

まえがき

この小冊子は、スピンの大きさ S が 1 の量子スピン系のハミルトニアンを数値的に対角化するためのプログラムパッケージ KOBEPACK/ 1 Version 1.0 の利用説明書である。よく知られているように、 $S=1/2$ の場合のハミルトニアンを対角化するための優れたプログラムパッケージとして、田口善弘氏(東工大理)と西森秀稔氏(東工大理)とによる TITPACK Version 1^{1,2)} および西森秀稔氏による TITPACK Version 2^{3,4)} がある。KOBEPACK/ 1 Version 1.0 の開発に当っては、両氏のご厚意により、TITPACK Version 1 および Version 2 で用いられている多くの技法(例えば、中規模行列用ルーチンにおける効率的な行列要素格納法)を参考にさせていただき、配列 IPAIR やサブルーチン副プログラム HSHLDR, VEC3, VEC12, BISEC 等をそのまま使わせていただいた。さらに、本利用説明書のレイアウトも、西森秀稔氏による TITPACK Version 2 の大変分かりやすい操作説明書³⁾ に沿った形になっており、また、一部で同じ文章を用いさせていただいた。西森秀稔、田口善弘の両氏に厚く感謝申し上げる。

KOBEPACK/ 1 Version 1.0 のコーディングで我々が用いた新しい方法は、考える系を 2 つの部分に分割する方法(部分空間コーディング法)である。⁵⁾ 後に詳しく述べるように、この方法を用いることにより、大幅に計算時間を短縮させ、かつ、大幅に記憶容量を節減させることが出来、實際上、TITPAK における大規模行列用ルーチンに相当するルーチンは必要でなくなっている。

KOBEPACK/ 1 Version 1.0 の著作権は、TITPACK Version 2 から流用させていただいた部分(サブルーチン副プログラム DIAGNLZ, HSHLDR, VEC3, VEC12, BISEC)を除いて、筆者等に属する。学術研究上の目的で、個人的に利用することについては特に制限はない。ただし、KOBEPACK/ 1 Version 1.0 をそのままあるいは改変して利用することによって得られた成果を発表する際には、著作者名とプログラム名を明記して下さるようお願いする。

KOBEPACK/ 1 Version 1.0 の開発および本利用説明書の作成は、文部省科学研究費重点領域研究「計算物理学 — 物性研究における新展開 —」の助成によりなされた。深く感謝の意を表す。また、開発者および著者の一人(鍋木誠)は富士通株式会社からの研究助成に感謝する。

1992 年 12 月

利根川 孝, 鍋木 誠, 西野友年

目次

§1. 概説	1
1.1 KOBEPACK/1 Version 1.0 に含まれる副プログラムとそれぞれの機能	1
1.2 ルーチン群の選択と各ルーチンの特徴	3
1.3 小規模行列用ルーチンの概要とフローチャート	3
1.4 中規模以上行列用ルーチンの概要とフローチャート	5
1.5 他機種への移植	8
§2. 部分空間コーディング法および各ルーチンにおけるプログラム技法	12
2.1 部分空間コーディング法における2次元サーチ法	12
2.2 部分空間コーディング法における行列要素の計算方法	14
2.2.1 小規模行列用ルーチンの場合	16
2.2.2 中規模以上行列用ルーチンの場合	17
§3. 使用法	23
3.1 小規模行列用ルーチン	23
3.2 中規模以上行列用ルーチン	35
3.3 物理量計算用ルーチン	51
3.4 ユーザが用いるその他の副プログラム	58
§4. エラーメッセージ	61
§5. 提供方法など	67
参考文献	69
Appendix A. ユーザが直接使わない下位ルーチンの説明	70
A.1 小規模行列用ルーチンの下位ルーチン	70
A.1.1 CRESZ の下位ルーチン	70
A.1.2 SMALL の下位ルーチン	71
A.1.3 CRELMS の下位ルーチン	73
A.1.4 CHECKS の下位ルーチン	75
A.2 中規模以上行列用ルーチンの下位ルーチン	76
A.2.1 CRESZ の下位ルーチン	76
A.2.2 ELMM の下位ルーチン	76
A.2.3 LNCZM の下位ルーチン	85
A.2.4 LNCZMINV の下位ルーチン	104
A.2.5 CHECKM の下位ルーチン	107
A.3 物理量計算用ルーチンの下位ルーチン	108
A.4 ユーザが用いるその他の副プログラムの下位ルーチン	110
A.4.1 FIND0 の下位ルーチン	110
A.4.2 CHKWV の下位ルーチン	110
A.5 多くのルーチンに共通な下位ルーチン	111

§1. 概説

KOBEPACK/ 1 Version 1.0 は次のハミルトニアンで記述される $S=1$ の量子スピン系を取り扱う.

$$\mathcal{H} = \sum_{\langle \ell, \ell' \rangle} \left\{ J_{\ell, \ell'}^x (S_\ell^x S_{\ell'}^x + S_\ell^y S_{\ell'}^y) + J_{\ell, \ell'}^z S_\ell^z S_{\ell'}^z + K_{\ell, \ell'} (\vec{S}_\ell \cdot \vec{S}_{\ell'})^2 \right\} + \sum_{\ell} \left\{ D_\ell (S_\ell^z)^2 - H_\ell S_\ell^z \right\}. \quad (1)$$

ここで, $\langle \ell, \ell' \rangle$ は相互作用しているスピンが占めるサイト対を表す. スピン数が 2 から 2 4 までの有限系について, このハミルトニアンの対角化を $S_{\text{total}}^z = \sum_{\ell} S_\ell^z$ の固有値 m_{total} を与えた部分空間内で行うことにより, 各部分空間内での最低エネルギー状態, 第 1, 第 2, 第 3 励起状態のエネルギー固有値, 固有ベクトルを求める. 格子の形, 相互作用が及ぶ範囲, 境界条件等は, $\langle \ell, \ell' \rangle$ を適当に与えることによって任意に設定できる. なお, 次のような物理量を計算するサブルーチン副プログラムも用意されている.

スピン対相関関数 :

$$\omega^\alpha(\ell, \ell') = \langle S_\ell^\alpha S_{\ell'}^\alpha \rangle, \quad (2)$$

$$\Delta\omega^z(\ell, \ell') = \langle S_\ell^z S_{\ell'}^z \rangle - s_\ell^z s_{\ell'}^z \quad (s_\ell^z = \langle S_\ell^z \rangle); \quad (3)$$

ダイマー相関関数 :^{6,7)}

$$\tau(\ell, \ell') = 4 \langle (S_\ell^x S_{\ell+1}^x + S_\ell^y S_{\ell+1}^y) (S_{\ell'}^x S_{\ell'+1}^x + S_{\ell'}^y S_{\ell'+1}^y) \rangle, \quad (4)$$

$$\Delta\tau(\ell, \ell') = 4 \langle (S_\ell^x S_{\ell+1}^x + S_\ell^y S_{\ell+1}^y) (S_{\ell'}^x S_{\ell'+1}^x + S_{\ell'}^y S_{\ell'+1}^y) \rangle - t_\ell t_{\ell'} \quad (t_\ell = 2 \langle S_\ell^x S_{\ell+1}^x + S_\ell^y S_{\ell+1}^y \rangle); \quad (5)$$

ストリング相関関数 :⁸⁾

$$\sigma^\alpha(\ell, \ell') = - \left\langle S_\ell^\alpha \exp(i\pi \sum_{n=\ell+1}^{\ell'-1} S_n^\alpha) S_{\ell'}^\alpha \right\rangle. \quad (6)$$

ここで, $\alpha=x, z$ であり, $\langle \dots \rangle$ は求められた固有ベクトルでの期待値を示す.

KOBEPACK/ 1 Version 1.0 は FORTRAN 77 の標準仕様に従って書かれている. ただし, 8 文字の副プログラム名および変数名を用いている箇所があるので, これがサポートされていない場合には, その部分を書き換える必要がある. KOBEPACK/ 1 Version 1.0 には, ACOS および SX 用 (科学技術計算ライブラリ ASL を使用) の版とワークステーション SUN 用の版とが準備されている. これらを他機種へ移植する際に変更すべき点については 1.5 節を参照されたい.

1.1 KOBEPACK/ 1 Version 1.0 に含まれる副プログラムとそれぞれの機能 (関数副プログラムについてのみ副プログラム名の右肩に * 印をつける. その他はすべてサブルーチン副プログラムである.)

小規模行列用ルーチン (すべての行列要素をあらかじめ計算しておいてから対角化する.)

副プログラム名	およその機能
SMALL	行列要素の計算, および Householder 法, バイセクション法, 逆反復法とによるエネルギー固有値, 固有ベクトルの計算.
CRELMS	行列要素の計算.
CHECKS	エネルギー固有値, 固有ベクトルの精度チェック.

中規模以上行列用ルーチン (0 でない行列要素をあらかじめ計算しておいてから対角化する. ただし, 主記憶に 0 でない行列要素を格納するための領域が不足する場合には, 不足する分についてのみ毎回計算しながら対角化する.)

副プログラム名	およその機能
ELMM	行列要素の計算.
LNCZM	Lanczös 法, バイセクション法によるエネルギー固有値の計算.
LNCZMINV	Lanczös 法あるいは Lanczös 法と CG 法とを組合せた逆反復法による固有ベクトルの計算.
CHECKM	エネルギー固有値, 固有ベクトルの精度チェック.

ユーザがメインルーチンで用いるその他の副プログラム

副プログラム名	およその機能
NCONF*	対角化する行列の次元 IDIM の計算.
CRESZ	スピン配列の生成と Storage Table, Lookup Table の作成.
FIND0	$m_1 = m_2 = \dots = m_{NS} = 0$ であるスピン配列の番号の計算 (m_ℓ は S_ℓ^z の固有値, NS は系に含まれる全スピン数).
CHKWV	固有ベクトルがもつ並進対称性を調べるためのデータの作成.
DISPNF	整数の +, 0, - による 3 進法表示.
CRFNST1	$s_\ell^z, \omega^x(l, l'), \Delta\omega^z(l, l'), t_\ell, \Delta\tau(l, l')$ の計算.
CRFNS1	$s_\ell^z, \omega^x(l, l'), \Delta\omega^z(l, l')$ の計算.
CRFNT1	$t_\ell, \Delta\tau(l, l')$ の計算.
AVRGS1	s_ℓ^z の計算.
AVRGT1	t_ℓ の計算.
CRFNST2	$\omega^x(l, l'), s_\ell^z$ を与えた上での $\Delta\omega^z(l, l')$, および t_ℓ を与えた上での $\Delta\tau(l, l')$ の計算.
CRFNS2	$\omega^x(l, l')$, および s_ℓ^z を与えた上での $\Delta\omega^z(l, l')$ の計算.
CRFNT2	t_ℓ を与えた上での $\Delta\tau(l, l')$ の計算.
AVRGSISJ	$\omega^x(l, l'), \omega^z(l, l')$ の計算.
AVRGTITJ	$\tau(l, l')$ の計算.
STRODRZ	$\sigma^z(l, l')$ の計算.
STRODRX	$\sigma^x(l, l')$ の計算.

以上の各副プログラム内で呼び出されている多くの下位副プログラムがあるが, 通常ユーザはこれらを意識しなくてよい. これらに興味のあるユーザは 1.3 節, 1.4 節 で図示するフローチャートおよび Appendix A を参照されたい.

1.2 ルーチン群の選択と各ルーチンの特徴

対角化する行列の次元 IDIM が小さいと Lanczös 法が安定して正しい結果を与えないことが多い。一方、IDIM が大きくなると、低い方の数個のエネルギー固有値およびそれらに対応する固有ベクトルを求めるという目的のためには、Lanczös 法の方が適している。TITPACK Version 2 の操作説明書³⁾にあるように、一般的に云って、IDIM がおよそ 100 以下では小規模行列用ルーチンを用い、それ以上では中規模以上行列用ルーチンを用いるのがよい。

小規模行列用ルーチンを用いると、すべてのエネルギー固有値および固有ベクトルを完全に求めることが出来る。

中規模以上行列用ルーチンでは、0 でない行列要素をあらかじめ計算しておいてから対角化する。ただし、主記憶に 0 でない行列要素を格納するための領域が不足する場合には、不足する分についてのみ毎回計算しながら対角化する。計算に要する時間の面から云えば、当然、すべての 0 でない行列要素を格納するための領域を確保しておくのがよい。また、本ルーチンでは 2 種類の方法で固有ベクトルの計算が行えるようになっている。一方の方法では、計算に必要な記憶容量と時間が少なく済むが、得られた結果の精度があまりよくなく (TITPACK Version 2 の操作説明書³⁾によれば、6 桁程度あるいはそれ以下)、他方の方法では、より多くの記憶容量と計算時間を必要とするが、より良い精度 (TITPACK Version 2 の操作説明書³⁾によれば、8 ないし 10 桁程度)の結果が得られる。より詳しくは、LNCZMINV (3.2 節 [5]) の [使用上の注意] の項を参照されたい。

1.3 小規模行列用ルーチンの概要とフローチャート

次元 IDIM が小さな行列に対して、すべての行列要素をあらかじめ計算し、それらを主記憶に格納しておいてから、そのデータを用いて対角化を行う。処理の流れの概略は次の通りである。

- i) 行列の次元 IDIM を計算する。
- ii) スピン配列を生成し、Storage Table, Lookup Table を作成する。
- iii) 行列要素の計算を行う。さらに、
 - iii a) ACOS および SX 用の版では、エネルギー固有値のみを計算する場合には、それを科学技術計算ライブラリ ASL 中のサブルーチン副プログラム DCSMSN を用いて求め (この場合の処理はここで終了する)、エネルギー固有値および固有ベクトルの両方を計算する場合には、それらを DCSMSS を用いて求める。
 - iii b) SUN 用の版では、先ず、Householder 法により行列の 3 重対角化を行う。次に、エネルギー固有値のみを計算する場合には、3 重対角行列の固有値をパイセクション法で求める (3 重対角行列の固有値は元の行列の固有値に等しく、この場合の処理はここで終了する)。さらに、固有ベクトルをも計算する場合には、LU 分解を用いた逆反復法で 3 重対角行列の固有ベクトルを求め、それを 3 重対角化する前の基底での表示に戻すことにより、元の行列の固有ベクトルを求める。IDIM は 3 以上でなければならない。ここで使用する副プログラムは、TITPACK Version 2 でのものをそのまま利用させていただいている。
- iv) 次の処理の準備として、もう一度行列要素の計算を行う。

- v) エネルギー固有値, 固有ベクトルの精度チェックを行う.
- vi) 相関関数などの物理量の計算を行う.

1.4 中規模以上行列用ルーチンの概要とフローチャート

次元 IDIM が 100 程度より大きな行列に対して、0 でない行列要素の位置と値をあらかじめ計算し、それらを主記憶に格納しておいてから、そのデータを用いて対角化を行う。ただし、主記憶にこれらを格納するための領域が不足する場合には、不足する分についてのみ毎回計算しながら対角化を行う。処理の流れの概略は次の通りである。

- i) 行列の次元 IDIM を計算する。
- ii) スピン配列を生成し、Storage Table, Lookup Table を作成する。
- iii) 0 でない行列要素の位置と値を、主記憶内にあらかじめ設定した格納領域の範囲内で計算する。
- iv) 格納領域からはみ出る 0 でない行列要素がある場合には、その位置と値を計算しながら、次の処理を行う。
 - iv a) ACOS および SX 用の版では、先ず Lanczös 法により行列の 3 重対角化を行う。次に、エネルギー固有値のみを計算する場合には、科学技術計算ライブラリ ASL 中のサブルーチン副プログラム DCSTSN を用いて、3 重対角行列の固有値を求め (3 重対角行列の固有値は元の行列の固有値に等しく、この場合の処理はここで終了する)、エネルギー固有値および固有ベクトルの両方を計算する場合には、DCSTSS を用いて 3 重対角行列の固有値、固有ベクトルを求める。
 - iv b) SUN 用の版では、先ず、Lanczös 法により行列の 3 重対角化を行う。次に、エネルギー固有値のみを計算する場合には、3 重対角行列の固有値をバイセクション法で求める (3 重対角行列の固有値は元の行列の固有値に等しく、この場合の処理はここで終了する)。さらに、固有ベクトルをも計算する場合には、次の処理の準備として、LU 分解を用いた逆反復法で 3 重対角行列の固有ベクトルを求める。3 重対角行列の固有値、固有ベクトルの計算に使用する副プログラムは TITPACK Version 2 でのものをそのまま利用させていただいている。
- v) 前の処理で求めた 3 重対角行列の固有ベクトルより、Lanczös 法あるいは Lanczös 法と CG 法とを組合せた逆反復法で、元の行列の固有ベクトルを求める。どちらの方法によるかで、必要な記憶容量と得られた結果の精度が異なる。
- vi) エネルギー固有値、固有ベクトルの精度チェックを行う。
- vii) 相関関数などの物理量の計算を行う。

1.5 他機種への移植

ACOS および SX 用の版では, 科学技術計算ライブラリ ASL 中のサブルーチン副プログラム DCSMSS (実対称行列の固有値・固有ベクトル), DCSMSN (実対称行列の固有値), DCSTSS (実対称 3 重対角行列の固有値・固有ベクトル), DCSTSN (実対称 3 重対角行列の固有値) および固有な組込み関数副プログラム RAND (一様乱数) を用い, また, SX に固有なベクトル化指示オプション *VDIR …… を用いている. 他的大型汎用機またはスーパーコンピュータへの移植に際しては, これらの部分を書換えなければならない. 参考のため, これらの副プログラムの概要を記す.

CALL DCSMSS(A, LNA, N, EPS, E, M, VE, LNV, ISW, IW1, W1, IERR)

実対称行列の固有値を, ハウスホルダー法, バイセクション法により, 大きい方から M 個, または小さい方から M 個求め, それらに対応する固有ベクトルを, 逆反復法により求める.

[引数]

A(LNA, N) (入力; 8 バイト実数)

対角化する行列の行列要素. 第 i 行第 j 列の行列要素を A(i, j) に与える. DCSMSS を実行の後, 配列 A の値は保存されない.

LNA (入力; 4 バイト整数)

配列 A の整合寸法. $LNA \geq N$ でなければならない.

N (入力; 4 バイト整数)

対角化する行列の次元 ($N \geq 2$).

EPS (入力; 8 バイト実数)

固有値の収束判定のための絶対誤差の上限を与えるパラメータ. $EPS \geq 0.0$ のときには入力値が与えられ, $EPS < 0.0$ のときには内部で自動的に値が設定される.

E(M) (出力; 8 バイト実数)

E(1), E(2), …, E(M) の順に, 大きい方から M 個, または小さい方から M 個の固有値が返される.

M (入力; 4 バイト整数)

求めたい固有値, 固有ベクトルの個数. $1 \leq M \leq N$ でなければならない.

VE(LNV, M) 出力; (8 バイト実数)

M 個の固有ベクトルが返される. E(i) に対応するものが VE(1, i) から VE(N, i) までという順に並べられている.

LNV (入力; 4 バイト整数)

配列 VE の整合寸法. $LNV \geq N$ でなければならない.

ISW (入力; 4 バイト整数)

処理スイッチ. $ISW \geq 0$ のとき, 大きい方から M 個の固有値およびそれらに対応する固有ベクトルが求められ, $ISW < 0$ のとき, 小さい方から M 個の固有値およびそれらに対応する固有ベクトルが求められる.

IW1(M) (作業領域; 4 バイト整数)

W1(8*N) (作業領域; 8 バイト実数)

IERR (出力; 4 バイト整数)

エラーインディケータ. 正常終了の場合 0 が返される. 異常終了の場合に返される IERR の値の詳細については, 日本電気株式会社発行のマニュアル⁹⁾を参照されたい.

CALL DCSMSN(A, LNA, N, EPS, E, M, ISW, W1, IERR)

実対称行列の固有値を, ハウスホルダー法, バイセクション法により, 大きい方から M 個, または小さい方から M 個求める.

[引数]

A(LNA, N) (入力; 8 バイト実数)

対角化する行列の行列要素. 第 i 行第 j 列の行列要素を A(i, j) に与える. DCSMSN を実行の後, 配列 A の値は保存されない.

LNA (入力; 4 バイト整数)

配列 A の整合寸法. $LNA \geq N$ でなければならない.

N (入力; 4 バイト整数)

対角化する行列の次元 ($N \geq 2$).

EPS (入力; 8 バイト実数)

固有値の収束判定のための絶対誤差の上限を与えるパラメータ. $EPS \geq 0.0$ のときには入力値が与えられ, $EPS < 0.0$ のときには内部で自動的に値が設定される.

E(M) (出力; 8 バイト実数)

E(1), E(2), ..., E(M) の順に, 大きい方から M 個, または小さい方から M 個の固有値が返される.

M (入力; 4 バイト整数)

求めたい固有値の個数. $1 \leq M \leq N$ でなければならない.

ISW (入力; 4 バイト整数)

処理スイッチ. $ISW \geq 0$ のとき, 大きい方から M 個の固有値が求められ, $ISW < 0$ のとき, 小さい方から M 個の固有値が求められる.

W1(5*N) (作業領域; 8 バイト実数)

IERR (出力; 4 バイト整数)

エラーインディケータ. 正常終了の場合 0 が返される. 異常終了の場合に返される IERR の値の詳細については, 日本電気株式会社発行のマニュアル⁹⁾を参照されたい.

CALL DCSTSS(D, N, SD, EPS, E, M, VE, LNV, ISW, IW1, W1, IERR)

実対称 3 重行列の固有値をバイセクション法により, 大きい方から M 個, または小さい方から M 個求め, それらに対応する固有ベクトルを逆反復法により求める.

[引数]

D(N) (入力; 8 バイト実数)

対角化する 3 重対角行列の対角要素. 第 i 行第 i 列の行列要素を D(i) に与える.

N (入力; 4 バイト整数)

対角化する 3 重対角行列の次元 ($N \geq 2$).

SD(N) (入力; 8 バイト実数)

対角化する 3 重対角行列の副対角要素. 第 i 行第 $i+1$ 列の行列要素を $SD(i)$ に与える ($1 \leq i \leq N-1$).

EPS (入力; 8 バイト実数)

固有値の収束判定のための絶対誤差の上限を与えるパラメータ. $EPS \geq 0.0$ のときには入力値が与えられ, $EPS < 0.0$ のときには内部で自動的に値が設定される.

E(M) (出力; 8 バイト実数)

$E(1), E(2), \dots, E(M)$ の順に, 大きい方から M 個, または小さい方から M 個の固有値が返される.

M (入力; 4 バイト整数)

求めたい固有値, 固有ベクトルの個数. $1 \leq M \leq N$ でなければならない.

VE(LNV, M) 出力; (8 バイト実数)

M 個の固有ベクトルが返される. $E(i)$ に対応するものが $VE(1, i)$ から $VE(N, i)$ までという順に並べられている.

LNV (入力; 4 バイト整数)

配列 VE の整合寸法. $LNV \geq N$ でなければならない.

ISW (入力; 4 バイト整数)

処理スイッチ. $ISW \geq 0$ のとき, 大きい方から M 個の固有値およびそれらに対応する固有ベクトルが求められ, $ISW < 0$ のとき, 小さい方から M 個の固有値およびそれらに対応する固有ベクトルが求められる.

IW1(M) (作業領域; 4 バイト整数)

W1(6*N) (作業領域; 8 バイト実数)

IERR (出力; 4 バイト整数)

エラーインディケータ. 正常終了の場合 0 が返される. 異常終了の場合に返される $IERR$ の値の詳細については, 日本電気株式会社発行のマニュアル⁹⁾ を参照されたい.

CALL DCSTSN(D, N, SD, EPS, E, M, ISW, W1, IERR)

実対称 3 重行列の固有値をバイセクション法により, 大きい方から M 個, または小さい方から M 個求める.

[引数]

D(N) (入力; 8 バイト実数)

対角化する 3 重対角行列の対角要素. 第 i 行第 i 列の行列要素を $D(i)$ に与える.

N (入力; 4 バイト整数)

対角化する 3 重対角行列の次元 ($N \geq 2$).

SD(N) (入力; 8 バイト実数)

対角化する 3 重対角行列の副対角要素. 第 i 行第 $i+1$ 列の行列要素を $SD(i)$ に与える ($1 \leq i \leq N-1$).

EPS (入力; 8 バイト実数)

固有値の収束判定のための絶対誤差の上限を与えるパラメータ。EPS \geq 0.0 のときには入力値が与えられ、EPS $<$ 0.0 のときには内部で自動的に値が設定される。

E(M) (出力; 8 バイト実数)

E(1), E(2), ..., E(M) の順に, 大きい方から M 個, または小さい方から M 個の固有値が返される。

M (入力; 4 バイト整数)

求めたい固有値の個数。1 \leq M \leq N でなければならない。

ISW (入力; 4 バイト整数)

処理スイッチ。ISW \geq 0 のとき, 大きい方から M 個の固有値が求められ, ISW $<$ 0 のとき, 小さい方から M 個の固有値が求められる。

W1(3*N) (作業領域; 8 バイト実数)

IERR (出力; 4 バイト整数)

エラーインディケータ。正常終了の場合 0 が返される。異常終了の場合に返される IERR の値の詳細については, 日本電気株式会社発行のマニュアル⁹⁾を参照されたい。

R = RAND(A)

一様乱数を求める。なお, RAND の型は 4 バイト実数である。

[引数]

A (入力; 4 バイト実数)

求めたい一様乱数の最大値を与える (0 以上 A 以下の実数値をとる一様乱数が求められる)。

一方, SUN 用の版では, 固有な組込み関数副プログラム IRAND (一様乱数) を用いている。他のワークステーションへの移植に際しては, これらの部分を書換えなければならない。勿論, これらの書換えを行えば, この版を大型汎用機またはスーパーコンピュータで走らすことも出来る。この副プログラムの概要は次の通りである。

IR = IRAND(IFLAG)

一様乱数を求める。なお, IRAND の型は 4 バイト整数である。

[引数]

IFLAG (入力; 4 バイト整数)

IFLAG に 0 を与えると, 0 以上 2147483647 以下の整数値をとる一様乱数が求められる。