5. 固有値問題

$\mathbf{A}\vec{x} = \lambda\vec{x}$ を満たすとき、$\vec{x}$ を **A** の固有ベクトル(eigenvector)、λ を固有値(eigenvalue)というのはどこかで勉強したことがあるだろう。行列の固有値問題(eigenvalue problem)は、偏微分方程式を解くために使われることもあり、化学、構造力学、経済学、社会学など応用範囲も広い。まずは慣れるために(思い出すために!?) 実際の例で考えよう。

5.1. とりあえず使ってみる

Octave には固有値問題を解く関数 **eig** が用意されているので、とりあえずこれを使ってみることにする。

(例題) 行列 **A** の固有値、固有ベクトルを求めよ。

$$\mathbf{A} = \begin{pmatrix} 2 & -1 & 3 \\ 0 & 1 & 2 \\ 3 & 0 & 2 \end{pmatrix}$$

```
> A=([2 -1 3; 0 1 2 ; 3 0 2]);
> [x lamb]=eig(A)
x=
 0.623645  0.655312  0.034646
 0.370341  0.484480 -0.946980
 0.688415 -0.579521 -0.319418
lamb=
 4.71774  0.00000  0.00000
 0.00000 -1.39234 0.00000
 0.00000  0.00000  1.67460
```

eig(A) を実行すると、列が固有ベクトルで構成される行列 **x** と、それぞれの固有値を対角要素に持つ対角行列 **lamb** が出力される。固有値だけをベクトルとして取り出したいなら、**lamb** の対角要素をベクトルにする関数 **diag** を使えばよい。

```
> diag(lamb)
ans=
 4.71774
-1.39234
 1.67460
```

検算のため、1 列目の固有ベクトルを使って **A***x* を求めると、

```
> A*X(:,1)
 2.9422
 1.7472
 3.2478
```

また、λx は、

```
> lamb(1,1)*X(:,1)
 2.9422
 1.7472
 3.2478
```

となって、一致するので、確かに **x** と **lamb** は、**A** の固有ベクトルと固有値になっていることが分かる。

5.2. シュレディンガー方程式を固有値問題として解く

弦楽器の弦を考える。両端を固定されているので、弓で弾く(初期位置と初速度を与える)と定在波が立ち、音が鳴る。節の数が少なければ低い音(低エネルギー)になるし、節の数が多くなれば高い音(高エネルギー)になる、なんていう話はどこかで勉強したことだろう。

次に量子力学の最初に良く出てくる「箱に閉じ込められた電子の状態」を考える。電子も波だから、電子の波動関数も、両端が固定された弦と同じような定在波が立つ、という話もどこかで勉強したことだろう。

量子力学で勉強する(時間依存のない)シュレディンガー方程式 $H|\phi\rangle = E|\phi\rangle$ は、まさに固有値問題だ。つまり、ハミルトニアン H が行列 $\mathbf{A}$、波動関数 $|\phi\rangle$ をベクトル $\vec{x}$ と考えれば、$\mathbf{A}\vec{x} = \lambda\vec{x}$ と同じということ。そこで、シュレディンガー方程式を固有値問題として解いてみる。ハミルトニアン H の固有値を求めると、固有エネルギー$E(=\lambda)$と、波動関数 $|\phi\rangle\,(=\vec{x})$が求まるというわけだ。

あるポテンシャルに閉じ込められた粒子、例えば電子を考えよう。簡単のため 1 次元で考える。ポテンシャルを v として、シュレディンガー方程式を具体的に書けば、

$$\left(-\frac{\hbar^2}{2m}\frac{d^2}{dx^2} + v\right)|\phi\rangle = \varepsilon|\phi\rangle$$

である。$\hbar^2/2m$ を基本単位とする単位系で書き直すと、

$$\left(-\frac{d^2}{dx^2} + V\right)|\phi\rangle = E|\phi\rangle$$

となる。V は $\hbar^2/2m$ を単位としたときのポテンシャルである。固有値問題として考えたいので、固有状態 $|\phi\rangle$ をあからさまなベクトルであらわそう。求めたい波動関数 $|\phi\rangle$ は数学的には無限

かつ連続量なのだが、これをある範囲だけ切り取り、それを n 個の要素に分割して、各点での振幅と位相を要素に持つ $n\times1$ のサイズのベクトルとして定義しようというわけだ。

$$|\phi\rangle=\boldsymbol{\varphi}=\begin{pmatrix}\phi_1\\ \phi_2\\ \vdots\\ \phi_n\end{pmatrix}$$

繰り返しになるが、要素 ϕ_k は複素数で、点 k における波動関数の振幅と位相である。では、ハミルトニアン d^2/dx^2 は行列でどのように表せるだろうか? まずは簡単のため d/dx を考える。微分は、行列では差分になるので、

$$\frac{d}{dx}|\phi\rangle\Rightarrow\frac{1}{\Delta}\begin{pmatrix}\phi_2-\phi_1\\ \phi_3-\phi_2\\ \phi_4-\phi_3\\ \\ \phi_n-\phi_{n-1}\end{pmatrix}$$

微分したことにより一行減り、サイズが $(n–1)\times1$ のベクトルになっていることに注意。(ただし、もともと波動関数は連続なので、このあたりは厳密に考えなくても本質的な問題はない) 簡単のため、刻み Δ を 1 とする単位系で考える。d/dx を行列表現にすれば、

$$\frac{d}{dx}=\begin{pmatrix}-1 & 1 & & & 0\\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ 0 & & & -1 & 1\end{pmatrix}$$

と書くことができる。つぎに、$-d^2/dx^2$ はどうするか。k 番目の要素の微分は、

$$\left.\frac{d^2\phi}{dx^2}\right|_k\Rightarrow\lim\frac{\frac{\phi_{k+1}-\phi_k}{\Delta}-\frac{\phi_k-\phi_{k-1}}{\Delta}}{\Delta}=\frac{\phi_{k+1}-2\phi_k-\phi_{k-1}}{\Delta^2}$$

とかけるので、

$$\frac{d^2\phi}{dx^2}=\begin{pmatrix}-2 & 1 & & & 0\\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1\\ 0 & & & 1 & -2\end{pmatrix}\begin{pmatrix}\phi_1\\ \phi_2\\ \\ \\ \phi_n\end{pmatrix}$$

のような三重対角行列を使って表現できる。このように 2 階微分の演算子は行列表現では 3 重対角行列になる。物理の問題では、2 階微分が頻繁に出てくるので、この形の三重対角行列も数値計算でよく登場する[1]。

ここで一つ問題になるのは、k=1 と k=n の状態をどう扱うかという点だ。これは境界条件と呼ばれ、数値計算では非常に重要な概念だ。例えば、本来 1 行目は、$\phi_2-2\phi_1+\phi_0$ とすべきだが、ϕ_0 は範囲外なので定義されていないし、n 行目は本来、$\phi_{n+1}-2\phi_n+\phi_{n-1}$ となるはずだが、ϕ_{n+1} は範囲外なので、定義されていない。波動関数が定義域の範囲外でほぼゼロとみなせられる場合は ϕ_0 も ϕ_{n+1} もゼロであるとして、上に挙げた三重対角行列で表せばよいだろう。しかし、端で波動関数がゼロにならないことがありうる場合は端の効果を取り入れないと正しい答えにはならない。そのようなときに頻繁に使われるのが、k=n+1 以降は、k=1 以降と同じものが繰り返すとみなす方法である。これは周期境界条件と呼ばれ、数値計算では非常によく使われる手法の一つである。具体的には、$\phi_2-2\phi_1+\phi_0$ を $\phi_2-2\phi_1+\phi_n$ で、$\phi_{n+1}-2\phi_n+\phi_{n-1}$ を $\phi_1-2\phi_n+\phi_{n-1}$ で、置き換える。行列で書けば、非対角要素の右上と左下を 0 ではなく 1 とする。

$$\frac{d^2\phi}{dx^2}=\begin{pmatrix}-2 & 1 & & & 1\\ 1 & -2 & 1 & & \\ & & \ddots & & \\ & & 1 & -2 & 1\\ 1 & & & 1 & -2\end{pmatrix}\begin{pmatrix}\phi_1\\ \phi_2\\ \\ \\ \phi_n\end{pmatrix}=\mathbf{D}$$

今回はこの周期境界条件を課して計算してみよう。そうすると、波動関数が端でゼロにならない場合でも計算できる。

V も同様に行列表示しよう。こちらは簡単で、対角要素に点 k でのポテンシャル V_k が並ぶだけだ。

$$\mathbf{V}=\begin{pmatrix}V_1 & & & 0\\ & V_2 & & \\ & & \ddots & \\ 0 & & & V_n\end{pmatrix}$$

したがって、シュレディンガー方程式は、

$$(\mathbf{D}+\mathbf{V})\boldsymbol{\varphi}=E\boldsymbol{\varphi}$$

という行列の固有値問題に帰着できた。

実際に計算してみるために、ポテンシャルを簡単な量子力学

[1]一般的に物理の問題では、3 重対角行列のように、スパース(疎)行列(ほとんどの行列要素が 0 の行列)を扱うことが多いため、一般的な行列ではなく、スパースな行列に最適化したしたアルゴリズムが用いられることが多い。

の教科書で見かける井戸型としよう(図 5.2.1)。計算する範囲は適当に決めていいが、今回は井戸幅が計算範囲の 1/5 になるように設定してみた。

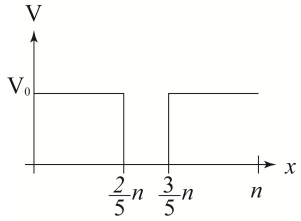

図 5.2.1

それでは実際にプログラムにしてみよう。あとからポテンシャルの形をいろいろ変えて試してみたいので、最低限のプログラムの変更で済むように、ポテンシャルを与える部分と、ハミルトニアンを与える部分は、メインプログラムとは別の関数として用意する。まず、井戸型ポテンシャルを与える関数(**well**)を作る。

```
% well.m
function v=well(x,v0);
[m n]=size(x);
a=round(2/5*n); b=round(3/5*n);
    %round は、ある数をもっとも近い整数へ丸める関数
v=v0.*ones(1,n);  %まず、すべての領域で v=v0 にする
for k=a:b         %井戸の中だけで for ループを回す
  v(k)=0;         %井戸の中は V=0 にする。
end
```

この **well** は、位置ベクトル **x** とポテンシャル障壁の高さ **v0** を入力すると、$1\times n$ のサイズのベクトル **v** を出力する関数である[2] 。

次に位置のベクトル **x** と、ポテンシャルのベクトル **v** を入力すると、ハミルトニアン **H** を出力する関数(**hamilton**)を作る。

```
%hamilton.m
function H=hamilton(x,v)
[m n]=size(x);
D=-2*eye(n)+circshift(eye(n),[-1 0]) ...
   +circshift(eye(n),[1 0]); %2 階微分を表す行列
V=diag(v);   %ベクトル v を対角要素にもつ行列 V を作る。
H=-D+V;  %ハミルトニアンを出力
```

4 行目で使われている **eye(n)** は **eye(n,n)** を省略したもので、サイズが $n\times n$ の単位行列を表す。また、**circshift** は、行列の要素を循環的にシフトする関数で、たとえば、**circshift(eye(n),[1 0])** なら、$n\times n$ の単位行列の行を一つプラス方向にずらし、n 行目を 1 行目に持ってきたような行列を出力する。係数をマイナスにすれば、循環の方向を逆にできる。

メインプログラム shroed.m は次のようになる。

```
%shroed.m
Clear; n=500;      %計算する領域のサイズを指定
x=1:n;             %空間をベクトルで定義
v0=0.05;           %ポテンシャルの高さを指定
v=well(x,v0);      %ポテンシャルを作る
H=hamilton(x,v);   %ハミルトニアンを作る
[fai,E]=eig(H);    %固有値問題を解く
En=diag(E);        %行列 E の対角要素(固有値)だけを、
                   %    ベクトルとして取り出す
Eoff=ones(n,1)*En';              %本文参照
plot(x,v,'k-',x, ...   %最後に...と書くと改行できる
  real(fai(:,1:8))/50+Eoff(:,1:8));
  %前文末に「...」と書くと改行しても次の行とつなげられる
xlabel('x'); ylabel('Energy/Wavefucntion');
legend('Potential');
axis([0 500 -0.001 0.056]);
```

なお、**Eoff** は、波動関数とポテンシャルを同時にプロットしたときに、波動関数を固有エネルギー分だけオフセットをくわえて見やすくするため行列で、物理的に本質的なものではない。なおこのオフセット行列は、下のような $n\times 1$ と $1\times n$ の行列のかけ算を使って作っている。

$$E_{\text{off}}=\begin{pmatrix}1\\1\\\vdots\\1\end{pmatrix}\begin{pmatrix}E_1 & E_2 & \cdots & E_n\end{pmatrix}=\begin{pmatrix}E_1 & E_2 & \cdots & E_n\\E_1 & E_2 & & E_n\\\vdots & \vdots & & \vdots\\E_1 & E_2 & \cdots & E_n\end{pmatrix}$$

plot の中で **fai** を **50** で割っているのも、単に波動関数を 50 分の 1 にスケールして、見やすくするためで、50 という数字に物理的な意味はない。

shroed.m を実行すると、量子化エネルギー**En** と波動関数 **fai** が同時に求まり、図 5.2.2 のように表示されるはず。**fai**, **E** はそれぞれ $n\times n$ のサイズの行列なので、固有値、固有ベクトルは実際には n 個ずつ出力されるのだが、計算結果から実際に井戸の中に閉じ込められている状態は、下から 8 番目ぐらいまでのようなので、**fai(:,1:8)** として、波動関数もエネル

[2] **n** を常に **5** の倍数に選べば、1/5 は割り切れるから、well.m の 4 行目の **round** は必要ない。

ギーの低い方から 8 個までだけを表示させている。実行結果が図 5.2.2。

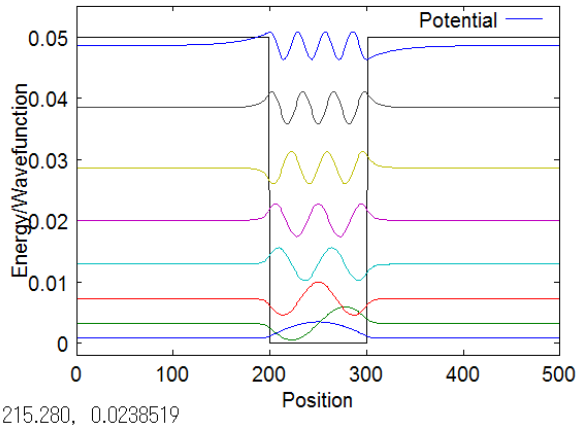

図 5.2.2 井戸型ポテンシャルに閉じ込められた粒子の波動関数

エネルギーが高い準位になればなるほど、実効的に障壁の高さが下がっていくので、障壁の中にも波動関数がしみ出している様子がよくわかる。

次に井戸型ポテンシャルの変わりに、調和振動子（2 次関数型)のポテンシャルを計算してみよう。調和振動子型のポテンシャルを作るために、以下の **harmonic.m** を作る。

```
% harmonic.m
function v=harmonic(x,v0);
[m n]=size(x); c=round(n/2); %c は中点
for k=1:n
  v(k)=20*(k-c)^2/c^2*v0;  %ポテンシャルを作る
          %係数の 20 は結果を見やすくするためのもの
end
```

ポテンシャルを変えたので、shroed.m の **v=well(x,v0)** の部分も、**v=harmonic(x,v0)** に置き換えておく(shroedHarmonic.m)。それ以外は同じなので、メインプログラムは省略する。

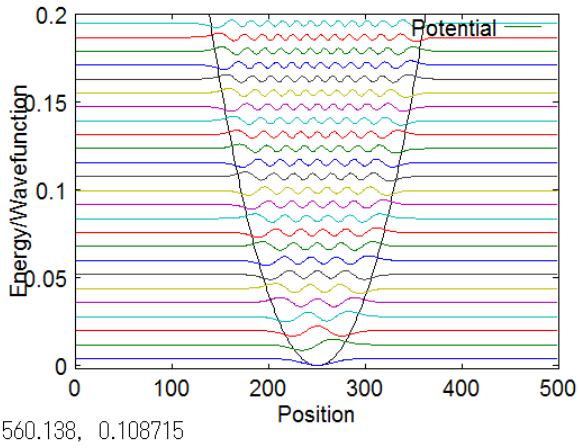

図 5.2.3 調和振動子の波動関数

この shroedHarmonic.m では、基底準位から 30 準位程度をプロットしている。調和振動子なので、エネルギー準位が等間隔になっている様子が確認できるだろう。

少し趣向を変えて、確率振幅をカラープロットするように、**plot** 以下の部分の表示方法に手を加えてみた。カラープロットには、**pcolor** を使う(shroedHarmonic2D.m)。**pcolor** の使い方は **help** 参照。

```
pcolor(x,En(1:30),fai(:,1:30)'.^2);
          %カラープロットに pcolor を使う
shading flat      %shading の設定  詳細は help を
xlabel('Position');ylabel('Energy');
title('Probability amplitude');
axis([100 400 0 0.22]);
```

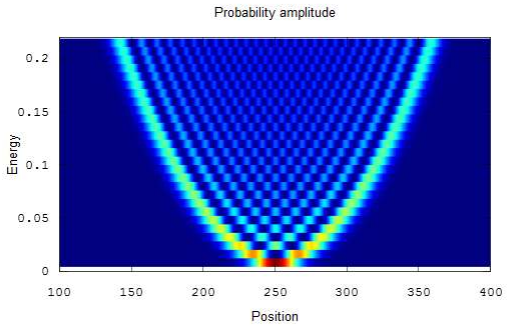

図 5.2.4 調和振動子の確率振幅

横軸が位置、縦軸が固有エネルギー、カラーマップが確率振幅を示している。プリントでは印刷の関係で色が分からないと思うが、実際に実行してみると、赤が振幅の大きいところ、青

が振幅の小さいところで表現される。

最後に、井戸型ポテンシャルが繰り返すような周期的ポテンシャルを考えてみよう。井戸の幅と障壁の幅を同じにして、それが無限に繰り返されるようなポテンシャル(multiwell.m)を考える。

```
% mutiwell.m
function v=multiwell(x,v0)
[m n]=size(x);
a=50; k=n/a;
for h=0:k-1
    v(h*a+1:(h+1/2)*a)=v0;
    v((h+1/2)*a+1:(h+1)*a)=0;
end
```

メインプログラムの方は、shroed.m の **v=well(x,v0)** の部分を、**v=multiwell(x,v0)** に変更し、プロットする部分も、**plot(x,v,'k-',x,-real(fai(:,1:30))/50+Eoff(:,1:30))** などと変更しておく。実行結果は図 5.2.4。

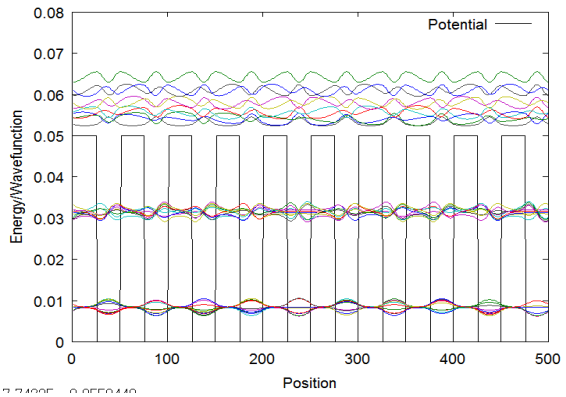

図 5.2.4 周期ポテンシャル中の波動関数

井戸型ポテンシャルも、調和振動子型ポテンシャルも、計算範囲外で波動関数がゼロと見なせる問題だったが、今回の周期ポテンシャルの場合は、計算範囲外で波動関数がゼロにはならない。しかし、周期境界条件を課しているので、図の左端の波動関数が、滑らかに右端の波動関数に繋がっている様子が確認できる。調和振動子では、準位の間隔は等間隔だったが、今回の結果は、縦軸の値で 0.01 と 0.03 のエネルギーのところに波動関数が密集してしていて、その間には準位が存在していないことが分かる。波動関数が密集しているところは、エネルギーバンド、存在していないところは、エネルギーギャップと呼ばれ、固体中の電子などを考える際にとても重要な概念だ。というのも、固体では電子に対してポテンシャルを作る原子が規則的に並んで周期構造を作っているからだ。

量子力学で勉強したように、調和振動子や、障壁が無限大の井戸型ポテンシャルの場合は、シュレディンガー方程式を解析的に解くことができるが、任意のポテンシャルでは、解析的に解けない。しかし数値計算を使えば、ポテンシャル形状も適当にデザインでき、波動関数も、量子化エネルギーも設計することができる。実際、半導体や金属で構造を作れば、井戸の幅を原子スケールで制御でき、材質を選べば障壁の高さが制御できる。そのような量子井戸でレーザーを作れば、固有エネルギー、つまりレーザー光の発光エネルギー=光の色も設計でき、実際にいろいろなところで使われている。

5.3. 固有値を求めるアルゴリズム(QR 法)

実際に、$\mathbf{A}\vec{x}=\lambda\vec{x}$ の固有値を数値計算で求めるにはどうすれば良いだろうか。線形代数では、固有値を求めるには、固有方程式 $\det(\lambda\mathbf{I}-\mathbf{A})=0$ を解けばいい、と勉強したのではないだろうか。4×4 行列までならそれでもいいが、それ以上では問題がある。というのも、ガロアの証明で有名な「5 次以上の方程式の解の公式は存在しない」というアーベル-ルフィニの定理によれば、5 次以上の行列についての固有方程式の解は一般的には求められない、つまり 5 次以上の一般的な行列に関する固有値の厳密解は求められないからだ。

固有値を求めるのに、べき乗法を習ったことがある人もいるかもしれない。べき乗法もよく使われる方法ではあるが、基本的には、絶対値最大の固有値と、それに対応する固有ベクトルを求める手法なので、すべての固有状態を求めたい場合には不向きである。

ここでは、現在、一般的な行列の固有値問題を数値計算で解くのによく使われる **QR 法**というアルゴリズムを紹介したい。これは、求めたい行列 **A** を、正規直交行列 **Q**（**Q** は、$\mathbf{QQ}^{\mathrm{T}}=\mathbf{Q}^{\mathrm{T}}\mathbf{Q}=\mathbf{I}$ を満たす） と、上三角行列 **R** に分解する **QR 分解**が基本になっている。上三角行列 **R** は、LU 分解のときの **U** と同じものなのだが、歴史的経緯から **U** ではなく **R** と呼ばれているらしい。あからさまに書くと、

$$\mathbf{A}=\mathbf{QR}=\begin{pmatrix} q_{11} & q_{12} & \cdots & q_{1n} \\ q_{21} & q_{22} & & q_{2n} \\ \vdots & & \ddots & \vdots \\ q_{n1} & q_{n2} & \cdots & q_{nn} \end{pmatrix}\begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & & r_{2n} \\ & & \ddots & \vdots \\ 0 & & & r_{nn} \end{pmatrix}$$

という感じ。QR 分解するアルゴリズムには、ハウスホルダー法、ギブンス法などがあるが、計算物理からだんだん離れた話題になるので、詳しくは省略。自作プログラムで固有値を求めたい人は QR 法が載っている線形代数の本を調べてみてほしい。

ここでは、QR 法の雰囲気を味わうために、何らかのアルゴリズムで、QR 分解できたとして、QR 法で固有値を求める方法を説明しよう。まず、求めたい行列 **A** を $\mathbf{A}=\mathbf{A}^{(0)}$ とする。

ステップ 1 $\mathbf{A}^{(0)}$を QR 分解し($=\mathbf{Q}^{(0)}\mathbf{R}^{(0)}$)、$\mathbf{Q}^{(0)}$と $\mathbf{R}^{(0)}$をひっくり返してかけ算し、これを $\mathbf{A}^{(1)}$ ($=\mathbf{R}^{(0)}\mathbf{Q}^{(0)}$)とする。

ステップ 2 $\mathbf{A}^{(1)}$ を QR 分解し($=\mathbf{Q}^{(1)}\mathbf{R}^{(1)}$)、$\mathbf{A}^{(2)}=\mathbf{R}^{(1)}\mathbf{Q}^{(1)}$ を求める。

これをひたすら繰り返す

・・・

ステップ k　$\mathbf{A}^{(k-1)}=\mathbf{Q}^{(k-1)}\mathbf{R}^{(k-1)}$ に分解し、$\mathbf{A}^{(k)}=\mathbf{R}^{(k-1)}\mathbf{Q}^{(k-1)}$ を求める。

k が十分大きければ(不思議なことに!?)、$\mathbf{A}^{(k)}$ の対角要素が固有値に収束していく。つまり、QR 分解して、ひっくり返して **R**×**Q** を求め、それをまた QR 分解して、**R**×**Q** を求め…を繰り返していくと、だんだんと固有値に近づいていくのだ。

本当にそうなるのか? Octave には、QR 分解する関数 **qr** が用意されているので、これを使って、**5.2.**で解いた調和振動子のハミルトニアンの固有値を QR 法で解いてみる。ハミルトニアンは、hamilton.m を流用して、**eig** と QR 法の比較をするプログラムを書く。計算を速く済ませるために、今回は行列のサイズを **n=200** と **5.2.**のときより小さめにした[3]。

```
%QRshroed.m
clear; n=200; x=1:n; v0=0.05;
v=harmonic(x,v0);
H=hamilton(x,v);
[fai,E]=eig(H); En=diag(E);  %eig を使って求める
Eqr=[];              %Eqr を空の行列にする
for m=1:4            %反復回数の違う計算を 4 回行う
   for k=1:40*m      %QR の反復を 40*m 回数繰り返す。
     [Q R]=qr(H);    %H を QR 分解
     H=R*Q;          %R*Q を求め、改めて H とする。
   end
   Em=diag(H);       %H の対角要素を取り出す
   Eqr=[Eqr Em];
  %一つの前の Eqr に Em を連結して新たな Eqr とする
end
Eqr=sort(Eqr);       %固有値を小さい順に並べる
plot(x,En,'ko',x,Eqr);
xlabel('Index'); ylabel('Energy');
legend('eig','QR40','QR80','QR120', ...
    'QR160','Location','NorthWest')  %脚注参照
```

QRshroed.m を実行すると、図 5.3.1 が表示される[4]。横軸が固有値を小さい順に並べたときの番号、縦軸が固有エネルギー。黒丸が **eig** で求めた固有値、それ以外が QR 法で求めた固有値だ。QR の反復回数が 40 回程度では、固有値が 10 準位ぐらいまでしか正しく得られていないが、反復回数が増えるごとに、**eig** で求めた固有値に近づいていく様子が分かるだろう。

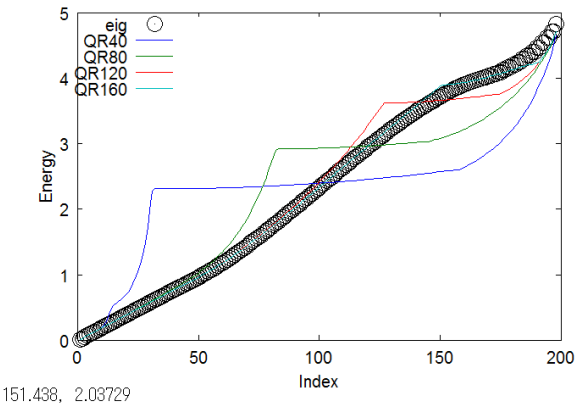

図 5.3.1　QR 法の反復回数を変えた場合の固有エネルギー

図 5.3.1 の 60 準位までの部分を拡大したものが、図 5.3.2。準位間隔が等間隔であるという調和振動の特徴を反映して、準

[3] なお、この QRshroed.m は、単に QR 法の雰囲気を味わうためだけのものなので、高速化も最適化もされていない。また、固有値を求めたい行列が三重対角行列になっているので、この問題を解くだけなら、わざわざ QR 法をつかうまでもない。

[4] デフォルトの凡例(legend)の位置では、プロットデータと、かぶってしまうので、**'Location','NorthWest'**で、凡例を左上に表示させている。

位の番号 Index に対して、エネルギーが直線になっている様子が分かる。図 5.3.1 で、70 準位あたりを超えると、直線から外れてしまうので、誤差が発生していると考えられる。

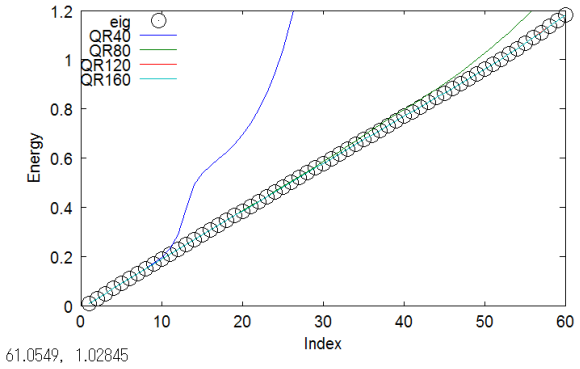

図 5.3.2 5.3.1 の拡大図

一般的には、ヘッセンベルグ行列(上三角行列の、対角要素より一つ下まで 0 でない行列)に変換してから、反復を実行させる方が高速であることが知られている。興味があれば、自分で QR 法のプログラムを作ってみてほしい。

ちなみに、QR 法は 1961 年に開発された方法だが、固有値を求めるアルゴリズムには、コンピュータが作られるずっと以前の 1846 年に開発された Jacobi 法というものもある。通常の Jacobi 法では、だいたい 10×10 行列を越えると計算が遅くなってしまうのだが、最近の研究では、Jacobi 法は QR 法よりは遅いが、いろいろ工夫すれば精度は QR 法より高くなるという報告もあるらしいので、興味があれば Jacobi 法もトライしてみてはどうだろうか。