JAERI-Data/Code
97-011

複素数行列計算精度評価ツール(Ver.1.0)利用手引書

1997年3月

市原　潔

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

本レポートは，日本原子力研究所が不定期に公刊している研究報告書です。
入手の問合わせは，日本原子力研究所研究情報部研究情報課（〒319-11 茨城県那珂郡東海村）あて，お申し越しください。なお，このほかに財団法人原子力弘済会資料センター（〒319-11 茨城県那珂郡東海村日本原子力研究所内）で複写による実費頒布をおこなっております。

This report is issued irregularly.
Inquiries about availability of the reports should be addressed to Research Information Division, Department of Intellectual Resources, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken 319-11, Japan.

編集兼発行　日本原子力研究所
印　　刷　㈱原子力資料サービス

JAERI－Data/Code 97－011

複素数行列計算精度評価ツール（Ver.1.0)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

（1997年２月13日受理）

「複素数行列計算精度評価ツール」（以下、「本ツール」）は，複素数連立一次方程式およびエルミート行列固有値問題を数値計算によって求めた場合の誤差の分析に必要なデータ取得を支援するためのサブルーチン集である。複素数連立一次方程式とエルミート行列固有値問題のそれぞれについて，逐次計算機用のものと並列計算機用のものとが用意されている。

複素数連立一次方程式については，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については，Korn-Kato の公式に基づいて固有値の存在範囲を計算する。

本書は，本ツールで使用している精度評価公式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 東京都目黒区中目黒２－２－54

JAERI−Data/Code 97−011

A User's Manual of Tools for Error Estimation of Complex Number Matrix Computation(Ver.1.0)

Kiyoshi ICHIHARA

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

(Received February 13, 1997)

"Tools for Error Estimation of Complex Number Matrix Computation" is a subroutine library which aids the users in obtaining the error ranges of the complex number 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 error estimation subroutines for Hermitian matrix eigen values' derive the error ranges of the eigen values according to the Korn-Kato's formula.

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, Numerical Analysis

目　　次

Contents

1. 序　　論

厳密解が既知でない連立一次方程式や行列の固有値問題を解いたときの結果の妥当性をチェックする簡便な方法として、いわゆる残差計算がある。すなわち、残差ベクトル $Ax-b$ または、$Ax-\lambda x$ を計算して、それがゼロに近いことをもって、「解がそれほど誤ったものではない」とみなすものである。

しかしこの方法では、残差が小さいことがわかったとしても、数値解自体がどの程度の精度なのかは把握できない。また、万一数値解が大きな誤差を含むことが判明した場合、プログラムの不具合によるものなのか、行列が悪条件で本質的に解きがたいものなのかを、にわかに判別することができない。

「複素数行列計算精度評価ツール」(以下、「本ツール」)は、複素数を係数に持つ連立一次方程式(以下、「複素数連立一次方程式」)およびエルミート行列固有値問題の解を数値計算によって求めた場合の計算結果を、的確かつ簡便に検査することを支援するサブルーチン集である。複素数の連立一次方程式とエルミート行列固有値問題のそれぞれについて、逐次計算機用のものと分散メモリ型並列計算機用のものとが用意されている。これらを用いることで、残差ベクトルのノルム以外に、条件数や誤差限界等の情報を得ることができ、万一解が不当である場合の原因の切り分けがより容易になる。特に分散メモリ型並列計算機では、行列が複数のプロセッサエレメント(PE)に分散配置されている関係上、誤差に係わる情報を入手するためのプログラミングが逐次計算機ほど容易ではないので、行列計算サブルーチンの改良等に際して精度チェックの必要が生じた場合に、本ツールは有用であると考えられる。

複素数連立一次方程式については、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については、Korn-Kato の公式に基づいて固有値の存在範囲を計算する。

「複素数行列計算精度評価ツール(Ver.1.0)利用手引書」(以下、「本書」)は、本ツールの数学的背景と利用方法を説明したものである。

2. 複素数行列計算精度評価ツールの概観

2.1 サブルーチン一覧

本書で解説の対象とするサブルーチンを以下に一覧する。

(1) サブルーチン名: SZGEES (Cf. 5.1)

- 機能　　: 連立一次方程式の条件数、誤差限界等の計算
- 逐次／並列: 逐次計算機
- 行列の型　: 倍精度複素数密行列
- 行列の配置: 単一メモリ配置

(2) サブルーチン名: PZGEES (Cf. 5.2)

- 機能　　: 連立一次方程式の条件数、誤差限界等の計算
- 逐次／並列: 分散メモリ型並列計算機(MPI)
- 行列の型　: 倍精度複素数密行列
- 行列の配置: 分散メモリ上に、列サイクリック分割縮小配置

(3) サブルーチン名: SQHE3R (Cf. 5.3)

- 機能　　: エルミート行列固有値問題の固有値誤差の計算
- 逐次／並列: 逐次計算機
- 行列の型　: 倍精度複素数のエルミート密行列
- 行列の配置: 単一メモリ配置

(4) サブルーチン名: PLHE3R (Cf. 5.4)

- 機能　　: エルミート行列固有値問題の固有値誤差の計算
- 逐次／並列: 分散メモリ型並列計算機(MPI)
- 行列の型　: 倍精度複素数のエルミート密行列
- 行列の配置: 分散メモリ上に、列サイクリック分割縮小配置

2.2 開発言語・プロセッサ間通信

開発言語は FORTRAN77 である。分散メモリ並列計算機で利用するサブルーチンでは、プロセッサ間通信に、MPI (Message Passing Interface) [4] を用いている。

2.3 利用上の留意点

本ツールはソースプログラムの形で提供されるため、ユーザ自身がコンパイルする必要がある。

逐次計算機で利用するサブルーチンについては、特に注意すべき点はない。

分散メモリ並列計算機で利用するサブルーチンの場合、SPMD モデル (Single Program Multiple Data Model) で並列化されたメイン・プログラムからコールされることを前提とする。すなわち、並列プロセス内で本ツールのサブルーチンを呼び出す。

行列データの分割は予め行っておく (Cf. 4.2)。

2.4 開発環境

本ツールの開発および動作確認は、日本原子力研究所計算科学技術推進センターのIBM SP2 (IBM RISC システム/6000 SP), AIX Version 4 で行った。

3. 行列計算の誤差解析

本章では、本ツールに含まれる精度評価サブルーチンが、どのような公式に基づいて誤差限界等を計算しているかを示す。

最初の節では、本ツールで使用している公式の一覧を与える。ユーザが既に線形代数の誤差解析等に関する予備知識を持っている場合には、この節だけで本ツールの機能が把握できるであろう。

3.2 節には、連立一次方程式の誤差解析、3.3 節にはエルミート行列の誤差解析について要点を説明する。公式の証明など、より詳細な理論については、巻末に挙げた参考書等を参照されたい。なお、以下の説明で $\|A\|$ は、A のノルムを表す（Cf.3.2.2）。ノルムには、1–ノルム、2–ノルム、無限大–ノルムなど幾つかの種類があるが、それらを区別する場合にはそれぞれ $\|A\|_1$、$\|A\|_2$、$\|A\|_\infty$ などと書くことにする。特定のノルムに限定しない場合は単に $\|A\|$ と書くことにする。

3.1 使用公式の要約

本ツールで使用している公式を概観するため、説明抜きで列挙する。
式の意味、式中に含まれる記号等の説明は、3.2節以降を参照されたい。

(1) 連立一次方程式誤差解析

関連するサブルーチン：逐次計算機用　SZGEES（Cf. 5.1）
並列計算機用　PZGEES（Cf. 5.2）

(a) 公式1

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

(b) 公式2

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

(2) エルミート行列固有値問題誤差解析

関連するサブルーチン：逐次計算機用　SQHE3R（Cf. 5.3）
並列計算機用　PLHE3R（Cf. 5.4）

(a) Korn–Kato の公式

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

ここで、

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

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

3.2 連立一次方程式の解の誤差解析

本節に述べる公式は、連立一次方程式の係数行列の成分が実数、複素数のいずれの場合でも成り立つので、特に「複素数」であることを断らずに説明を進める。

3.2.1 誤差解析の公式

問題の設定としては、

$$Ax = b \tag{1}$$

を満たす真の解を x、数値解法による解(数値解)を $\tilde{x}$ とするとき、相対誤差

$$\frac{\|x - \tilde{x}\|}{\|x\|} \tag{2}$$

の絶対値の上限、すなわち誤差限界をいかにして見積るかということになる。

まず最初に「LAPACK利用の手引」p.65 [8] に述べられている「4.3.1 標準誤差解析」から、「通常の誤差限界」の公式を引用する。

[公式1]

$$\frac{\|x - \tilde{x}\|}{\|x\|} \leq \kappa(A)\ \|\frac{\delta}{z}\| \approx \kappa(A)\ \epsilon \tag{3}$$

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

$\frac{\delta}{z}$: 相対後退誤差.

ϵ : マシンイプシロン.

ここで、$\frac{\delta}{z}$ は、相対後退誤差を表し [8] 、通常はマシンイプシロン ϵ で代替可能である。ここで、一番右側の右辺が、A の条件数とマシンイプシロンだけであらわされ、$\tilde{x}$ や b に無関係であることに注目されたい。つまり、この式を用いれば、丸め誤差の影響による誤差の幅を事前に見積ることができる。この公式の利点は、式の構造が簡便なことである。問題点としては、$\tilde{x}$ の計算結果が全く反映されていないことである。また、上述の「LAPACK利用手引書」はこの式では誤差を過大評価する恐れがあることを指摘している。

次に、残差ベクトルのノルムを考慮に入れた誤差評価式を示す。Stewert [6] p.196 によると、以下のように記述される。

[公式2]

$$\frac{\|x - \tilde{x}\|}{\|x\|} \leq \kappa(A)\frac{\|A\tilde{x} - b\|}{\|b\|} \tag{4}$$

この公式では、残差ベクトルのノルム $\|A\tilde{x} - b\|$ および 右辺ベクトルのノルム $\|b\|$ を計算する必要があり、やや煩雑であるが、$\tilde{x}$ の計算結果が反映される分、さきほどの公式1よりすぐれている。なお、全体の計算負荷の中で、行列条件数の計算部分が圧倒的に大きな割合を占めるので、それに比べれば、$\|A\tilde{x} - b\|$ および $\|b\|$ 等の計算負荷は、無視できるほど小さい。

3.2.2 条件数の計算

［ノルムの定義］

まず念のために行列のノルムの定義を述べる。行列のノルムの本来の定義は以下の通りである。

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} \tag{5}$$

1-ノルム、2-ノルム、∞-ノルムの場合は、以下のように具体的に表される。

$$\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^{n} |a_{ij}| \tag{6}$$

$$\|A\|_2 = \sqrt{\max |\sigma(A^* A)|} \tag{7}$$

$$\|A\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^{n} |a_{ij}| \tag{8}$$

1-ノルム、∞-ノルムについては、

$$\|A\|_\infty = \|A^T\|_1 \tag{9}$$

の関係がある。また、2-ノルムについては、行列積 A^*A の絶対値最大固有値 $\sigma(A^*A)$ を求め、次にその平方根を求める必要がある。ここで、A^* は、A の共役転置行列である。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。

$$\|A\|_2 = \max |\sigma(A)| \tag{10}$$

［条件数の定義と計算法］

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。

$$\kappa(A) = \|A\| \|A^{-1}\| \tag{11}$$

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。どのノルムの場合でも、$\|A\|$ 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。

$\|A^{-1}\|$ の計算法は、大きく分けて2通り存在する。以下に簡単に解説する。

(1) 逆行列 A^{-1} を計算してから、そのノルムを求める方法。

文字通り、まず逆行列を求め、次にそのノルムを計算する。逆行列の計算は、計算負荷が高いが、確実な方法といえる。もちろん元の行列が悪条件の場合は、逆行列が正確に求まらない恐れがあるが、それは次に述べる近似的方法でも同様である。

本ツールではこの方法を採用している。ちなみに計算量としては、行列のLU分解をするために $\frac{2}{3}n^3$ 回、次にLU分解結果を用いて逆行列を計算するのに、同じく $\frac{2}{3}n^3$ 回、したがって計 $\frac{4}{3}n^3$ の演算回数を要する。ノルムの計算等は、演算回数が n^2 のオーダーなので無視できる。

(2) 逆行列を計算せずに、$\|A^{-1}\|$ の近似値を求める方法。

逆行列を直接計算することなく、$\|A^{-1}\|$ の近似値を得る方法である。行列のノルムの本来の定義に戻り、以下の式の右辺に現れる最大値を、ベクトル x を変化させながら探索するものである。

$$\|A^{-1}\| = \max_{x \neq 0} \frac{\|A^{-1}x\|}{\|x\|} \tag{12}$$

具体的なアルゴリズムについては、Higham [5] を参照されたい。なお、ノルムの種類は、1-ノルムまたは∞-ノルムに限定される。

LAPACKに含まれるサブルーチンSGECON などでは、LU分解のついでに条件数を求めており、そこではHighamのアルゴリズムが使われている。「LAPACK利用の手引」[8] によると、この方法は上記(1)の方法に比べて計算負荷がずっと少ないという利点がある。また、ごくまれな場合を除いて、近似の度合いは概ね高いとのことである。

ちなみに計算量としては、行列のLU分解をするために$\frac{2}{3}n^3$ 回の演算が必要であるが、一旦LU分解できてしまえば、$\|A^{-1}\|$ の近似値を求めるための反復計算は n^2 のオーダーの演算量であり、無視できるので、トータルの演算量は結局$\frac{2}{3}n^3$ 程度で済む。

本ツールにおける連立一次方程式精度評価ツールでは、条件数の計算において、上記のうち、「(1) 逆行列 A^{-1} を計算してから、そのノルムを求める方法」を採用している。その理由は、一言で言えば、たとえ計算時間を要しても確実な方法をとることにしたいためである。つまり、LAPACKのサブルーチンSGECON等の場合は、本来の目的がLU分解であり、わずかな手間を追加することで近似条件数という副産物を計算しているのであるが、本ツールの場合は精度評価が主目的である以上、演算負荷をいとわずに確実な方法をとるべきと考えられるからである。

なお、今後の課題として、さまざまな行列に対して近似条件数がどの程度条件数を近似するかのケーススタディを行うために、本ツールに近似条件数計算のアルゴリズムを追加することは、無意味ではないと考えられる。

3.3 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[12] を参照されたい。

3.3.1 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

$$|\lambda - \rho| \leq ||Au - \rho u||_2 \tag{13}$$

を満たすAの固有値λが存在する。[12] ($||x||_2 \equiv \sqrt{\sum_{i=1}^{n} |x_i|^2}$.)

例3.3.1-1) 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。

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

このとき

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

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。

3.3.2 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[12)]

$$\rho - \frac{\varepsilon^2}{\overline{\lambda} - \rho} \leq \lambda \leq \rho + \frac{\varepsilon^2}{\rho - \underline{\lambda}}, \tag{14}$$

および

$$\sin\theta \leq \frac{2}{\overline{\lambda} - \underline{\lambda}} \left[\left(\rho - \frac{\underline{\lambda} + \overline{\lambda}}{2} \right)^2 + \varepsilon^2 \right]^{1/2} \tag{15}$$

が成り立つ。ここで

$$\varepsilon \equiv ||Au - \rho u||_2 \tag{16}$$

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(14)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3.3.2-1) 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、

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

が導かれる。

3.3.3 Korn-Katoの公式

Korn-Katoの公式[10]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。

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n \tag{17}$$

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]

$$\hat{\lambda}_k^{(R)} - \frac{(\hat{\lambda}_k^{(S)})^2 - (\hat{\lambda}_k^{(R)})^2}{\lambda_{k+1} - \hat{\lambda}_k^{(R)}} \leq \lambda_k \leq \hat{\lambda}_k^{(R)} + \frac{(\hat{\lambda}_k^{(S)})^2 - (\hat{\lambda}_k^{(R)})^2}{\hat{\lambda}_k^{(R)} - \lambda_{k-1}} \tag{18}$$

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

$$\hat{\lambda}_k^{(R)} = \frac{(\hat{x}_k, A\hat{x}_k)}{(\hat{x}_k, \hat{x}_k)} \tag{19}$$

$$\hat{\lambda}_k^{(S)} = \sqrt{\frac{(A\hat{x}_k, A\hat{x}_k)}{(\hat{x}_k, \hat{x}_k)}} \tag{20}$$

式(18)において、λ_{k-1}, λ_{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

$$\begin{aligned}
\underline{\lambda} &\longrightarrow \lambda_{k-1}, \\
\overline{\lambda} &\longrightarrow \lambda_{k+1}, \\
\lambda &\longrightarrow \lambda_k, \\
\rho &\longrightarrow \hat{\lambda}_k^{(R)}, \\
\varepsilon &\longrightarrow (\hat{\lambda}_k^{(S)})^2 - (\hat{\lambda}_k^{(R)})^2
\end{aligned} \tag{21}$$

としたものに対応している。

4. 配列の格納形式

本ツールでの行列の格納形式について述べる。

いわゆる通常の（並列でない）計算機のための格納仕様（以下、「逐次版」）と、分散メモリ型並列計算機のための格納仕様（以下、「並列版」）に分けて述べる。行列要素は倍精度複素数、すなわち実部8バイト、虚部8バイトを合わせて16バイトの長さを持つひとつの複素数として宣言されているものとする。

complex*16 配列名

備考　：複素数成分の行列を扱う他の方式として、実部と虚部を分けて、それぞれ倍精度実数の別々の配列に格納する仕方がある。本ツールでは、現在のところ、そのような格納方式はサポートしていない。上述のように、実部虚部を合わせてひとつの複素数として扱っている。

4.1 逐次版

(1) 複素数密行列

通常の格納方式同様である。すなわち、倍精度複素数の2次元配列を宣言し、宣言したサイズ以下の大きさの領域を使用する。

たとえば、

```
PARAMETER (LDA=7)
COMPLEX*16 AG(LDA,LDA)
```

のように宣言して、未知数の数が 6 の場合、下図の － の部分は実際にアクセスされることはない。

$$A_G = \left(\begin{array}{c|c|c|c|c|c|c} z_{11} & z_{12} & z_{13} & z_{14} & z_{15} & z_{16} & - \\ z_{21} & z_{22} & z_{23} & z_{24} & z_{25} & z_{26} & - \\ z_{31} & z_{32} & z_{33} & z_{34} & z_{35} & z_{36} & - \\ z_{41} & z_{42} & z_{43} & z_{44} & z_{45} & z_{46} & - \\ z_{51} & z_{52} & z_{53} & z_{54} & z_{55} & z_{56} & - \\ z_{61} & z_{62} & z_{63} & z_{64} & z_{65} & z_{66} & - \\ - & - & - & - & - & - & - \end{array}\right)$$

(2) エルミート行列

一般の複素数密行列と同じ格納方式とする。すなわち、下三角部分または上三角部分のみを格納するのではなく、全部を格納する。

エルミート性により、

$$z_{ji} = \bar{z}_{ij}$$

が成り立つので、以下のように格納されている必要がある(ここで、$\bar{z}$ は、複素数zの共役複素数であることを示す)。

$$H_G = \left(\begin{array}{c|c|c|c|c|c|c} z_{11} & z_{12} & z_{13} & z_{14} & z_{15} & z_{16} & - \\ \bar{z}_{12} & z_{22} & z_{23} & z_{24} & z_{25} & z_{26} & - \\ \bar{z}_{13} & \bar{z}_{23} & z_{33} & z_{34} & z_{35} & z_{36} & - \\ \bar{z}_{14} & \bar{z}_{24} & \bar{z}_{34} & z_{44} & z_{45} & z_{46} & - \\ \bar{z}_{15} & \bar{z}_{25} & \bar{z}_{35} & \bar{z}_{45} & z_{55} & z_{56} & - \\ \bar{z}_{16} & \bar{z}_{26} & \bar{z}_{36} & \bar{z}_{46} & \bar{z}_{56} & z_{66} & - \\ - & - & - & - & - & - & - \end{array}\right)$$

4.2 並列版

(1) 複素数密行列

元の行列を指定された列幅で列サイクリック分割して格納する。

以下に例を挙げて説明する。たとえば、未知数が7個の問題を解くとして、単一メモリイメージでの元の行列（以下、「グローバル行列」）が7×7 であるとし、サイクリック分割の列幅を1として、PE_0、PE_1、PE_2の3個のプロセッサエレメント(以下、「PE」）で処理するとする。グローバル行列においては、以下のように分割される。

$$\text{元の行列} = \left(\begin{array}{c|c|c|c|c|c|c} z_{11} & z_{12} & z_{13} & z_{14} & z_{15} & z_{16} & z_{17} \\ z_{21} & z_{22} & z_{23} & z_{24} & z_{25} & z_{26} & z_{27} \\ z_{31} & z_{32} & z_{33} & z_{34} & z_{35} & z_{36} & z_{37} \\ z_{41} & z_{42} & z_{43} & z_{44} & z_{45} & z_{46} & z_{47} \\ z_{51} & z_{52} & z_{53} & z_{54} & z_{55} & z_{56} & z_{57} \\ z_{61} & z_{62} & z_{63} & z_{64} & z_{65} & z_{66} & z_{67} \\ z_{71} & z_{72} & z_{73} & z_{74} & z_{75} & z_{76} & z_{77} \end{array}\right)$$

$$\text{ランク番号} = \left(\begin{array}{c|c|c|c|c|c|c} 0 & 1 & 2 & 0 & 1 & 2 & 0 \end{array}\right)$$

グローバル配列の中で、各PEが自分の処理担当範囲だけを部分的に格納する配列をローカル配列という。例えば、ローカル配列が以下のように宣言されているとする。

```
COMPLEX*16 AL(7,3)
```

このとき、ローカル配列の内容は、以下のようになる。

$$AL_{pe0}=\begin{pmatrix} z_{11} & z_{14} & z_{17} \\ z_{21} & z_{24} & z_{27} \\ z_{31} & z_{34} & z_{37} \\ z_{41} & z_{44} & z_{47} \\ z_{51} & z_{54} & z_{57} \\ z_{61} & z_{64} & z_{67} \\ z_{71} & z_{74} & z_{77} \end{pmatrix},\ AL_{pe1}=\begin{pmatrix} z_{12} & z_{15} & - \\ z_{22} & z_{25} & - \\ z_{32} & z_{35} & - \\ z_{42} & z_{45} & - \\ z_{52} & z_{55} & - \\ z_{62} & z_{65} & - \\ z_{72} & z_{75} & - \end{pmatrix},\ AL_{pe2}=\begin{pmatrix} z_{13} & z_{16} & - \\ z_{23} & z_{26} & - \\ z_{33} & z_{36} & - \\ z_{43} & z_{46} & - \\ z_{53} & z_{56} & - \\ z_{63} & z_{66} & - \\ z_{73} & z_{76} & - \end{pmatrix}$$

(2) エルミート行列

上記「(1) 複素数密行列」と同様の格納方式とする。エルミート行列ではあるが、エルミート性を利用するような特別な格納方法はとらない。つまり、下三角部分または上三角部分のみを格納するのではなく、全部を格納する。

留意点として、グローバル行列の状態でエルミート性により、

$$z_{ji}=\bar{z}_{ij}$$

が成り立たなくてはならない。上記「(1) 複素数密行列」と同様の宣言をされた配列を用いて例で示すと、以下のようになる (複素共役の位置に注目)。

$$\text{元の行列}\quad=\left(\begin{array}{c|c|c|c|c|c|c} z_{11} & z_{12} & z_{13} & z_{14} & z_{15} & z_{16} & z_{17} \\ \bar{z}_{12} & z_{22} & z_{23} & z_{24} & z_{25} & z_{26} & z_{27} \\ \bar{z}_{13} & \bar{z}_{23} & z_{33} & z_{34} & z_{35} & z_{36} & z_{37} \\ \bar{z}_{14} & \bar{z}_{24} & \bar{z}_{34} & z_{44} & z_{45} & z_{46} & z_{47} \\ \bar{z}_{15} & \bar{z}_{25} & \bar{z}_{35} & \bar{z}_{45} & z_{55} & z_{56} & z_{57} \\ \bar{z}_{16} & \bar{z}_{26} & \bar{z}_{36} & \bar{z}_{46} & \bar{z}_{56} & z_{66} & z_{67} \\ \bar{z}_{17} & \bar{z}_{27} & \bar{z}_{37} & \bar{z}_{47} & \bar{z}_{57} & \bar{z}_{67} & z_{77} \end{array}\right)$$

$$\text{ランク番号}=\left(\begin{array}{c|c|c|c|c|c|c} 0 & 1 & 2 & 0 & 1 & 2 & 0 \end{array}\right)$$

これを、たとえば、

complex*16 AL(7,3)

のように宣言されたローカル配列に格納する場合、以下のようになる。

$$AL_{pe0}=\begin{pmatrix} z_{11} & z_{14} & z_{17} \\ \bar{z}_{12} & z_{24} & z_{27} \\ \bar{z}_{13} & z_{34} & z_{37} \\ \bar{z}_{14} & z_{44} & z_{47} \\ \bar{z}_{15} & \bar{z}_{45} & z_{57} \\ \bar{z}_{16} & \bar{z}_{46} & z_{67} \\ \bar{z}_{17} & \bar{z}_{47} & z_{77} \end{pmatrix},\ AL_{pe1}=\begin{pmatrix} z_{12} & z_{15} & - \\ z_{22} & z_{25} & - \\ \bar{z}_{23} & z_{35} & - \\ \bar{z}_{24} & z_{45} & - \\ \bar{z}_{25} & z_{55} & - \\ \bar{z}_{26} & \bar{z}_{56} & - \\ \bar{z}_{27} & \bar{z}_{57} & - \end{pmatrix},\ AL_{pe2}=\begin{pmatrix} z_{13} & z_{16} & - \\ z_{23} & z_{26} & - \\ z_{33} & z_{36} & - \\ \bar{z}_{34} & z_{46} & - \\ \bar{z}_{35} & z_{56} & - \\ \bar{z}_{36} & z_{66} & - \\ \bar{z}_{37} & \bar{z}_{67} & - \end{pmatrix}$$

5. プログラム利用方法

5.1 複素数連立一次方程式の誤差評価(逐次版) szgees

5.1.1 機能

複素数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。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)。

5.1.2 コーリングシーケンス

```
      call szgees
           (EPSMCH,
     &      LDA    ,NN     ,AA     ,XX     ,BB     ,
     &      IWORK  ,DWORK  ,ZWORK  ,
     &      RNRM1  ,RNRMI  ,BNRM1  ,BNRMI  ,
     &      ANRM1  ,ANRMI  ,AIVNR1,AIVNRI,RCOND1,RCONDI,
     &      ERBDA1,ERBDAI,ERBDB1,ERBDBI,IRTN          )
```

5.1.3 引数

用法

in　入力用。値は壊されない。

out　出力用。

inout　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。

work　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)

C　文字型(character * (*))

変数／配列(次元)	I/O	型	意味
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()などにより求めることができる。
LDA	in	I	行列格納用配列ALの一次元目の宣言サイズ。
NN	in	I	連立一次方程式の次元を示す。
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。
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
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($=\|A\|_\infty$)。
AINVR1	out	D	逆行列 A^{-1} の 1−ノルム($=\|A^{-1}\|_1$)。
AINVRI	out	D	逆行列 A^{-1} の ∞−ノルム($=\|A^{-1}\|_\infty$)。
RCOND1	out	D	1−ノルムで計算した条件数($\kappa_1(A)=\|A\|_1\|A^{-1}\|_1$)。
RCONDI	out	D	∞−ノルムで計算した条件数($\kappa_\infty(A)=\|A\|_\infty\|A^{-1}\|_\infty$)。
ERBDA1	out	D	公式1、1−ノルムで計算した誤差限界($=\kappa_1(A)\ \epsilon$)。
ERBDAI	out	D	公式1、∞−ノルムで計算した誤差限界($=\kappa_\infty(A)\ \epsilon$)。
ERBDB1	out	D	公式2、1−ノルムで計算した誤差限界($=\kappa_1(A)\frac{\|Ax-b\|_1}{\|b\|_1}$)。
ERBDBI	out	D	公式2、∞−ノルムで計算した誤差限界($=\kappa_\infty(A)\frac{\|Ax-b\|_\infty}{\|b\|_\infty}$)。
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　特になし。

5.2 複素数連立一次方程式の誤差評価(並列版) pzgees

5.2.1 機能

複素数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。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。

5.2.2 コーリングシーケンス

```
      CALL PZGEES
           (nprocs,myrank,
     &      whohas,nclmL ,mapG2L,mapL2G,
     &      EPSMCH,
     &      LDA   ,ncmaxL,nnG   ,aL    ,XX    ,BB    ,
     &      nzcc  ,nzwmax,IWORK ,DWORK ,ZWORK ,
     &      RNRM1 ,RNRMI ,BNRM1 ,BNRMI ,
     &      ANRM1 ,ANRMI ,AIVNR1,AIVNRI,RCOND1,RCONDI,
     &      ERBDA1,ERBDAI,ERBDB1,ERBDBI,IRTN  )
```

5.2.3 引数

用法

in　入力用。値は壊されない。

out　出力用。

inout　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。

work　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)

C　文字型(character * (*))

変数／配列(次元)	I/O	型	意味
nprocs	in	I	プロセスの数
myrank	in	I	自プロセスのランク
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($=0,1,\cdots,nprocs-1$)がwhohas(j)に格納されているものとする。whohas の内容は、全PEについて共通である。
nclmL(0:nprocs-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($=0,1,\cdots$,nprocs -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について共通である。
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。
nnG	in	I	連立一次方程式の次元を示す。$nnG \leq LDA$
aL(LDA,ncmaxL)	i/o	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た形で与える。 LU分解されたものが戻値となる。
XX(LDA)	in	I	$Ax=b$ の解 x。
BB(LDA)	in	I	$Ax=b$ の右辺ベクトル b。
nzcc	in	I	通信時に使用する作業領域の長さ。通常は nnG を指定。
nzwmax	in	I	逆行列計算時に一度に右辺に与える単位行列の本数。

変数／配列(次元)	I/O	型	意味
IWORK(LDA)	w	I	作業用配列
DWORK(LDA*2)	w	D	作業用配列
XWORK(LDA*nzwmax +(nzwmax+1)*2)	w	Z	作業用配列
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$(=\|A\|_\infty)$。
AINVR1	out	D	逆行列A^{-1}の 1−ノルム$(=\|A^{-1}\|_1)$。
AINVRI	out	D	逆行列A^{-1}の ∞−ノルム$(=\|A^{-1}\|_\infty)$。
RCOND1	out	D	1−ノルムで計算した条件数$(\kappa_1(A)=\|A\|_1\|A^{-1}\|_1)$。
RCONDI	out	D	∞−ノルムで計算した条件数$(\kappa_\infty(A)=\|A\|_\infty\|A^{-1}\|_\infty)$。
ERBDA1	out	D	公式1、1−ノルムで計算した誤差限界$(=\kappa_1(A)\epsilon)$。
ERBDAI	out	D	公式1、∞−ノルムで計算した誤差限界$(=\kappa_\infty(A)\epsilon)$。
ERBDB1	out	D	公式2、1−ノルムで計算した誤差限界$(=\kappa_1(A)\frac{\|Ax-b\|_1}{\|b\|_1})$。
ERBDBI	out	D	公式2、∞−ノルムで計算した誤差限界$(=\kappa_\infty(A)\frac{\|Ax-b\|_\infty}{\|b\|_\infty})$。
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。

5.3 エルミート密行列の固有値誤差評価(逐次版) sqhe3r

5.3.1 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)。

5.3.2 コーリングシーケンス

```
      CALL SQHE3R
     &     (LDA    ,NN     ,AA     ,EVAL   ,EVEC   ,
     &      ZWORK ,
     &      RAY    ,EVALLW,EVALHI,RN     ,THT    ,IRTNCD)
```

5.3.3 引数

用法

in　入力用。値は壊されない。

out　出力用。

inout　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。

work　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)

C　文字型(character * (*))

変数／配列(次元)	I/O	型	意味
LDA	in	I	行列格納用配列AAの宣言サイズ。
NN	in	I	行列の次数。
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。
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(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$\Delta\lambda^+$ 。各モードにつき、 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 $\sin\theta \leq$ THT.
IRTNCD	out	I	エラーコード。 ＝ 0：正常終了。 ＝ −1：引数誤り。 ＝ 2：固有値が昇順ソートされていない。 ＝ 8：特異行列。

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。

5.4 エルミート密行列の固有値誤差評価(並列版) plhe3r

5.4.1 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。

5.4.2 コーリングシーケンス

```
      CALL PLHE3R
     &     (nprocs,myrank,LDA,ncmaxL,NDIM,iblock,llrank,lrank2,
     &      aL,evecL,EIG,RAY,ERRM,ERRP,RN,THT,ierr,EPS,IDBG,iopt,
     &      WK,IUP,IDW,WK1,WK2,evecLw)
```

5.4.3 引数

用法

in　入力用。値は壊されない。

out　出力用。

inout　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。

work　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)

C　文字型(character * (*))

変数／配列(次元)	I/O	型	意味
nprocs	in	I	プロセスの数
myrank	in	I	自プロセスのランク
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。
NDIM	in	I	行列の次数。
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:nprocs-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。データ分割や配列の縮小は行なわない。
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 $\Delta\lambda^+$ 。 各モードにつき、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.
ierr	out	I	エラーコード。 =0: 正常終了。
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

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。

参考文献

1) K. E. Atkinson, *An Introduction to Numerical Analysis, 2nd ed.*, John Wiley & Sons.

2) T. F. Coleman and C. Van Loan, *Handbook for Matrix Computations.*, SIAM. 1988.

3) G. H. Golub and C. F. Van Loan, *Matrix Computations.*, Johns Hopkins Univ. 1989.

4) W. Gropp, E. Lusk and A. Skjellum, *Using MPI.*, MIT Press. 1994.

5) 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.

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

7) O. Tatebe, *LU Decomposition on Distributed Memory Machines*, in IPSJ SIG Notes, 95-HPC-57,pp.55-60,August 1995. SWoPP'95 Beppu.

8) 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.

9) 小国力 他、行列計算ソフトウェア－WS,スーパーコン,並列計算機－、丸善 1991.

10) 戸川隼人、マトリクスの数値計算、オーム社 1971.

11) 渡部力 他、Fortran77による数値計算ソフトウェア、丸善 1989.

12) 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.

13) 固有値誤差評価ライブラリ利用手引書、日本原子力研究所 計算科学技術推進センター 1996.

14) 実密対称行列の固有値問題並列解法ルーチン、日本原子力研究所 計算科学技術推進センター 1996.

記号

$\|a\|$	スカラー値 a の絶対値($= \max(a, -a)$)。
A^T	行列 A の転置行列。
A^*	行列 A の共役転置行列。A^H という書き方の文献もある。
$\|A\|$	行列 A のノルム($= \max_{x \neq 0} \frac{\|Ax\|}{\|x\|}$)
$\|A\|_1$	行列 A の1-ノルム($= \max_{1 \leq j \leq n} \sum_{i=1}^{n} \|a_{ij}\|$)
$\|A\|_2$	行列 A の2-ノルム($= \sqrt{\max \|\sigma(A^*A)\|}$)
$\|A\|_\infty$	行列 A の∞-ノルム($= \max_{1 \leq i \leq n} \sum_{j=1}^{n} \|a_{ij}\|$)
$\tilde{x}$	$Ax = b$ の数値解。
$\|x\|$	ベクトル x のノルム。
$\|x\|_2$	ベクトル x の2-ノルム($= \sqrt{\sum_{i=1}^{n} \|x_i\|^2}$)
(x, y)	ベクトル x、y の内積($= x^*y$ または x^Hy)
x^T	ベクトル x の転置ベクトル。
x^*	ベクトル x の共役転置ベクトル。x^H という書き方の文献もある。
$\bar{z}$	複素数 z の共役複素数。
$\frac{\delta}{z}$	相対後退誤差。
ϵ	マシンイプシロン（計算機精度)。
λ	固有値。
ρ	Rayleigh 商 ($= (u, Au)/(u, u)$)
$\sigma(A)$	行列 A の固有値のうち、絶対値最大値。
$\kappa(A)$	行列 A の条件数($= \|A\|\|A^{-1}\|$)