                     +AA(IP ,2)*XX(IP   - LL     )
     &                     +AA(IP ,3)*XX(IP   -       1)
     &                     +AA(IP ,4)*XX(IP            )
     &                     +AA(IP ,5)*XX(IP   +       1)
     &                     +AA(IP ,6)*XX(IP   + LL     )
     &                     )
        RESINF = MAX ( RESINF , DABS(RES) )
  100 CONTINUE
      WRITE(*,600) LL    ,NN    ,LLNN,RESINF
  600 FORMAT(/' ','/*** SAMP25 ***/',
      &          ' ',' LL,NN,LLNN        = ',I5, ',' ,I5, ',' ,I5,
      &         /' ',' INF-NORM OF AX-B = ',1PE14.7 )
      END
```

12.2.4 トラブルシューティング

典型的なエラーの原因と対処方法について述べる。

(1) 格子数が多すぎる。

```
(dg2zpfdss-E01) Your value exceeds the upper limits:
L_max= NNN1 , N_max= NNN2
  Reenter L and N !
```

意味　：　分割格子数が多すぎる。

原因　：　指定した分割格子数が、プログラム内の配列の大きさを越えてしまった。

　　L_max = x 方向最大（圧力）格子数。

　　N_max = z 方向最大（圧力）格子数。

対策　：　1) 最大格子数を超えないように、再度、L、N を入力しなおす。

　　2) プログラム内の配列の大きさを変更してコンパイルし直す。

　　変更箇所は、Include/cmvars というファイル内の、LMAX、NMAX の値である (12.2.5)。

　　"make clearn" の後に、"make" を投入すればよい。

```
[Include/cmvars の中の修正個所]
parameter ( LMAX=101, NMAX=21 )
```

(2) x 方向の格子分割数が奇数でない。

```
(dg2zpfdss-E02) L must be odd number for red-black/4-colored SOR.
  Reenter L and N !
```

意味　：　連立一次方程式の解法に red-black SOR または、4-colored SOR を採用しているにもかかわらず、L の値が奇数でない。

原因　：　red-black SOR または、4-colored SOR の場合、L の値が奇数でなければならない。

対策　：　L が奇数になるように再度 L、N の値を入力しなおす。なお、N の値は偶数／奇数のいずれでもよい。

備考　：　このエラーメッセージは、インストール時のソースファイルをそのままコンパイルして利用している時は、出力されない。

　　もし、ユーザが、サブルーチン datain 内の ilsolv の値の設定を'2' または'5' に変更し、連立一次方程式の解法として red-black SOR または、4-colored SOR を選択した場合で、L の値が奇数でないときに、本メッセージが現れる (Cf.12.2.5)。その理由は、通し番号 I が奇数の時に、その格子点の情報更新時に参照される格子点の通し番号 $I \pm 1$、$I \pm L$ がいずれも偶数でなければならないからである。

(3) 収束していない。

```
(dg2zpfdss-W01) Perhaps NO CONVERGENCE !
```

意味　：　最終タイムステップにおいて、収束せずに計算が終了した。

原因　：　連立一次方程式の解法選択および問題の自由度数の大きさに対して、プログラム内で設定されているタイムステップ数が小さすぎる。

　　あるいは、反復計算に係わる諸パラメタの値が適当でない。

対策　：以下に要点を記す。サブルーチン／変数の意味等は12.2.5 を参照のこと。

1) タイムステップ数を増やす。

サブルーチン datain 内で設定している

lpend または linner

の値を変更する。

2) 1タイムステップ内での反復回数を増やす。

サブルーチン datain 内で設定している

maxitr

の値を変更する。

3) 連立一次方程式の解法ルーチンを変更する。

サブルーチン datain 内で設定している

ilsolv

の値を変更する。

備考　：本プログラムでは反復解法を用いているため、収束しない場合の万全の対策は一般に存在しない。場合によっては、プログラム自体の改良が必要になることがある。

12.2.5 プログラム仕様 dg2zpfdss

モジュール構成

dg2zpfdss は、以下のモジュールで構成される。

```
dg2zpfdss.f  :メインプログラム
+- clearv.f  :ゼロクリア
+- datain.f  :パラメタ設定
+- initvr.f  :よく使われる値ををあらかじめ設定しておく
+- mkcoef.f  :coef(*,2:6) 等の設定
+- ilvpcg.f  :ハイパープレーンのリストベクトルdd(*) の設定
+- setbnd.f  :境界条件の設定(u,w)
+- calcu1.f  :u1 の計算
+- calcw1.f  :w1 の計算
+- setbcv.f  :境界条件の設定(u1,w1)
+- caluvw.f  :圧力Poisson 方程式の計算
    +- linsor.f  :SOR natural ordering
    +- lsorrb.f  :SOR red-Black
    +- linpcg.f  :PCG natural ordering
    +- lpcghp.f  :PCG hyperplane
    +- lsor4c.f  :SOR 4-color
    +- lbcgs2.f  :BiCG-STAB, GP-BiCG, GP-BiCG(ω), BiCG-STAB2
```

インクルードファイル

共通パラメタ、共通変数は、以下のインクルードファイルに定義されている。

(1) ./Include/cmvars

格子分割数、配列の大きさなどのパラメタ、流速、圧力などの主要配列やレイノルズ数などの共通変数を宣言する。

(2) ./Include/cmbicg

連立一次方程式のサブルーチンが使用する作業領域を宣言する。

主要パラメタ／変数

型
- C　文字型(character ∗ (∗))
- I　整数型(integer)
- D　倍精度実数型(double precision)

変数名／パラメタ名	型	意　味
filenm	C	試験行列格納先ファイル名(ユーザ入力、20バイト以内)。
LMAX	I	x 方向圧力最大格子点数パラメタ(default:101)。
NMAX	I	z 方向圧力最大格子点数パラメタ(default: 21)。
LNMAX	I	= LMAX ∗ NMAX 。
LN2MAX2	I	= LNMAX + 2 ∗ LMAX
LL	I	x 方向圧力格子点数(ユーザ入力)。
NN	I	z 方向圧力格子点数(ユーザ入力)。
LN	I	= LL ∗ NN 。
uinit	D	一様流の流速(default:1.0)。
xlen	D	長方形領域流れ方向長さ(default:15.0)。
zlen	D	長方形領域高さ　　　　(default:1.0)。
dx	D	x 方向格子間隔(= xlenLL)。
dz	D	z 方向格子間隔(= zlenNN)。
dt	D	時間刻み幅。dt は以下の安定条件を満たすように決める。 $\Delta t \leq \frac{1}{\frac{u}{\Delta x}+\frac{2\alpha}{\Delta x^2}}$ ここで、動粘性係数 $\alpha = \frac{uL}{Re}$。また、代表長さLとして"zlen" を採用している。さらに、"dt" の仮数部が、1、 2、 5 のいずれかになるように調整している。
omega	D	連立一次方程式の解法に SOR タイプのもの(ilsolv=1,2,5) を使用しているときの緩和係数 ω の値(default:1.8)。
eps	D	1タイムステップ内の反復計算での解の変動 $\frac{\|x_{new}-x_{old}\|_\infty}{\|b\|_\infty}$ の収束判定ための ε 。 なお、x_{old}、x_{new} は、それぞれ反復前、反復後の解の値である。

ilsolv	I	連立一次方程式の解法選択(default:24= BiCG-STAB 2)。 1: SOR 2: red-black SOR 3: PCG (natural ordering) 4: PCG (hyperplane ordering) 5: 4-colored SOR 21: BiCG-STAB 22: GP-BiCG(ω) 23: GP-BiCG 24: BiCG-STAB 2
lpbgn	I	外側ループ開始値(default:1)。
lpend	I	外側ループ終了値(default:20)。
linner	I	内側ループ回数(default:50)。 備考: 外側ループの最後では流速プロファイルの中央の値 (理論解では 1.5 * uinit) を計算経過情報として出力するが、内側ループでは出力しない。 本プログラムでのタイムステップは、(lpend-lpbgn+1)*linner となる。
maxitr	I	1タイムステップ内での最大反復回数(default:200)。
itrcnt	I	1タイムステップ内での反復回数。
itrtot	I	上記 itrcnt 累積値(合計値)。
u (LMAX+1,NMAX+2)	D	流速の x 成分。
w (LMAX+2,LMAX+1)	D	流速の z 成分。
u1(LMAX+1,LMAX+2)	D	流速予測値の x 成分。
w1(LMAX+2,LMAX+1)	D	流速予測値の z 成分。
p (LMAX+2,LMAX+2)	D	圧力(u1,w1 を計算するための作業領域)。
coef(LNMAX ,2:6)	D	圧力方程式の係数行列。 なお、本プログラム内ではダミー領域を設けていない。
b(LNMAX)	D	圧力方程式の右辺行列。 なお、本プログラム内ではダミー領域を設けていない。
x(LN2MAX)	D	圧力方程式を解いた解。ダミー領域有り。

12.3 2次元有限要素法帯生成プログラム JAERI_dgbzpfess

有限要素法によって2次元応力問題を解く際に現れる剛性行列（左辺行列）と荷重ベクトル（右辺ベクトル）を生成して、ファイルに格納するプログラムである。これらの他に、変位ベクトルの理論的厳密解もファイルに格納する。ユーザは、剛性行列と荷重ベクトルをファイルから読み込んで適当な連立一次方程式解法ルーチンを用いて解き、その結果を理論的厳密解と比べることができる。プログラム使用方法、試験行列の形式および利用方法を示すものである。付録には、プログラムの保守に必要なドキュメントを付している。本プログラムを作成するにあたり、Smith [15] およびZienkiewics and Taylor [16] を参考にした。

12.3.1 プログラム利用方法

プログラムとデータの関係を下図に示す。このうち、「試験行列を利用するユーザプログラム」は、ユーザが自作するものである。試験行列の読み込み方法は、12.3.5 を参考にされたい。

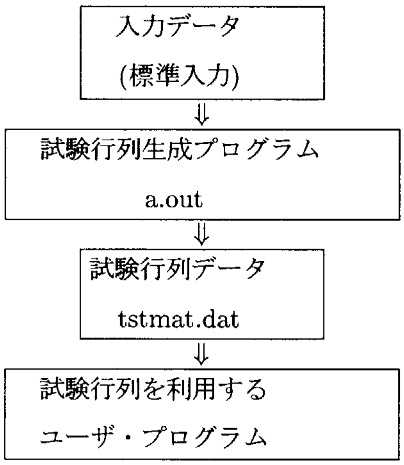

Fig. 12.3.1-1 The relations between programs and data.

12.3.2 入力データ

本プログラムで試験行列生成のために入力するデータは、以下に述べる5個のみである。入力先は標準入力である。

(1) タイトル (内部変数名 ＝ TITLE :文字型80バイト)
データのタイトルを0〜80バイト以内で記述する。

(2) x方向要素数 (内部変数名 ＝ NXE :整数)
x方向の要素数を指定する。

(3) y方向要素数 (内部変数名 ＝ NYE :整数)
y方向の要素数を指定する。

(4) x方向長さ (内部変数名 ＝ XLEN :単精度実数)
2次元長方形弾性体のx方向長さを指定する。

(5) y方向長さ (内部変数名 ＝ YLEN :単精度実数)
2次元長方形弾性体のy方向長さを指定する。

これらの入力データを用意することで、5.3.4節で述べた問題について、有限要素法の試験行列を生成させることができる。本プログラムでは上記入力データをフリーフォーマットで読み込むので、順序と属性を守ればカラム位置は任意でよい。例えば、以下のような要素分割を考える。

x方向要素数 = 2 ⇒ NXE として指定 y方向要素数 = 3 ⇒ NYE として指定

x方向長さ = 6.0 ⇒ XLEN として指定 y方向長さ = 9.0 ⇒ YLEN として指定

この場合の入力データは、例えば以下のように作ればよい。

```
TITLE: This is a comment .
2 3 6.0 9.0
```

本プログラムは、このデータに基づいて図 12.3.2-1 に示す要素分割および節点データ順序付けを行い、また図 12.3.2-2 に示す自由度順序番号付けを行う。ここで、(u,v) は変位のx,y方向成分についての未知数の順序番号の組み合わせを示し、数字ゼロは固定境界条件である(したがって未知数ではない)ことを表す。

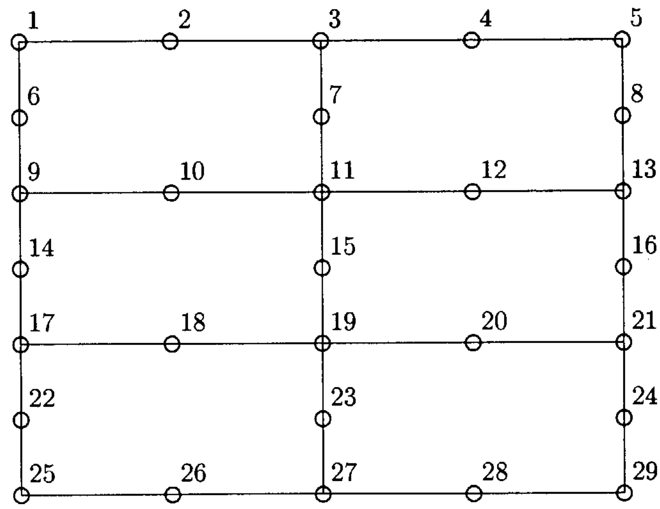

Fig. 12.3.2-1 Node numbering at the automatic mesh generation.

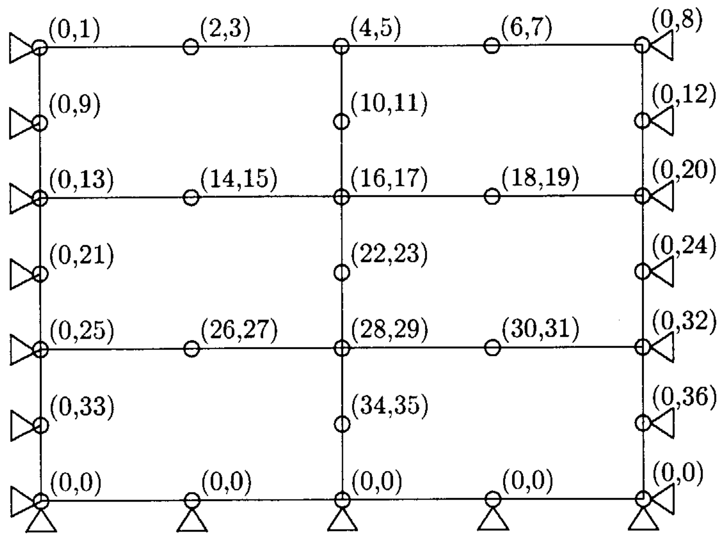

Fig. 12.3.2-2 Numbering of degrees of freedom at the automatic mesh generation.

12.3.3 実行方法

通常は、あらかじめ適当なファイルに入力データを格納しておき、リダイレクトによってプログラムにファイル内容を読み込ませるのがよい。実行形式のプログラムは a.out である。入力データが格納された

ファイルの名前を仮に case1.dat とすると以下のようなコマンドを入力すればよい。

```
[起動コマンド] a.out < case1.dat
```

以下のようなメッセージが出力された時点で、試験行列がファイルに格納されたことになる。試験行列の具体的格納形式等については、12.3.4章で述べる。

```
[終了時メッセージ] (DGBZPFESS-I000) A TEST MATRIX AND SO ON ARE GENERATED.
                  MATRIX SIZE = NNNNNNN1
                  HALF BAND WIDTH = NNNNNNN2
                  FILE = tstmat.dat
```

MATRIX SIZE	:	試験行列の次元を示す。NNNNNNN1 は8桁以下の整数である。
HALF BAND WIDTH	:	対角成分を除いた半バンド幅を示す。NNNNNNN2 は8桁以下の整数である。
FILE	:	試験行列の格納先ファイルを示す。通常は tstmat.dat である。

備考1: 試験行列格納先ファイルの変更方法：

試験行列格納先ファイルを tstmat.dat 以外に変更したい場合、メインプログラムのCFILOTの値を変更してコンパイルし直せばよい。

```
CHARACTER*32 CFILOT
DATA CFILOT /'fem2pl.mat'/
```

備考2: 有限要素解析結果の出力：

本プログラムでは、試験行列と荷重ベクトルを生成するだけでなく、それらを用いて解を求め、標準出力に出力することができる。

メインプログラムのLCALCの値を.TRUE.に変更してコンパイルし直せばよい。

```
LOGICAL LCALC
...
DATA LCALC / .TRUE. /
```

12.3.4 生成される試験行列

試験行列は、一般に対称帯行列となり、下三角部分がファイルに書き出される。また、右辺ベクトルと厳密解も同時に出力される。

ファイルには、フォーマットなしの内部形式で出力される。以下に、本プログラムが試験行列等を出力している部分の記述と変数の説明を示す。

```
      WRITE(IUNIT) N ,IW
      DO 10 II=1,N
        WRITE(IUNIT) (KB(II,JJ),JJ=1,IW+1),LOADS (II),ANSUV (II)
10    CONTINUE
```

(1) N：整数

総自由度数。解くべき方程式の変数の数と解釈してよい。

(2) IW：整数

対称帯行列の対角成分を除いた半バンド幅。

(3) KB(*,*)：単精度実数2次元配列。

剛性行列。対称帯行列の下三角部分。

(4) LOADS(*)：単精度実数1次元配列。

右辺ベクトル。

(5) NASUV(*)：単精度実数1次元配列。

完全弾性体2次元応力問題の理論解。

対称帯行列 KB での格納内容は以下のようになる。すなわち、正方行列 A の i,j 成分($j \le i$)が、$KB(i, j-i+IW+1)$ に格納されていると考える。

以下に簡単な例を示す。

$$
\begin{array}{c}
\begin{array}{ccccc} 1 & 2 & 3 & \cdots & N \end{array} \\
A = \begin{pmatrix}
a_{1,1} & & & & \\
a_{2,1} & a_{2,2} & & (\mathrm{sym.}) & \\
a_{3,1} & a_{3,2} & a_{3,3} & & \\
0 & a_{4,2} & a_{4,3} & a_{4,4} & \\
0 & 0 & a_{5,3} & a_{5,4} & a_{5,5}
\end{pmatrix}
\end{array}
\Rightarrow
\begin{array}{c}
\begin{array}{ccccc} 1 & \cdots \mathrm{IW\text{-}1} & \mathrm{IW} & \mathrm{IW+1} \end{array} \\
KB = \begin{pmatrix}
0 & \cdots & 0 & 0 & a_{1,1} \\
0 & \cdots & 0 & a_{2,1} & a_{2,2} \\
 & \cdots & a_{3,1} & a_{3,2} & a_{3,3} \\
 & \cdots & a_{4,2} & a_{4,4} & a_{4,4} \\
 & \cdots & a_{5,3} & a_{5,4} & a_{5,5}
\end{pmatrix}
\end{array}
$$

12.3.5 試験行列の利用

生成した試験行列等を利用するためには、ファイルから読み込まなくてはならない。本節では、出力した形式と同じ形式で対称帯行列の下三角部分に読み込む方法についてサブルーチン例を示す。

```
      SUBROUTINE MATINB(CFILIN,MAXN   ,MAXBND
     &                        ,N     ,IW     ,KB     ,LOADS ,ANSUV )
CC
CE     THIS SUBROUTINE READS THE STIFFNESS BAND MATRIX
CE       READ   TYPE = REAL SYMMETRIC BAND MATRIX (LWR TRIANGLE)
CE       STORED TYPE = REAL SYMMETRIC BAND MATRIX (LWR TRIANGLE)
CC
C/* IN */
      CHARACTER*32 CFILIN
CJ        試験行列格納ファイル名
      INTEGER      MAXN   ,MAXBND
CJ        総自由度数最大値、半バンド幅最大値
C/* OUT */
      INTEGER      N
CJ        総自由度数
      INTEGER      IW
CJ        半バンド幅
      REAL         KB    (MAXN   ,MAXBND)
CJ        剛性行列（対称帯行列の下三角部分を格納）
      REAL         LOADS (MAXN   )
CJ        右辺荷重ベクトル
      REAL         ANSUV (MAXN   )
CJ        変位理論解
C/* OUT */
CC    (NONE)
C/* CONSTANTS */
      INTEGER      IUNIT
      LOGICAL      LDEBUG
      DATA         IUNIT / 50 /
      DATA         LDEBUG / .TRUE. /
C/*** BEGIN ***/
      OPEN (UNIT=IUNIT,FILE=CFILIN,FORM='UNFORMATTED')
      READ (IUNIT) N   ,IW
      IF ( N.GT.MAXN   .OR. IW+1.GT.MAXBND ) THEN
```

```
          WRITE(6,*) 'INSUFFICIENT REGION SIZE.'
          WRITE(6,*) '  MAXN   MUST >= N    =',N
          WRITE(6,*) '  MAXBND MUST >= IW+1=',IW+1
          STOP
      ENDIF
      DO 10 II=1,N
        READ (IUNIT) (KB(II,JJ),JJ=1,IW+1),LOADS (II),ANSUV (II)
   10 CONTINUE
      CLOSE(IUNIT)
      RETURN
C/*** MATINB ***/
      END
```

また、本ツールで生成される剛性行列と荷重ベクトルが離散化誤差を含むものであるのに対し、変位ベクトルの厳密解は連続体を仮定して得た丸め誤差のみのものであるので、本ツールで生成した A、x、b について $Ax-b=0$ が必ずしも成り立つわけではない。したがって、上記プログラムの実行結果である数値解と理論解との差は、必ずしもゼロにならないことに注意されたい。

12.3.6 トラブル・シューティング等

ここでは典型的なエラーに対する原因と対処方法について述べる。

(1) 問題の総自由度数が大きすぎる。

```
(DGBZPFESS-E001) INSUFFICIENT REGION SIZE FOR UNKNOWN-VAR-COUNT.
                 MAXN MUST >= NNN3 AND NNN4
```

意味　：　解こうとしている問題の総自由度数が大きすぎる。

原因　：　解こうとしている問題の総自由度数に比べて、プログラム内で用意している配列の大きさが小さすぎる。NNN3 は、解こうとしている問題の総自由度数を表す。NNN4 は、解こうとしている問題の節点数を表す。

対策　：　メインプログラム内のPARAMETER文の MAXN の値を、上記 NNN3 、 NNN4 のいずれよりも大きくなるように変更してコンパイルし直す。

```
PARAMETER (MAXN=400,MAXBND=100)
```

(2) バンド幅が大きすぎる。

```
(DGBZPFESS-E002) INSUFFICIENT REGION SIZE FOR HALF-BAND-WIDTH .
                 MAXBND MUST >= NNN5
```

意味　：　解こうとしている問題のバンド幅が大きすぎる。

原因　：　解こうとしている問題のバンド幅に比べて、プログラム内で用意している2次元配列のバンド幅の大きさが小さすぎる。NNN5 は、解こうとしている問題のバンド幅を表す。

対策　：　メインプログラム内のPARAMETER文の MAXBND の値を、上記 NNN5 よりも大きくなるように変更してコンパイルし直す。

```
PARAMETER(MAXN=400,MAXBND=100)
```

12.3.7 プログラム仕様：モジュール構成

プログラム JAERI_dgbzpfess は、以下のモジュールで構成される。

```
JAERI_dgbzpfess メインプログラム
 +--CLRVEC ベクトルをゼロクリアする。
 +--DGEN28 分割数等に基づき、有限要素解析に必要なデータを生成する。
 +--CLRMAT 行列をゼロクリアする。
 +--FORMD  ２次元応力－歪行列(3*3)を生成する。
 +--GAUSCO ガウス求積法の重み係数と分点位置を格納する。
 +--TAB2D8 ２次元要素８節点の座標と全体系の節点との対応表を生成する。
 +--SHP2D8 形状関数の要素座標系に対する導関数を求める。
 +--MATMAT 行列と行列の積を求める。
 +--INV2X2 2*2行列の逆行列を求める。
 +--FORMB  歪－変位行列を生成する。
 +--MATRAN 転置行列を求める。
 +--MATSCA 行列をスカラー倍する。
 +--MATADD 行列と行列の和を求める。
 +--FORMKB 要素行列を全体行列（対称帯行列の下三角部分）に重ね合わせる。
 +--MATOUT 生成した行列、右辺ベクトル等をファイルに出力する。
 +--CHOLES コレスキー分解する（圧縮形式対称帯行列）。
 +--CHOSUB コレスキー分解の結果を用いて後退代入により解を得る。
 +--PRINTV ベクトルの内容を標準出力に出力する。
 +--MATVEC 行列とベクトルの積を求める。
```

12.3.8 プログラム仕様：主要パラメタ

主要パラメタは以下の通りである。

MAXN	:	INTEGER
	:	総自由度数許容最大値。
	:	出荷時のデフォルト値は400。
MAXBND	:	INTEGER
	:	半バンド幅+1の値の許容最大値。
	:	出荷時のデフォルト値は100。
IKB1	:	INTEGER
	:	剛性行列の行数(変数の数)の許容最大値。
	:	通常は MAXN の値と等しい値とする。
IKB2	:	INTEGER
	:	剛性行列の列数(半バンド幅+1)の許容最大値。
	:	通常は MAXBND の値と等しい値とする。
ILOADS	:	INTEGER
	:	荷重ベクトルLOADS(*)の長さの許容最大値。
	:	通常は MAXN の値と等しい値とする。
INF	:	INTEGER
	:	境界固定の有無を示す配列NF(*,2)の第1添え字の長さの許容最大値。
	:	通常は MAXN の値と等しい値とする。

主要パラメタ間の関係は以下の通りである。

```
PARAMETER(MAXN=400,MAXBND=100)
PARAMETER(IKB1=MAXN,IKB2=MAXBND,ILOADS=MAXN,INF=MAXN)
```

つまり、領域不足等が起きた場合は、総自由度数許容最大値 MAXN または半バンド幅+1 許容最大値 MAXBND のいずれかを調整すればよい。

12.3.9 プログラム仕様：主要変数

主要変数の説明を以下に示す。

TITLE	:	CHARACTER*80
	:	データタイトル。
MODE(1)	:	INTEGER
	:	モード・スイッチ。 1を指定する。
NXE	:	INTEGER
	:	X方向要素数。
NYE	:	INTEGER
	:	Y方向要素数。
XLEN	:	REAL
	:	X方向長さ。
YLEN	:	REAL
	:	Y方向長さ。
N	:	INTEGER
	:	総自由度数=方程式の次元(u,v 未知数の合計)
IW	:	INTEGER
	:	対角成分を含まない半バンド幅(全バンド幅=2*IW+1)
NN	:	INTEGER
	:	分割ノード数。
	:	u,vの自由度の有無を考慮せずに節点を数え上げたもの。
NR	:	INTEGER
	:	拘束条件の数(u,v について別々に数える)。
NGP	:	INTEGER
	:	ガウス求積法の選択番号(=プログラム内で3に固定:2次元8節点)
ELENX	:	REAL
	:	要素のX方向長さ。
ELENY	:	REAL
	:	要素のY方向長さ。
E	:	REAL
	:	ヤング率(=1.0E06)。
V	:	REAL
	:	ポアソン比(=0.3)。

KB(*,*) : REAL
: 剛性行列（実数帯対称の下三角部分を格納)。

LOADS(*) : REAL
: 右辺ベクトル。連立一次方程式のルーチンを呼ぶ前は荷重ベクトル、呼んだ後は変位ベクトルが格納されている。

ANSUV(*) : REAL
: 変位解の厳密解(u,v成分が混在しているので注意すること)。

NF(*,2) : INTEGER
: 境界固定の有無(I番目ノードのu,v成分について)。
: 0のとき固定、1のとき自由(未知数)。

CFILOT : CHARACTER*80
: 試験行列データ等の格納先ファイル。
: 初期値は、'tstmat.dat'。

12.3.10 実行環境とインストール手順

実行環境は以下の通りである。

［実行環境］

(1) Fortran コンパイラ

本プログラムのソースファイルをコンパイルして、実行形式を作成するために Fortran 77 以上が必要である。通常の UNIX ワークステーションにおいて普及している"f77" 等でよい。

(2) 計算機

上記の Fortran コンパイラを実行するためとコンパイル済みの本プログラム実行形式を実行するための計算機が必要である。通常の UNIX ワークステーションでよい。Fortran コンパイラが使用可能であれば、パソコンまたは汎用計算機でもよい。

インストール手順を以下に述べる。ここでは、通常の UNIX ワークステーションを仮定する。

［インストール手順］

(1) 本プログラムのソースファイルの複写

適当なディレクトリを必要に応じて作成し、その下に本プログラムのソースファイルやコンパイル時に make コマンドが参照する 'Makefile' 等を格納する。

(2) 'Makefile' の修正

使用するコンパイラの名称に応じて、'Makefile' を適宜修正する。'Makefile' については、下記で例を挙げて解説する。

(3) コンパイル

make コマンドを投入し、正常に終了することを確認する。本プログラムの実行形式 'fem2pl.exe' が生成されていることを、UNIX の 'ls -l' コマンド等で確認する。

参考までに 'Makefile' の例を以下に示す。

```
## [makefile]
##
## /* BEGIN:Modify here ! */
FL      =ftn
        FFLAGS =-O3
## /* END  :Modify here ! */
fem2pl.exe:      fem2pl.o femlib.o
$(FL)  fem2pl.o femlib.o \
-o fem2pl.exe  $(FFLAGS)
```

Fig. A.4-1 An example of 'Makefile'

'Makefile' の中で、変更の可能性があるのは、変数 'FL' と 'FFLAGS' である。

```
FL =ftn

FFLAGS =-O3
```

変数 'FL' には、Fortran コンパイラを起動するためのコマンド名を与える。上記例では、'ftn' がコマンド名であることを示す。もし、自分のシステムでのFortran コンパイラのコマンド名が'f77' の場合は、そのように変更する必要がある。

変数 'FFLAGS' には、コンパイラ・オプションを与える。上記例では、'-O3' が指定されているので、「レベル３の最適化」が行われることになる。ユーザの利用するFortran コンパイラのコンパイラ・オプションを調べた上で、適切なパラメタに変更する必要がある。

国際単位系(SI)と換算表

表1　SI基本単位および補助単位

量	名　称	記　号
長さ	メートル	m
質量	キログラム	kg
時間	秒	s
電流	アンペア	A
熱力学温度	ケルビン	K
物質量	モル	mol
光度	カンデラ	cd
平面角	ラジアン	rad
立体角	ステラジアン	sr

表3　固有の名称をもつSI組立単位

量	名　称	記号	他のSI単位による表現
周波数	ヘルツ	Hz	s^{-1}
力	ニュートン	N	$m{\cdot}kg/s^2$
圧力，応力	パスカル	Pa	N/m^2
エネルギー，仕事，熱量	ジュール	J	N·m
工率，放射束	ワット	W	J/s
電気量，電荷	クーロン	C	A·s
電位，電圧，起電力	ボルト	V	W/A
静電容量	ファラド	F	C/V
電気抵抗	オーム	Ω	V/A
コンダクタンス	ジーメンス	S	A/V
磁束	ウェーバ	Wb	V·s
磁束密度	テスラ	T	Wb/m^2
インダクタンス	ヘンリー	H	Wb/A
セルシウス温度	セルシウス度	℃	
光束	ルーメン	lm	cd·sr
照度	ルクス	lx	lm/m^2
放射能	ベクレル	Bq	s^{-1}
吸収線量	グレイ	Gy	J/kg
線量当量	シーベルト	Sv	J/kg

表2　SIと併用される単位

名　称	記　号
分，時，日	min, h, d
度，分，秒	°, ′, ″
リットル	l, L
トン	t
電子ボルト	eV
原子質量単位	u

$1\ eV = 1.60218 \times 10^{-19}\ J$

$1\ u = 1.66054 \times 10^{-27}\ kg$

表4　SIと共に暫定的に維持される単位

名　称	記　号
オングストローム	Å
バーン	b
バール	bar
ガル	Gal
キュリー	Ci
レントゲン	R
ラド	rad
レム	rem

$1\ Å = 0.1\ nm = 10^{-10}\ m$

$1\ b = 100\ fm^2 = 10^{-28}\ m^2$

$1\ bar = 0.1\ MPa = 10^5\ Pa$

$1\ Gal = 1\ cm/s^2 = 10^{-2}\ m/s^2$

$1\ Ci = 3.7 \times 10^{10}\ Bq$

$1\ R = 2.58 \times 10^{-4}\ C/kg$

$1\ rad = 1\ cGy = 10^{-2}\ Gy$

$1\ rem = 1\ cSv = 10^{-2}\ Sv$

表5　SI接頭語

倍数	接頭語	記　号
10^{18}	エクサ	E
10^{15}	ペタ	P
10^{12}	テラ	T
10^{9}	ギガ	G
10^{6}	メガ	M
10^{3}	キロ	k
10^{2}	ヘクト	h
10^{1}	デカ	da
10^{-1}	デシ	d
10^{-2}	センチ	c
10^{-3}	ミリ	m
10^{-6}	マイクロ	μ
10^{-9}	ナノ	n
10^{-12}	ピコ	p
10^{-15}	フェムト	f
10^{-18}	アト	a

(注)

1. 表1－5は「国際単位系」第5版，国際度量衡局 1985年刊行による。ただし，1 eV および1 uの値はCODATAの1986年推奨値によった。
2. 表4には海里，ノット，アール，ヘクタールも含まれているが日常の単位なのでここでは省略した。
3. barは，JISでは流体の圧力を表わす場合に限り表2のカテゴリーに分類されている。
4. EC閣僚理事会指令ではbar，barnおよび「血圧の単位」mmHgを表2のカテゴリーに入れている。

換　　算　　表

力	$N(=10^5 dyn)$	kgf	lbf
	1	0.101972	0.224809
	9.80665	1	2.20462
	4.44822	0.453592	1

粘　度　$1\ Pa{\cdot}s(N{\cdot}s/m^2) = 10\ P$(ポアズ)$(g/(cm{\cdot}s))$

動粘度　$1\ m^2/s = 10^4 St$(ストークス)(cm^2/s)

圧力	MPa(=10 bar)	kgf/cm^2	atm	mmHg(Torr)	lbf/in^2(psi)
	1	10.1972	9.86923	7.50062×10^3	145.038
	0.0980665	1	0.967841	735.559	14.2233
	0.101325	1.03323	1	760	14.6959
	1.33322×10^{-4}	1.35951×10^{-3}	1.31579×10^{-3}	1	1.93368×10^{-2}
	6.89476×10^{-3}	7.03070×10^{-2}	6.80460×10^{-2}	51.7149	1

エネルギー・仕事・熱量	$J(=10^7 erg)$	kgf·m	kW·h	cal(計量法)	Btu	ft·lbf	eV
	1	0.101972	2.77778×10^{-7}	0.238889	9.47813×10^{-4}	0.737562	6.24150×10^{18}
	9.80665	1	2.72407×10^{-6}	2.34270	9.29487×10^{-3}	7.23301	6.12082×10^{19}
	3.6×10^6	3.67098×10^5	1	8.59999×10^5	3412.13	2.65522×10^6	2.24694×10^{25}
	4.18605	0.426858	1.16279×10^{-6}	1	3.96759×10^{-3}	3.08747	2.61272×10^{19}
	1055.06	107.586	2.93072×10^{-4}	252.042	1	778.172	6.58515×10^{21}
	1.35582	0.138255	3.76616×10^{-7}	0.323890	1.28506×10^{-3}	1	8.46233×10^{18}
	1.60218×10^{-19}	1.63377×10^{-20}	4.45050×10^{-26}	3.82743×10^{-20}	1.51857×10^{-22}	1.18171×10^{-19}	1

1 cal = 4.18605 J (計量法)
　= 4.184 J (熱化学)
　= 4.1855 J (15 °C)
　= 4.1868 J (国際蒸気表)

仕事率　1 PS (仏馬力)
　= 75 kgf·m/s
　= 735.499 W

放射能	Bq	Ci
	1	2.70270×10^{-11}
	3.7×10^{10}	1

吸収線量	Gy	rad
	1	100
	0.01	1

照射線量	C/kg	R
	1	3876
	2.58×10^{-4}	1

線量当量	Sv	rem
	1	100
	0.01	1

(86年12月26日現在)

誤差評価ライブラリ(Ver.1.0)　理論説明書／利用手引書