

6 1次元格子振動 (江藤 幹雄)

物理学に現れる多くの問題は、線形計算に帰着される。それらは行列を用いて表すことができるが、それは計算機を用いて連立方程式を解いたり、固有値を求めるのに非常に適した形式である。ここでは行列の固有値問題の1例として、1次元物質の格子振動を取り上げる。

6.1 フォノン

固体中、それぞれの原子は平衡点の周りで微小に振動している。ところが原子と原子の間には相互作用が働いているので、それぞれの振動は独立ではなく、連成振動を行うことになる。その連成振動の規準振動数が格子振動の固有モードとなり、それを量子化したものをフォノン (phonon) と呼ぶ。即ち、規準振動数が ω の時、そのモードはエネルギー $\hbar\omega$ を持つ粒子のように振る舞う。

フォノンはそれ自体、物質の比熱に寄与する (c.f. Einstein 模型、Debye 模型) 他、その分散関係 (ω の波数依存性) は中性子散乱などの実験で直接観測される。電子とフォノンの相互作用は電気伝導に大きな影響を与える一方、低温で超伝導が生ずる原因にもなる。また温度や圧力を変えていく時に見られる結晶の構造相転移には、フォノンの特性の異常が伴う。このようにフォノンの研究は固体物理の分野で非常に重要である。ここではフォノンの状態密度に現れる特異性 (van Hove singularity) を見てみよう。

6.2 計算技法

行列の対角化にはさまざまな手法があって、行列の性質によって使い分ける必要がある。それは自分で programming することもできるが、ここでは数値計算のライブラリーを利用する方法を学ぶ。

ライブラリーを利用する際の注意点としては、

- まずマニュアルを調べ、どういう計算手法を用いているのか、その手法が今の問題に適しているか、を必ずチェックする。Black box 的には決して使わない。
- マニュアルに従って、作業領域を確保し call する。その際、入力データが書き換えられる場合もあるので注意する。また行列計算の時の配列の準備の仕方について1つコメントをしておく。行列 $A(N, M)$ の入力の際、それを2次元配列で入力する時と、1次元化して入力する場合とがある。対称行列 $A(N, N)$ に対して大きさ $N(N+1)/2$ の1次元配列を要求された場合、

$$A(1, 1), A(2, 1), A(2, 2), A(3, 1), A(3, 2), A(3, 3), \dots, A(N, 1), A(N, 2), \dots, A(N, N)$$

の順序で 1 次元配列を作って入力データに用いる。

- 前もって、解が分かっている問題に対してライブラリーを用いた計算を行い、正常に動作することを確認しておく。この章の最初の 2 題では解析解が得られるので、その確認になる。(実際に数値計算を行うのは解析解が得られない問題に対してであるが、そのような問題でもパラメーターを特別な値に設定すると解析解が得られる場合がある。そのような時に計算結果と解析解を比較することでプログラムのチェックをすることができる。)

6.3 課題

- 6-A. 同種粒子からなる 1 次元系の、鎖方向の固有振動モードを求めよ。系は N 個の粒子からなり、周期的境界条件をつけるものとする。

(解) i 番目の原子の平衡点からの変位を x_i とする。原子間の相互作用の内、最近接間のみを考慮し、またそれが原子間の距離のみに依存すると仮定すると Lagrangian は

$$L = \sum_{i=1}^N \frac{M}{2} \dot{x}_i^2 - \sum_{i=1}^N V(x_{i+1} - x_i).$$

ここで、 $x_{N+1} = x_1$ とした。微小変位に対しては原子間の相互作用 V を次のように近似できる。

$$V(x_{i+1} - x_i) \approx V(0) + V'(0)(x_{i+1} - x_i) + \frac{1}{2}V''(0)(x_{i+1} - x_i)^2 = \frac{K}{2}(x_{i+1} - x_i)^2.$$

最右辺では平衡点の条件から $V'(0) = 0$ 、 $K \equiv V''(0) > 0$ とし、また定数を省いた。これより Lagrange の運動方程式は、

$$M\ddot{x}_i + K(2x_i - x_{i+1} - x_{i-1}) = 0.$$

規準振動を求めるために $x_i = A_i \exp(-i\omega t)$ とおく¹。さらに ω を $\omega_0 = \sqrt{K/M}$ を単位として測ることにすると、運動方程式は次の行列の形に帰着する。

$$(\mathcal{M} - \omega^2 \mathbf{1})\mathbf{A} = \begin{pmatrix} -\omega^2 + 2 & -1 & & & -1 \\ -1 & -\omega^2 + 2 & -1 & & \\ & \cdots & \cdots & \cdots & \\ & & -1 & -\omega^2 + 2 & -1 \\ -1 & & & -1 & -\omega^2 + 2 \end{pmatrix} \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{pmatrix} = \mathbf{0}$$

まず上述の行列 \mathcal{M} を作るプログラムを書く。次に固有値を計算するライブラリーを call し、 \mathcal{M} の固有値を求める。最後にその root を取れば規準振動数 ω が求ま

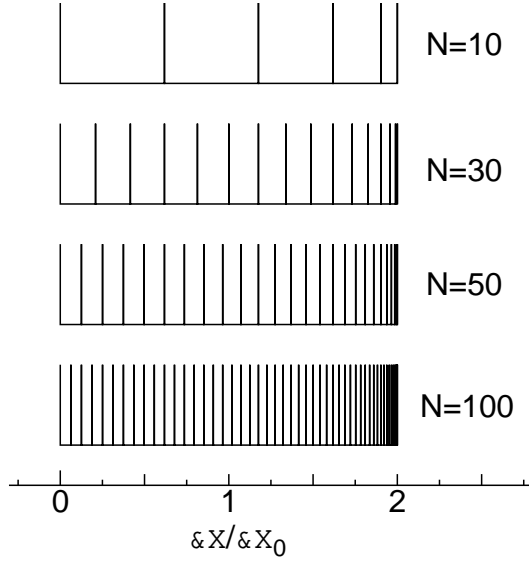


図 1

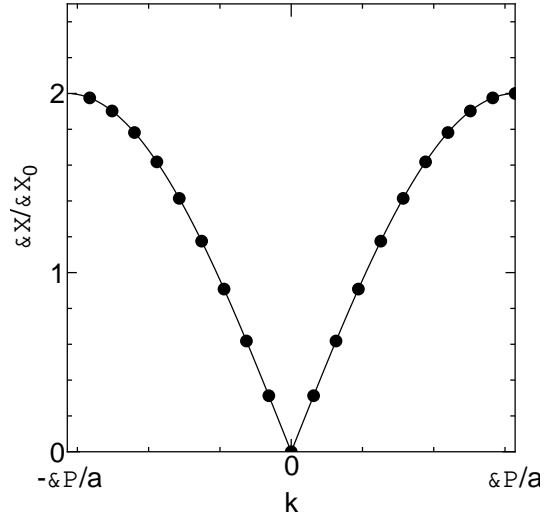


図 2

る。その分布の様子を $N = 10, 30, 50, 100$ に対してそれぞれ表したのが図 1 である。

実はこの問題は、Fourier 展開を利用することで容易に解くことが可能である。波数ベクトル k の成分がそのまま方程式 (1) の解になる。なぜならば、

$$x_i = A_0 e^{i(kai - \omega t)},$$

とし (a は原子間の平均距離)、(1) に代入すると、

$$-M\omega^2 + K(2 - e^{ika} - e^{-ika}) = 0,$$

これを整理して $\omega = 2\omega_0 |\sin ka/2|$ 、とすれば良いことが分かる。この ω の k 依存性を分散関係と呼ぶ。そのグラフを図 2 に示す。

有限系では波数 k の取る値が境界条件から限られる。条件

$$e^{-ikaN} = 1,$$

から $N = 2n$ の時、

$$k = \frac{2\pi}{a} \frac{j}{N}. \quad (j = -n + 1, -n + 2, \dots, 0, 1, \dots, n)$$

従って固有モードの値は、

$$\omega_j/\omega_0 = 2|\sin \pi j/N|.$$

¹ $x_i = A_i \exp(i\omega t)$ としても良い。が、後で i に対しても Fourier 展開をして $e^{i(kai - \omega t)}$ の形が出るが、それが $k > 0$ の時に進行波に対応するように ω の前に minus 符号を付けた。

図2の黒丸は $N=20$ の場合を示す。数値計算で得た規準振動の値が、この解析解と一致していることを確かめて欲しい²。

最後に固有モードの状態密度を計算して見よう。規準振動数が ω と $\omega + d\omega$ の間にあるモードの数を $N(\omega)$ と定義する。それは単位振動数当たりのモードの数を表すので状態密度と呼ばれる。その値は

$$N(\omega)d\omega = \frac{aN}{2\pi}dk \times 2,$$

factor 2 は $\pm k$ の寄与を取り入れた。これから

$$N(\omega) = \frac{aN}{2\pi} \frac{dk}{d\omega} = \frac{N}{\pi} \frac{1}{\omega_0 \cos ka/2} = \frac{N}{\pi \omega_0 \sqrt{1 - \omega^2/4\omega_0^2}}.$$

従って、 $N(\omega)$ は $\omega = 2\omega_0$ 付近で $\sim 1/\sqrt{1 - \omega/2\omega_0}$ のように発散する。これを van Hove singularity と呼ぶ³。

数値計算で状態密度の分布を調べるために、 ω に対してモードの数をヒストグラムで表したのが図3であるが、その特異性が現れている。

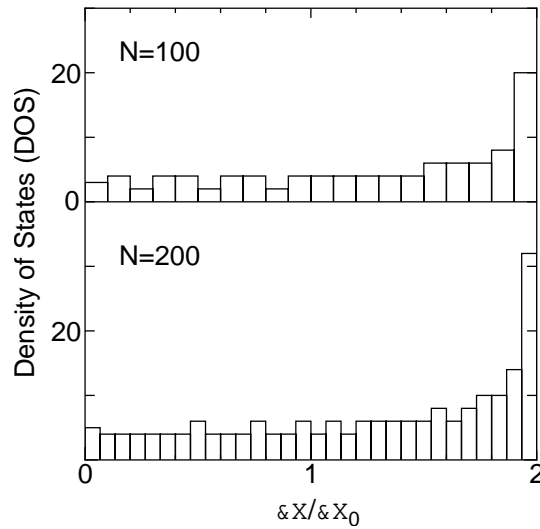


図3

6-B. 2種類の粒子が1つ置きに格子を組んでいる1次元系において、格子振動の固有モードを計算せよ。2種の粒子の質量比を $\gamma = M_2/M_1$ とし、それを変化させて振動モードの分布がどう変化するかを調べよ。

²今 $N = 2n$ としたが、 N が奇数でもかまわない。また j の範囲を $0, 1, \dots, (N-1)$ 等としても良いが独立なモードの数(自由度)は常に N である。上のようにとると $-\pi/a \leq k \leq \pi/a$ となり、その範囲を第1ブリルアン・ゾーンと呼ぶ。それは2次元、3次元にも拡張される。

³van Hove singularity は2次元以上でも存在するが、その特異性の度合は次元が増えるにつれて弱くなる。

(ヒント) 運動方程式は次式のようにになる。

$$\begin{aligned} M_1 \ddot{x}_i + K(2x_i - x_{i+1} - x_{i-1}) &= 0 \quad \text{for odd } i, \\ M_2 \ddot{x}_i + K(2x_i - x_{i+1} - x_{i-1}) &= 0 \quad \text{for even } i. \end{aligned}$$

$x_i = A_i \exp(-i\omega t)$ とおき、さらに ω を $\omega_0 = \sqrt{K/M_1}$ を単位として測り、運動方程式を行列の形で表す (対称行列にならない点に注意)。

図4に計算結果を示すが、固有モードの分布にギャップが生じていることがわかる。そのギャップは γ の値を大きくするにつれて広がっている。またギャップの両端と ω の上限の点で状態密度に特異性が現れるのがわかる。

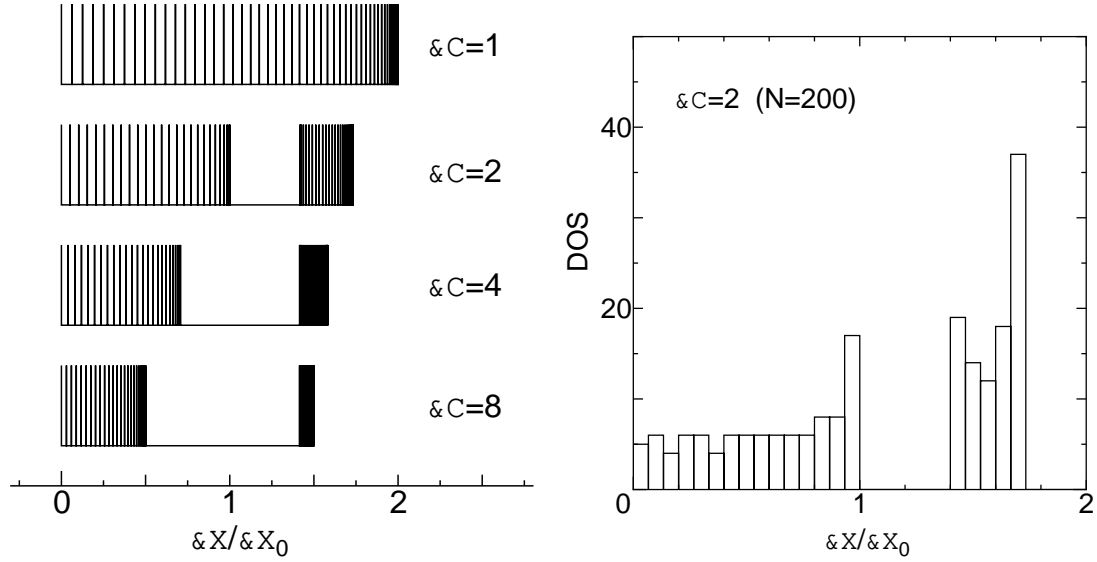


図 4

実はこの問題も以下のようにして解析解が求められる。
2 格子分の並進対称性を考え、次の形の解を仮定する。

$$x_i = C_1 e^{i(kai - \omega t)} \quad (\text{odd } i), \quad x_i = C_2 e^{i(kai - \omega t)} \quad (\text{even } i).$$

(ここで周期が $2a$ となることから、 $-\pi/2a \leq k \leq \pi/2a$ 、即ち第1ブリルアン・ゾーンが半分になることに注意。) これを運動方程式に代入すると

$$\begin{aligned} -\omega^2 M_1 C_1 + 2K C_1 - K(e^{ika} + e^{-ika}) C_2 &= 0 \\ -\omega^2 M_2 C_2 + 2K C_2 - K(e^{ika} + e^{-ika}) C_1 &= 0. \end{aligned}$$

これが $C_1 = C_2 = 0$ 以外の解を持つ条件より、

$$\begin{vmatrix} 2K - M_1 \omega^2 & -K(e^{ika} + e^{-ika}) \\ -K(e^{ika} + e^{-ika}) & 2K - M_2 \omega^2 \end{vmatrix} = 0.$$

これを解くことで、

$$\begin{aligned}\omega^2 &= \frac{K}{M_1 M_2} \left\{ (M_1 + M_2) \pm \sqrt{(M_1 + M_2)^2 - 4M_1 M_2 (1 - \cos^2 ka)} \right\} \\ &= \frac{\omega_0^2}{\gamma} \left\{ (1 + \gamma) \pm \sqrt{(1 + \gamma)^2 - 4\gamma \sin^2 ka} \right\},\end{aligned}$$

の2つのモードが得られる。これをそれぞれ光学フォノン (optical phonon)、音響フォノン (acoustic phonon) と呼ぶ (図5)。上述のギャップは、この2種類のモードの間に対応していたのであった。

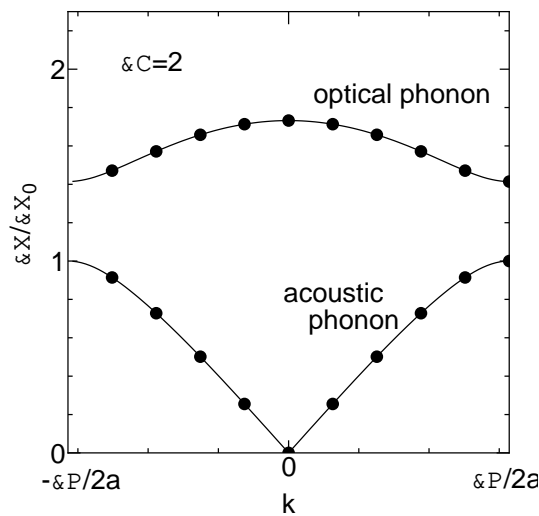


図5

- 6-C. 2種類の粒子がランダムな順序で格子を組んでいる1次元系では、振動モードの分布はどうなるか。但し、バネ定数 K は一定と仮定する。(これは合金やアモルファス半導体でのフォノンを研究するためのモデルになる。このような系では数値計算に頼らざるを得ない。)
- 6-D. 2種類の粒子が Fibonacci 列の順序で現れる1次元系では、振動モードの分布はどうなるか (バネ定数 K は一定と仮定)。但し Fibonacci 列とは次のような列を言う。

$$a_0 = [A], a_1 = [AB], a_2 = [ABA], a_3 = [ABAAB], \dots, (a_{n+1}; a_n, a_{n-1} \text{の合成})$$

ここで n を世代数と呼ぶが、各世代で周期的境界条件を課してモードを計算し、その分布の世代数依存性を調べよ。(この系はランダム系と同様に周期性を持たないが、ランダム系よりは規則性があり、広義の結晶という意味で“準結晶”と呼ばれる。実際の物質でも準結晶の構造を持つものが存在する。ref. 米沢富美子「準結晶」, 物理学最前線。)

6.4 話題

固体や分子中の電子状態の計算を行う場合でも、行列の固有値問題に帰着させる手法が幅広く使われている。即ち、適当な基底関数を準備して、波動関数をその線形結合で表現する時、Schrödinger 方程式を解いてその線形結合の係数を決定するのは、固有値問題になる。その基底関数としては、固体の構成要素である原子の波動関数を用いる場合 (LCAO (Linear combination of atomic orbitals) 法)、平面波を採用する場合 (平面波展開)、の 2 つが良く使われる。

電子間の相互作用がない場合の Schrödinger 方程式はそのまま固有値問題に帰着するが、実際の物質中では電子間にクーロン相互作用が働いている。多くの物質では、電子間相互作用の効果を平均場近似で取り入れれば充分であることが知られている。その計算方法は以下の通りである。(1) 波動関数の値として適当な初期条件を選び、それが作るクーロン相互作用の静電ポテンシャルを計算する。(2) そのポテンシャルを含んだ Schrödinger 方程式を解き、波動関数を求める。(3) 新しく求めた波動関数から静電ポテンシャルを作り直して、再び Schrödinger 方程式を解く。(4) 方程式を解く前と後の波動関数が一致するまでそれを繰り返す。即ち、自己無撞着な解を得るまで iteration を行う。

上の説明では簡単のために静電ポテンシャルという言葉を用いたが、実際には量子力学的な効果 (交換効果、相関効果) も存在する。それらも含んだ平均場の作り方にはいくつか方法があり、分子の場合の Hartree-Fock 法、固体のバンド計算で使われる局所密度汎関数近似が有名である。特に後者を用いることで、半導体や金属中の電子状態は非常に精度良く求められ、ほとんどの実験結果を定量的に説明することが現在可能となっている。

6.5 参考文献

フォノンの比熱は、統計物理の講義でも行う。分散関係、音響・光学フォノンの話は、固体物理の教科書に出ている。ここでは、キッテル, 「固体物理学入門(上)」(丸善)、を参考にしたが、Ashcroft and Mermin, “Solid State Physics,” (Saunders College)、Ziman, “Principles of the Theory of Solids,” (Cambridge University Press)、など代表的な本もある。バンド計算については、金森他, 「固体—構造と物性」(岩波講座, 現代の物理学 7)。

6.6 SSL II の使い方 (高野 宏)

いろいろなプログラムで共用できるサブルーチンを集めて、プログラムをコンパイル、リンクする際に必要なサブルーチンを組み込めるようにしたものを、ライブラリといいます。physhpk の上で、数値計算のサブルーチンを集めたライブラリとして、富士通の SSL II を利用することができます。この、SSL II は、パー

ソナルコンピュータからスーパーコンピュータまで広い範囲の計算機で利用することができます。このほか、有名な数値計算のライブラリとして、IMSL、NAG 等があります。利用法としては、どれもほとんど同様です。ここでは、SSL II の利用法について説明します。

6.6.1 マニュアル

SSL II のマニュアルとして、

FUJITSU SSL II 使用手引書 (科学用サブルーチンライブラリ)

FUJITSU SSL II 拡張機能使用手引書 (科学用サブルーチンライブラリ)

があり、学生実験室に備えてあります。各サブルーチンを使用する際の詳細な注意事項及び手法概要については、これらのマニュアルを参照してください。例として、課題 6-A のサンプル・プログラムで用いている実対称行列の固有値および固有ベクトルを求めるサブルーチン `dseig2` に関する記述を示します。

6.6.2 オンライン・マニュアル

使用法などに関する簡単なオンライン・マニュアルが `physhpk` 上に用意されています。`physhpk` に `rlogin` して、このオンライン・マニュアルを見ることができます。まず、どのような用途のサブルーチンがどのような名前で用意されているかを調べるには、

```
% man ssl2 ↵
```

とタイプします。名前のわかった特定のサブルーチンの使用法を調べるには、

```
% man サブルーチン名 ↵
```

とタイプします。英語のマニュアルが表示された場合、

```
% setenv LANG japanese ↵
```

とタイプしてから、上記の '`man ssl2`' または、'`man サブルーチン名`' を実行します。

`kterm` を使用していて、日本語が正しく表示されなかった場合、マウス・カーソルを `kterm` のウィンドウ上にもって行って、**CTRL** キーを押しながら、マウスの中ボタンを押します (まだ押したボタンは離さないで下さい)。すると、VT Options というメニューが表示されますから、マウスの中ボタンを押したままマウス・カーソルを移動させ、Shift-JIS Kanji Mode と書かれた行の上にもってきます。そのとき、Shift-JIS Kanji Mode と書かれた行が白黒が逆になって表示されま

SEIG2

B 21—21—0201 SEIG2, DSEIG2

実対称行列の固有値及び固有ベクトル
(バイセクション法, 逆反復法)

CALL SEIG2 (A, N, M, E, EV, K, VW, ICON)

(1) 機能

n 次の実対称行列 A の固有値を, バイセクション法により大きい方から又は小さい方から m 個求め, 対応する固有ベクトルを逆反復法により求める. 固有ベクトルは, $\|x\|_2 = 1$ となるように正規化する.

$1 \leq m \leq n$ なること.

(2) パラメタ

A ……入力. 実対称行列 A .

対称行列用圧縮モード.

大きさ $n(n+1)/2$ の 1 次元配列.

演算後内容は保存されない.

N ……入力. 実対称行列 A の次数 n .

M ……入力. 求める固有値の個数 m .

$M = +m$ のときは大きい方から求める.

$M = -m$ のときは小さい方から求める.

E ……出力. 固有値.

大きさ m の 1 次元配列.

EV ……出力. 固有ベクトル.

固有ベクトルは列方向に格納される.

EV(K, M) なる 2 次元配列

K ……入力. 配列 EV の整合寸法 ($\geq n$).

VW …作業領域. 大きさ $7n$ の 1 次元配列.

ICON ……出力. コンディションコード.

表 SEIG2-1 参照.

表 SEIG2-1 コンディションコード

コード	意 味	処 理 内 容
0	エラーなし.	
10000	$N = 1$ であった.	$E(1) = A(1), EV(1,1) = 1.0$ とする.
15000	m 個の固有値を求めた後, 固有ベクトルを求めるとき, 求まらないものがあった.	その固有ベクトルを 0 ベクトルとする.
20000	固有ベクトルは求められなかった.	固有ベクトルはすべて 0 ベクトルとする.
30000	$M = 0$, $N < M $ 又は, $K < N$ であった.	処理を打ち切る.

(3) 使用上の注意

a. 使用する副プログラム

① SSL II ……AMACH, TEIG2, TRBK, TRID1, MGSSL, UTEG2

② FORTRAN 基本関数 ……IABS, SQRT, SIGN, ABS, AMAX1, DSQRT

b. 注意

① 本サブルーチンは, 実対称行列用である.

実対称 3 重対角行列の m 個の固有値及び固有ベクトルを求める場合は, サブルーチン TEIG2 を用いること.

② 実対称行列 A の m 個の固有値だけを求める場合は, サブルーチン TRID1 とサブルーチン BSCT1 を続けて用いること.

c. 使用例

n 次の実対称行列 A の大きい方から又は小さい方から m 個の固有値及び対応する固有ベクトルを求める.

$n \leq 100$, $m \leq 10$ の場合.

```

C      **EXAMPLE**
      DIMENSION A(5050), E(10), EV(100, 10), VW(700)
10    CONTINUE
      READ(5, 500) N, M
      IF(N.EQ. 0) STOP
      NN=N*(N+1)/2
      READ(5, 510) (A(I), I=1, NN)
      WRITE(6, 600) N, M
      NE=0
      DO 20 I=1, N
        NI=NE+1
        NE=NE+I
        WRITE(6, 610) I,(A(J), J=NI, NE)
20    CONTINUE
      CALL SEIG2(A, N, M, E, EV, 100, VW, ICON)
      WRITE(6, 620) ICON
      IF(ICON.GE. 20000) GO TO 10
      MM=IABS(M)
      CALL SEPRT(E, EV, 100, N, MM)
      GO TO 10
500  FORMAT(2I5)
510  FORMAT(5E15.7)
600  FORMAT(1H1, 20X, 15HORIGINAL MATRIX, 5X, 2HN=,
*      I3, 5X, 2HM=, I3/1H0)
610  FORMAT(1H0, 7X, I3, 5E20.7/(11X, 5E20.7))
620  FORMAT(1H0, 20X, 5HICON=, I5)
      END
  
```

本使用例中のサブルーチン SEPRT は, 実対称行列の固有値及び固有ベクトルを出力するサブルーチンである. その詳細については, サブルーチン SEIG1 の使用例を参照のこと.

526

(4) 手法概要

n 次の実対称行列 A の固有値を、バイセクション法により大きい方から又は小さい方から m 個求め、対応する固有ベクトルを逆反復法により求める。

まず実対称行列 A をハウスホルダー法により 3 重対角行列 T に変換する。この操作を (4.1) で示す。

$$T = Q_H^T A Q_H \quad (4.1)$$

ここに、 Q_H は直交行列である。

この操作は、サブルーチン **TRID1** を用いて行う。

次にバイセクション法により m 個の固有値を求め、逆反復法により、対応する T の固有ベクトルを求める。逆反復法は、

$$(T - \mu I) x_r = x_{r-1}, \quad r=1, 2, \dots \quad (4.2)$$

を反復的に解いて固有ベクトルを求める方法である。ただし、(4.2) で、 μ はバイセクション法により求めた固有値、 x_0 は適当な初期ベクトルである。

この操作は、サブルーチン **TEIG2** を用いて行う。

T の固有ベクトルを y とすれば、 A の固有ベクトル x は、(4.1) の Q_H を用いて、

$$x = Q_H y \quad (4.3)$$

により得られる。(4.3) はハウスホルダー法の逆変換である。この操作はサブルーチン **TRBK** を用いて行う。

なお固有ベクトルは、 $\|x\|_2 = 1$ となっている。

なお、詳細については、参考文献 [12] 及び [13] pp.418

–439 を参照すること。

す。そこで、マウスの中ボタンを離してください。これらの操作をした後、‘man ssl2’ または、‘man サブルーチン名’ を実行すると、日本語が正しく表示されるはずです。

例として、課題 6-A のサンプル・プログラムで用いている実対称行列の固有値および固有ベクトルを求めるサブルーチン dseig2 に関するオンライン・マニュアルの記述を示します。この記述は

% man dseig2 ↵

とタイプして得られたものです。

SEIG2(3F) (Scientific Subroutine Library) SEIG2(3F)

[名称]

SEIG2, DSEIG2 — 実対称行列の固有値及び固有ベクトル
(バイセクション法, 逆反復法)

[記述形式]

CALL SEIG2(A,N,M,E,EV,K,VW,ICON)

[説明]

(1) 機能

n 次の実対称行列 A の固有値を, バイセクション法により大きい方から
または小さい方から m 個求め, 対応する固有ベクトルを逆反復法により
求める. 固有ベクトルは $\|x\|_2 = 1$ となるように正規化する.

$1 \leq m \leq n$ となること.

(2) パラメタ

A	入力	実対称行列 A . 対称行列用圧縮モード. 大きさ $n(n+1)/2$ の 1 次元配列. 演算後内容は保存されない.
N	入力	実対称行列 A の次数 n .
M	入力	求める固有値の個数 m . $M=+m$ のときは大きい方から求める. $M=-m$ のときは小さい方から求める.
E	出力	固有値. 大きさ m の 1 次元配列.
EV	出力	固有ベクトル. 固有ベクトルは列方向に格納される. $EV(K, M)$ なる 2 次元配列.
K	入力	配列 EV の整合寸法 ($\geq n$).
VW	作業領域	大きさ $7n$ の 1 次元配列.
ICON	出力	コンディションコード.

SEIG2(3F) (Scientific Subroutine Library) SEIG2(3F)

(3) コンディションコード

コード	意 味	処 理 内 容
0	エラーなし.	
1 0 0 0 0	N = 1 であった.	E (1) = A (1), E V (1 , 1) = 1 . 0 とする.
1 5 0 0 0	m個の固有値を求めた後, 固有 ベクトルを求めるとき, 求まら ないものがあった.	その固有ベクトルを 0 ベク トルとする.
2 0 0 0 0	固有ベクトルは求められなかつ た.	固有ベクトルはすべて 0 ベ クトルとする.
3 0 0 0 0	M = 0 , N < M または, K < N であった.	処理を打ち切る.

6.6.3 コンパイル、リンク

SSL II のサブルーチンを利用するには、マニュアルに書かれた使用法に従って、プログラム中にサブルーチン呼び出す call 文を書きます。そのプログラムのソースファイルをコンパイル、リンクするには、

```
% f90 ソースファイル名 -lssl2 ↵
```

とタイプします。最後の -lssl2 が、SSL II のライブラリをリンクすることを指定しています。既にコンパイルした、オブジェクトファイルのリンクのみを行なう場合も同様で、

```
% f90 オブジェクトファイル名 -lssl2 ↵
```

のように、最後に -lssl2 をつけて SSL II のライブラリをリンクすることを指定します。

6.6.4 利用状況を見る

SSL II を利用したプログラムは、ライセンスの関係で、physhpk, physhp{0,1,2,3,4,5}の上では同時に 6 本 (7 台の計算機上での合計) しか走りません。それより多い数のプログラムを走らそうとすると、実行は受け付けますが、SSL II を呼んでいるところで停止し、6 本の走っているプログラムの内のどれかが終了するのを待ちます。6 本の走っているプログラムの内のどれかが終了すると、そのプログラムが再び走りはじめます。複数のプログラムが待っている場合は、早い順に実行されていきます。現在の SSL II の利用状況を見るには、

```
% lmstat -a ↵
```

とタイプします。これにより、現在誰がどの計算機で SSL II を利用しているかがわかります。

6.7 プログラム例 (高野 宏)

課題 6-A の

$$\begin{aligned} M\ddot{x}_i + K(2x_i - x_{i+1} - x_{i-1}) &= 0; \\ i &= 1, \dots, N \\ x_{N+1} &= x_1, \quad x_0 = x_N \end{aligned}$$

の規準角振動数を求める問題を考える。

$$x_i = X_i \exp(-i\omega t)$$

とおき、

$$\mathbf{X} = \begin{pmatrix} X_1 \\ \vdots \\ X_N \end{pmatrix}$$

$$\mathbf{A} = (A_{i,j}), \quad A_{i,j} = 2\delta_{i,j} - \delta_{i+1,j} - \delta_{i-1,j}$$

$$\mathbf{M} = (M_{i,j}), \quad M_{i,j} = M\delta_{i,j}$$

とすると、

$$-\omega^2 \mathbf{M} \mathbf{X} + K \mathbf{A} \mathbf{X} = \mathbf{0}$$

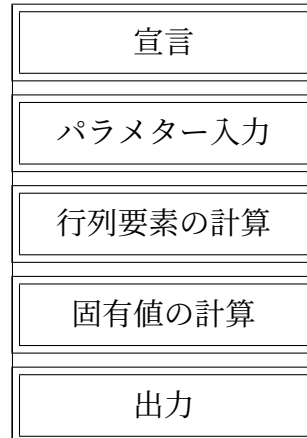
となる。ただし、 $\delta_{i\pm 1,j}$ は mod N で考えるものとする ($\delta_{N+1,1} = \delta_{0,N} = 1$)。 $\mathbf{M} = M\delta_{i,j}$ を用いると、この方程式は

$$\mathbf{A} \mathbf{X} = \lambda \mathbf{X}$$

$$\lambda = \omega^2 \frac{M}{K} = \left(\frac{\omega}{\omega_0} \right)^2$$

のように、対称行列 \mathbf{A} に対する固有値方程式になる。SSL II のサブルーチンを使って対称行列 \mathbf{A} の固有値 λ を求めるプログラム例が `/phys/tmp/comp-phys/samples/sec6/eigen1.f` である⁴。

プログラム全体は次のような構成になっている。



- | |
|----|
| 宣言 |
|----|

では、

<code>rmatrx(mmax)</code>	対称行列 \mathbf{A} を圧縮して格納する 1 次元配列
<code>eigval(nmax)</code>	固有値を返す 1 次元配列
<code>eigvec(nmax,nmax)</code>	固有ベクトルを返す 2 次元配列
<code>work (nmax7)</code>	サブルーチンの作業用 1 次元配列

⁴現在は削除 (山内)

を宣言している。nmax は扱う行列の最大次元で parameter 文で定義されている。mmax = nmax * (nmax + 1) / 2, nmax7 = nmax * 7 で parameter 文で定義されている。

1次元配列 rmatrix(mmax) には、 $N \times N$ の対称行列 $\mathbf{A} = (A_{i,j})$ の下三角成分 ($i \geq j$ の成分) が次のように格納される。

$$\begin{aligned} \text{rmatrix}(1) &= A_{1,1} \\ \text{rmatrix}(2) &= A_{2,1} \\ \text{rmatrix}(3) &= A_{2,2} \\ \text{rmatrix}(4) &= A_{3,1} \\ \text{rmatrix}(5) &= A_{3,2} \\ \text{rmatrix}(6) &= A_{3,3} \\ &\vdots \\ \text{rmatrix}(i(i-1)/2 + j) &= A_{i,j} \\ &\vdots \\ \text{rmatrix}(N(N+1)/2) &= A_{N,N} \end{aligned}$$

このように、 $i \geq j$ に対して、 $A_{i,j}$ は rmatrix の $k = i * (i - 1) / 2 + j$ 番目の要素として格納される。

- パラメーター入力

では、扱う行列の次元 (格子点数) ndim の入力、検査を行なっている。

- 行列要素の計算

では、すべての $i \geq j$ に対して $A_{i,j}$ を計算し (ここでは $A_{i,j} = 2\delta_{i,j} - \delta_{i+1,j} - \delta_{i-1,j}$) rmatrix(k) に代入している。ただし、 $k = i * (i - 1) / 2 + j$ 。

- 固有値の計算

では、対称行列の固有値、固有ベクトルを計算するサブルーチン dseig2 を call 文で呼んで、全ての固有値 (および固有ベクトル) を計算させている。その後、サブルーチンが正常に結果を出しているかどうかを検査している。

- 出力

では、得られた固有値の平方根を取り、標準出力に出力している。その際、得られた固有値が微小な負の数 ($-10^{-15} \sim 0$) の場合、その固有値を 0 とみなしている。これは、得られる固有値は、本来すべて 0 以上のはずであり、

0 になるべき固有値が誤差で微小な負の数になっているとみなしたものである。

eigen1.f のリスト

```

*-----
* 1-jigen koshi-shindo (doshu-ryushi)
*-----
      program phonon
*-----
* sengen
*-----
      implicit real*8(a-h,o-z)
c
      parameter(nmax =400)
      parameter(mmax =nmax*(nmax+1)/2)
      parameter(nmax7 =nmax*7)
c
      parameter(eps =1.0d-15)
c
      dimension rmatrx(mmax)
      dimension eigval(nmax)
      dimension eigvec(nmax,nmax)
      dimension work (nmax7)
*-----
* parameter nyuryoku
*-----
* nyuryoku
*-----
      write(7,*) 'number of lattice points ?'
      read (*,*) ndim
*-----
* kensa
*-----
      if((ndim.gt.nmax).or.(ndim.lt.1)) then
        write(*,*) 'error in ndim:',ndim
        stop
      endif
*-----
* gyoretsu-yoso no keisan
*-----
      do i=1,ndim
        if(i.eq.ndim) then
          ip=1
        else
          ip=i+1
        endif
        if(i.eq.1) then
          im=ndim
        else
          im=i-1
        endif
        do j=1,i
          k=(i*(i-1)/2)+j
          if(j.eq.i) then
            rmatrx(k)=2.0d0
          else if((j.eq.im).or.(j.eq.ip)) then
            rmatrx(k)=-1.0d0
          else
            rmatrx(k)=0.0d0
          endif
        enddo
      enddo
*-----
* gyoretsu no taikaku-ka (ssl2)
*-----
      call dseig2(rmatrx,ndim,-ndim,eigval,eigvec,nmax,work,icon)
*-----
* error check
*-----
      if(icon.ne.0) then
        write(*,*) 'ssl2(dseig2) error: icon=', icon
      endif
*-----
* shutsuryoku
*-----
      do j=1,ndim
        if((eigval(j).lt.0.0d0).and.(eigval(j).gt.-eps)) then
          eigval(j)=0.0d0
        endif
        eigval(j)=dsqrt(eigval(j))
        write(*,*) eigval(j)
      enddo
c
      stop
      end

```

6.8 課題 6-B の考え方 (高野 宏)

課題 6-B の

$$M_i \ddot{x}_i + K(2x_i - x_{i+1} - x_{i-1}) = 0;$$

$$i = 1, \dots, N$$

$$x_{N+1} = x_1, \quad x_0 = x_N$$

$$M_i = \begin{cases} M_1 & i \text{ が奇数} \\ M_2 & i \text{ が偶数} \end{cases}$$

の規準角振動数を求める問題を考える。

$$x_i = X_i \exp(-i\omega t)$$

とおき、

$$\mathbf{X} = \begin{pmatrix} X_1 \\ \vdots \\ X_N \end{pmatrix}$$

$$\mathbf{A} = (A_{i,j}), \quad A_{i,j} = 2\delta_{i,j} - \delta_{i+1,j} - \delta_{i-1,j}$$

$$\mathbf{M} = (M_{i,j}), \quad M_{i,j} = M_i \delta_{i,j}$$

とすると、

$$-\omega^2 \mathbf{M} \mathbf{X} + K \mathbf{A} \mathbf{X} = \mathbf{0}$$

となる。ただし、 $\delta_{i\pm 1,j}$ は mod N で考えるものとする ($\delta_{N+1,1} = \delta_{0,N} = 1$)。この方程式は

$$\mathbf{A} \mathbf{X} = \lambda \mathbf{M} \mathbf{X}$$

$$\lambda = \frac{\omega^2}{K}$$

のように、対称行列 \mathbf{A} と \mathbf{M} に対する固有値方程式になる。このような形の固有値問題を、一般化固有値問題という。

対称行列 \mathbf{M} が正定値 (全ての固有値が正) の場合、上記の一般化固有値問題は、次のように普通の固有値問題に書き直すことができる。

$$\mathbf{A} \mathbf{X} = \lambda \mathbf{M} \mathbf{X}$$

$$\mathbf{A} \mathbf{M}^{-1/2} \mathbf{M}^{1/2} \mathbf{X} = \lambda \mathbf{M}^{1/2} \mathbf{M}^{1/2} \mathbf{X}$$

$$\mathbf{M}^{-1/2} \mathbf{A} \mathbf{M}^{-1/2} \mathbf{M}^{1/2} \mathbf{X} = \mathbf{M}^{-1/2} \lambda \mathbf{M}^{1/2} \mathbf{M}^{1/2} \mathbf{X}$$

$$\mathbf{M}^{-1/2} \mathbf{A} \mathbf{M}^{-1/2} \mathbf{M}^{1/2} \mathbf{X} = \lambda \mathbf{M}^{1/2} \mathbf{X}$$

これより、

$$\tilde{\mathbf{A}}\tilde{\mathbf{X}} = \lambda\tilde{\mathbf{X}}$$

ただし、

$$\tilde{\mathbf{A}} = \mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{-1/2}$$

$$\tilde{\mathbf{X}} = \mathbf{M}^{1/2}\mathbf{X}$$

今、

$$(\mathbf{M}^{-1/2})_{i,j} = M_i^{-1/2}\delta_{i,j}$$

であることより、

$$(\tilde{\mathbf{A}})_{i,j} = M_i^{-1/2}A_{i,j}M_j^{-1/2}$$

を得る。特に、 $\gamma = M_2/M_1$ として

$$M_i = \begin{cases} M_1 & i \text{ が奇数} \\ M_2 = \gamma M_1 & i \text{ が偶数} \end{cases}$$

を用いると

$$(\tilde{\mathbf{A}})_{i,j} = \frac{1}{M_1}c_i^{-1/2}A_{i,j}c_j^{-1/2}$$

となる。ただし、

$$c_i = \begin{cases} 1 & i \text{ が奇数} \\ \gamma & i \text{ が偶数} \end{cases}.$$

これより

$$(\hat{\mathbf{A}})_{i,j} = c_i^{-1/2}A_{i,j}c_j^{-1/2}$$

$$\hat{\lambda} = M_1\lambda = \frac{\omega^2 M_1}{K} = \left(\frac{\omega}{\omega_0}\right)^2$$

とすると、

$$\hat{\mathbf{A}}\tilde{\mathbf{X}} = \hat{\lambda}\tilde{\mathbf{X}}$$

となる。

結局、

$$(\hat{\mathbf{A}})_{i,j} = c_i^{-1/2}A_{i,j}c_j^{-1/2}$$

$$A_{i,j} = 2\delta_{i,j} - \delta_{i+1,j} - \delta_{i-1,j}$$

$$c_i = \begin{cases} 1 & i \text{ が奇数} \\ \gamma & i \text{ が偶数} \end{cases}$$

で与えられる対称行列 \hat{A} の固有値 $\hat{\lambda}$ を求め、 $\hat{\lambda} = (\omega/\omega_0)^2$ より規準角振動数 ω を求めればよいことになる。この問題を数値的に解くには、課題 6-A 用のサンプルプログラムで、行列要素の計算をしている部分を書き換えるだけでよいことがわかる。

6.9 LAPACK の使い方 (山内 淳)

2005 年度にシステム更新のため、6.6 節で説明されていた旧システム physhpk 上の数値計算ライブラリ SSL II が使用できなくなりました。このためサンプルプログラムを新しいシステム physhprx.cm.phys.keio.ac.jp(当時のシステム、その後継 ks3.educ.cc.keio.ac.jp、現在の理工学 ITC の WS ルーム内の Linux PC システムでも同様) で利用可能なライブラリ LAPACK(version 3.0) を使うように書き換えました。基本的な使用方法、考え方は SSL II と同じなので、ここでは要点だけを説明します。

LAPACK は Linear Algebra PACKage の略でその名の通り線形代数の数値計算ライブラリであり、ソースコードが基本的に無料で公開 (<http://www.netlib.org/lapack/>) されています。一方で、様々なコンピューターベンダーが、自社マシン用にチューニングしたライブラリー (無料、有償) を提供しており、使用可能な環境であればベンダーが提供するライブラリーの方が多くの場合効率的です。LAPACK は、なるべく多くの部分を基本線形代数副プログラム (BLAS) への呼出しによって行うように設計されており、これによって移植性を高めています。というわけで、環境によっては、LAPACK の他に BLAS ライブラリを別途指定する必要があります。

6.9.1 マニュアル

LAPACK の公式な日本語マニュアルは用意されていません。オンラインでは、http://www.netlib.org/lapack/lug/lapack_lug.html で参照できます。他にも、種々のマニュアルが、<http://www.netlib.org/lapack/> にて提供されています。やや古い (version 2.0) が、丸善より日本語訳 (「行列計算パッケージ LAPACK 利用の手引」小国 力訳) も出ています。

6.9.2 オンライン・マニュアル

オンラインマニュアルは理工学 ITC の WS(ワークステーション) 上で使用できますが、英語版のみです。6.6 節の SSL II の場合と同様に使えます。

例えば、LAPACK 用に改編した課題 6-A のサンプル・プログラムで用いている実対称行列の固有値および固有ベクトルを求める倍精度用のサブルーチン dspev に関するオンライン・マニュアルを呼ぶには、次のようにタイプします。

```
% man dspev ↵
```

(注意) 次の man ページの出力は、以前のシステムでのものですが、基本的にほとんど変わりませんのでそのままにしています。

DSPEV(1) LAPACK version 3.0
 LAPACK driver routine (version 3.0) LAPACK driver routine (version 3.0)

NAME

DSPEV - compute all the eigenvalues and, optionally, eigenvectors of a real symmetric matrix A in packed storage

SYNOPSIS

SUBROUTINE DSPEV(JOBZ, UPLO, N, AP, W, Z, LDZ, WORK, INFO)

CHARACTER JOBZ, UPLO

INTEGER INFO, LDZ, N

DOUBLE PRECISION AP(*), W(*), WORK(*), Z(LDZ, *)

PURPOSE

DSPEV computes all the eigenvalues and, optionally, eigenvectors of a real symmetric matrix A in packed storage.

ARGUMENTS

JOBZ (input) CHARACTER*1
 = 'N': Compute eigenvalues only;
 = 'V': Compute eigenvalues and eigenvectors.

UPLO (input) CHARACTER*1
 = 'U': Upper triangle of A is stored;
 = 'L': Lower triangle of A is stored.

N (input) INTEGER
 The order of the matrix A. $N \geq 0$.

AP (input/output) DOUBLE PRECISION array, dimension $(N*(N+1)/2)$
 On entry, the upper or lower triangle of the symmetric matrix A, packed columnwise in a linear array. The j-th column of A is stored in the array AP as follows: if UPLO = 'U', $AP(i + (j-1)*j/2) = A(i,j)$ for $1 \leq i \leq j$; if UPLO = 'L', $AP(i + (j-1)*(2*n-j)/2) = A(i,j)$ for $j \leq i \leq n$.

On exit, AP is overwritten by values generated during the reduction to tridiagonal form. If UPLO = 'U', the diagonal and first superdiagonal of the tridiagonal matrix T overwrite the corresponding elements of A, and if UPLO = 'L', the diagonal and first subdiagonal of T overwrite the corresponding elements of A.

W (output) DOUBLE PRECISION array, dimension (N)
 If INFO = 0, the eigenvalues in ascending order.

Z (output) DOUBLE PRECISION array, dimension (LDZ, N)
 If JOBZ = 'V', then if INFO = 0, Z contains the orthonormal

- 1 - Formatted: December 16, 2005

LAPACK のオンライン・マニュアルの出力例 (1/2)

eigenvectors of the matrix A, with the i-th column of Z holding the eigenvector associated with W(i). If JOBZ = 'N', then Z is not referenced.

LDZ (input) INTEGER
The leading dimension of the array Z. LDZ \geq 1, and if JOBZ = 'V', LDZ \geq max(1,N).

WORK (workspace) DOUBLE PRECISION array, dimension (3*N)

INFO (output) INTEGER
= 0: successful exit.
< 0: if INFO = -i, the i-th argument had an illegal value.
> 0: if INFO = i, the algorithm failed to converge; i off-diagonal elements of an intermediate tridiagonal form did not converge to zero.

6.9.3 コンパイル、リンク

LAPACK のサブルーチンを利用するには、マニュアルに書かれた使用方法に従って、プログラム中にサブルーチンを呼び出す call 文を書きます。

そのプログラムのソースファイルをコンパイル、リンクするには、

```
% gfortran ソースファイル名 -lm -llapack ↩
```

とタイプします。

ここで、-l(ライブラリ名) がライブラリの指定です。一般には、ライブラリの置かれているディレクトリを -L(ディレクトリ名) で指定する必要がありますが、今回のケースではデフォルトとして設定されているライブラリディレクトリにライブラリがあるため、明記していません。実際にライブラリが置いてあるディレクトリ /usr/lib64 を ls コマンドで見るとライブラリファイルは全て文字列'lib'で始まっています。このような約束があるので、-l の次に指定するライブラリ名は文字列'lib'と拡張子 (.a,.so) を取り去ったもので指定します。因みに、.a と .so はそれぞれ Archived library, Shared Object の意味で、static 並びに dynamic link に対応します。

最初の -lm が、数学ライブラリ libm.so、次の -llapack が、LAPACK のライブラリ liblapack.so をリンクすることを指定しています。

既にコンパイルした、オブジェクトファイルのリンクのみを行なう場合も同様に、

```
% gfortran オブジェクトファイル名 -lm -llapack ↩
```

のように、LAPACK のライブラリをリンクすることを指定します。

さて、上記のようにコンパイル時に、色々なライブラリ指定をするのは、場合によっては、かなり面倒なものです。それを避ける方法の一つとして make コマンドの利用があります。これは、予めコンパイルの方法 (コンパイラ、オプション、ライブラリ) や、ソースファイル、ファイル間の依存性等の実行ファイルを生成するための規則を記述したファイル (makefile あるいは Makefile) を作成しておき、コンパイルの実行は

```
% make ターゲット名 ↩
```

のように行います。ここでターゲット名と言うのは目的とする動作につける名前のことで、通常は生成したい実行ファイル名です。

その他に、ソースを複数のファイルに分けているような時に、一部のファイルを更新した後、make を動かすと、オブジェクトファイルとソースファイルのタイムスタンプを比較して、変更のあったファイルだけを再コンパイルしてくれる等 make コマンドは多様な機能を持っています。

サンプルファイルの置いてあるディレクトリに makefile がおいてありますので、自分のディレクトリにコピーして使ってみてください。ファイルの中身を見ると

分かるように、ターゲット名として `e1.x` を指定すると、この後説明する `eigen1.f90` のコンパイルをしてくれます。因みに、ターゲット名として `clean` を指定すると、コンパイルに使用したオブジェクトファイル、その結果としての実行ファイルを削除してくれます。このように一つの `makefile` に複数のターゲットを設定することができます。

6.9.4 プログラム例

次ページに LAPACK を使用するように書き換えた `eigen1.f90` のリストを示します。

ここでは、パラメタ文で、`JOBZ='V'` と `UPLO='U'` を指定することで、LAPACK の実対称行列対角化 subroutine `dspev` に「固有値だけでなく固有ベクトルも計算する」こと、「圧縮配列として引きわたすのは上三角部分である」ことを指定しています⁵。

中程の行列要素を代入する部分は、もっと直接的に次のようにも書けます。

```
! Make MATRIX A(i,j)=A(j*(j-1)/2+i)
!
! Initialize(zero clear)
  rmatrx(:)=0.0d0
! Diagonal component(j,j)
  do j=1,ndim
    k=(j*(j-1)/2)+j
    rmatrx(k)=2.0d0
  end do
! (j-1,j) component
  do j=2,ndim
    k=(j*(j-1)/2)+j-1
    rmatrx(k)=-1.d0
  end do
! (1,n) component
  rmatrx( ndim*(ndim-1)/2+1 )=-1.d0
```

この場合はほとんどの行列要素が 0 なので、先ず全体の行列要素に 0.0d0 を代入した後で、対角成分に 2.0d0 を代入し、subroutine に引きわたす上対角部分の $(j-1, j)$ 要素に -1.0d0 を代入し、最後に $(1, n)$ 成分 (右上角) に -1.0d0 を代入しています。

⁵SSL2 版では下三角

LAPACK を利用するように書き換えられた eigen1.f90 のリスト

```

=====
! 1-jigen koshi-shindo (doshu-ryushi)
=====
program phonon
!=====
! sengen
!=====
implicit none
!
integer,parameter:: nmax =400, mmax =nmax*(nmax+1)/2
real(kind=8),parameter:: eps =1.0d-15
!
integer:: ndim,ip,im,i,j,k,icon
real(kind=8):: rmatrx(mmax),eigval(nmax),eigvec(nmax,nmax)
! For LAPACK
real(kind=8):: work(nmax*3)
character*1,parameter:: JOBZ='V',UPLO='U'
!=====
! parameter nyuryoku
!=====
!
! nyuryoku
!=====
write(7,*) 'number of lattice points ?'
read(*,*) ndim
!=====
! kensa
!=====
if((ndim.gt.nmax).or.(ndim.lt.1)) then
  write(*,*) 'error in ndim:',ndim
  stop
endif
!=====
! gyoretsu-yoso no keisan
!=====
! Make MATRIX A(j,i)=A(i*(i-1)/2+j)
do i=1,ndim
  if(i.eq.ndim) then
    ip=1
  else
    ip=i+1
  endif
  if(i.eq.1) then
    im=ndim
  else
    im=i-1
  endif
  do j=1,i
    k=(i*(i-1)/2)+j
    if (j.eq.i) then
      rmatrx(k)= 2.0d0
    else if ((j.eq.ip).or.(j.eq.im)) then
      rmatrx(k)=-1.0d0
    else
      rmatrx(k)= 0.0d0
    end if
  end do
end do
!=====
! gyoretsu no taikaku-ka (LAPACK)
!=====
! call DSPEV(JOBZ,UPLO,N,AP,W,Z,LDZ,WORK,INFO)
! INFO = 0 : Normal end
! > 0 : Not convergent
! < 0 : (=i) The value of the i-th argument is wrong
call DSPEV(JOBZ,UPLO,ndim,rmatrx,eigval,eigvec,nmax,work,icon)
!=====
! error check
!=====
if(icon.ne.0) then
  write(*,*) 'lapack(dspev) error: icon=', icon
endif
!=====
! shutsuryoku
!=====
do j=1,ndim
  if((eigval(j).lt.0.0d0).and.(eigval(j).gt.-eps)) then
    eigval(j)=0.0d0
  endif
  eigval(j)=dsqrt(eigval(j))
  write(*,*) eigval(j)
enddo
!
stop
end program phonon

```