

# 周期ポテンシャルにおける波束の時間発展

前田大輝

2017 年 8 月 16 日

## 概要

周期ポテンシャルは一定の周期構造を持つ分子の並びをモデル化したものであり、中でも最も簡単な **Kronig-Penny** モデルでは定常状態のエネルギーバンド構造を説明することができる。一方で時間に依存した現象では、波束が時間の 2 乗よりも早く緩和することが知られており、**Hyperdiffusion** と呼ばれている。このような、周期ポテンシャル中の粒子が真空中とは違った振る舞いをするに関する知見は、固体物理やナノリソグラフィなどの分野に応用できる場合がある。そこで本研究は、特定の (半) 周期ポテンシャルにおける波束がどのように緩和していくのかを解析することを目的として、主にコンピュータにおける数値計算的な手法を用いている。その結果 **Kronig-Penny** 型の周期ポテンシャルに入射する波束は、ステップ型のポテンシャルに入射した場合と比べて、反射波に遅れが生じることを発見した。

# 目次

第 1 章	研究の目的	2
1.1	波束の時間発展 . . . . .	2
1.2	Kronig-Penny モデル . . . . .	3
第 2 章	数値計算	5
2.1	Euler 法 . . . . .	5
2.2	数値計算法決定における指標 . . . . .	7
2.3	Fourie 変換法 . . . . .	10
2.4	有限要素法 . . . . .	10
第 3 章	先行研究	15
3.1	Bloch の定理と Kronig-Penny モデルの定常解 . . . . .	15
3.2	Zenjun の超拡散 . . . . .	16
第 4 章	結果	17
第 5 章	考察	18
第 6 章	schrpy について	19

# 第1章

## 研究の目的

物質中の電子は真空中と比べ、特異な振る舞いをすることが知られている。その代表例が結晶内におけるエネルギーバンド構造であり、結晶中の電子は連続したエネルギースペクトルを持つことができない。バンドギャップにあたるエネルギー状態をとる電子は結晶境界で全反射するか、ギャップ分のエネルギーを光子として放出し、バンドに適合したエネルギーを持って物質中に侵入する。

この物理現象について、定常状態での解析はなされているが、時間依存の詳細についてはまだ未解明である。近年は光格子時計により時間分解能が飛躍的に向上しているため、現在の「一瞬にしてバンド構造が作られる」という近似が成立しなくなる可能性がある。そこで、どの程度の時間分解能ならばその近似が成立するかの目安を求めること重要性が増している。

### 1.1 波束の時間発展

波束とは局在化した波動のことで、量子波束は古典的な質点に対応する。例えば、十分に細い電子パルスは古典的に点粒子とみなすことができる。分解能  $d$  と位置不確定性  $\delta x$  との関係 Eq. (1.1) が、波束と質点を対応させる古典近似の成立条件となる。

古典近似成立条件

$$\frac{\delta x}{d} \ll 1 \quad (1.1)$$

しかし、古典近似は時間とともに成立しなくなる。これは、自由量子波束が本質的に時間依存した現象であることに由来している。つまり、静止した自由粒子は古典的には定常的で時間依存しないが、一般に Schrödinger 方程式における静止<sup>\*1</sup>自由波束は時間とともに緩和するため、 $\delta x$  は時間とともに増加し、Eq. (1.1) が成立しなくなる。

$\delta x$  の目安として標準偏差  $\sigma(t)$  を使うと、Gauss 型波束の場合、 $\sigma(t)$  は次の Eq. (1.2) で表せる。

$$\sigma(t) = \sigma_0 + \alpha t \quad (1.2)$$

---

<sup>\*1</sup> 重心が移動しない

$\sigma_0$  は初期状態の標準偏差で、 $\alpha$  は質量と  $\sigma_0$  に依存する正の定数を意味している。Eq. (1.2) によって、一様ポテンシャル中あるいは真空状態での古典近似が成立する条件 Eq. (1.1) の陽に時間依存した形を得ることができる。

$$\frac{\sigma_0 + \alpha t}{d} \ll 1 \quad (1.3)$$

$t = 0$  のときには Eq. (1.1) が成立しているとして時間の一次についてのみとると、Eq. (1.1) が成立する限界時間  $\delta t_{lim}$  はおよそ 1.4 で表される。

$$\begin{aligned} \frac{\alpha}{d} \delta t_{lim} &\sim 1 \\ \delta t_{lim} &\sim \frac{d}{\alpha} \end{aligned} \quad (1.4)$$

しかし、Eq. (1.2) は前提として力が加わっていない状態を仮定している。よって、この式から直ちに一般の場合について古典近似が成立する時間の目安を見いだすことはできない。そこで、真空以外の場合、中でも特に物質中のポテンシャルについて、この目安を見出すことには一定の意味がある。

波束についてのまとめ

- 波束は古典近似で質点に対応する
- 波束は時間発展とともに緩和する
- 物質中での緩和速度についてはよくわかっていない

## 1.2 Kronig-Penny モデル

結晶中でエネルギー固有状態にある電子は取り得るエネルギーに飛躍が現れる。この構造のことをエネルギーバンド構造という。エネルギーバンド構造は結晶の並進対称性と関連がある。

Bloch は、結晶原子が周期的に並んでいる構造を、ポテンシャルが周期的に並んでいる構造と対応させて、周期ポテンシャル中の波動関数が従う一般的な形式 Eq. (1.5) を求めた。

Bloch の定理

$$\phi(x) = u(x)e^{-Kx} \quad (1.5)$$

$K$  は逆格子ベクトルと呼ばれる結晶の周期に依存する定数、 $u(x)$  は結晶と同じ周期をもつ関数で結晶構造に依存する。周期ポテンシャル中の波動関数が Eq. (1.5) に従うという定理を Bloch の定理と呼ぶ。

この定理を使って、Kronig と Penny はポテンシャルが Eq. (1.6) のように表される場合に、エネルギーバンド構造が形成されることを説明した。

Kronig-Penny ポテンシャル

$$V_{KP}(x) = \begin{cases} V_0 & (0 \leq x \leq a) \\ 0 & (a \leq x \leq d) \\ V_{KP}(x-d) & (otherwise) \end{cases} \quad (1.6)$$

Eq. (1.6) は箱型ポテンシャルが周期的に並んだ構造を表している。これは周期ポテンシャルの中でも最も単純な具体例の一つで、Kronig-Penny ポテンシャルと呼ばれる。結晶を Kronig-Penny ポテンシャルで表現するというモデルは簡単な構造ながらエネルギーバンド構造を説明することができる点で有効であり、Kronig-Penny モデルと呼ばれる。

また、定常状態についての構造は古くから知られている Kronig-Penny ポテンシャルだが、非定常状態は Bloch の定理が成立しないため、波動関数が Eq. (1.5) で表されることはない。これに関しては、Zejun<sup>\*2</sup>によって Gauss 型波束の緩和についての研究<sup>\*2</sup>が行われている。この結果、波束の標準偏差が時間に比例するよりも早く大きくなる現象が見出され、Heyperdiffusion が起こることを示した。

このように、物質中の波束や真空中の波束についてはある程度の知見が得られているが、真空中から物質中に入射する波束については未だ知見が得られていない。この場合、暗黙に真空中では波束とみなし、結晶中でエネルギーバンド状態であるとみなす近似<sup>\*3</sup>が行われる。しかし、波束は非定常状態であり、エネルギーバンド状態は定常状態であるため、厳密には2つの状態間を一瞬で移り変わるわけではない。時間分解能が高まった現代では、結晶に入射した波束が拡散しながらエネルギーバンドを作る過程が観測にかかる可能性がある。よって、エネルギーバンド構造が作られる時間の目安を与えることには一定の意味がある。

そこで、エネルギーバンドの構成時間を調べるために、最も簡単な周期ポテンシャルである Kronig-Penny モデルを用いて解析を行う。また Kronig-Penny モデル自体がナノリソグラフィーの素子や半導体超格子に対する良い近似として成立しているため、それらの性質を知る上で、この研究は実際的な研究となる。

#### Kronig-Penny モデルについてのまとめ

- 物質中の電子は定常状態でエネルギーバンド構造を作る
- Kronig-Penny モデルはエネルギーバンド構造を説明できる
- 真空中から Kronig-Penny ポテンシャルに入射する波束の緩和についてはよくわかっていない

<sup>\*2</sup> Zejun のポテンシャルは  $x = 0$  付近だけ井戸の間隔が広がっている

<sup>\*3</sup> 古典的には、電子が物質に侵入すると物質全体の電位がほとんど一瞬で上昇することに対応している。

## 第2章

# 数値計算

Kronig-Penny ポテンシャル中の Gauss 型波束の時間発展を解析的に得ることは非常に困難を伴う。そのため、近似的な方法を用いる必要がある。しかし、Schrödinger 方程式は二階の偏微分方程式であり、計算機で解くには少し工夫がいる。そこで、複数の方法を検討し、妥当な方法を提案する。

### 2.1 Euler 法

この節は主に学部生に対して、常微分方程式を数値的に解くことの雰囲気を知ってもらうことを目的としている。形式的な話が多いが、それはどんな方程式にも対応できるように問題を抽象化しているからで、” どんな難しい方程式も解くことができる ” という数値計算の力強さの源になっている。枝葉の些事は気にせず、大雑把に何をやっているのかを知ってほしい。

一階の常微分方程式は一般的に Eq. (2.1)、Eq. (2.2) の形で表すことができる。

$$\frac{dy}{dt} = f(y, t) \quad (2.1)$$

$$y(t_0) = y_0 \quad (2.2)$$

ここで  $y_0$  は初期条件として与えられる既知の値とする。Eq. (2.1) の右辺を積分し、積分定数として Eq. (2.2) を使えば、この方程式の解が得られる。しかし、計算機で数値的に解く際にこの方法は使うことができない。なぜなら、計算機は有限回の加減乗除<sup>\*1</sup>しか扱うことができないからである。形式的な微積分操作をプログラムしてやることで対応可能な場合もあるが、その場合は人間が手で解くのとあまり変わらない。今考えたいのは、解析的に解くことができない場合<sup>\*2</sup>である。

解析的手段が使えない場合、微分方程式の解析解を得る手段がないため、問題の設定を変更する。解析解を得ることは諦め、解の条件を少し緩めて、任意の精度で真の値に近い値を得ることができればよいことにする。そこで、無限回の計算を有限回で打ち切るような近似を行い、計算機でも可能にする。

近似方法は多岐にわたるが、数値計算の数ある方法の中でも、この節の目的達成に最も適した方法として Euler 法を取り上げる。あまり実用的な方法ではないが、数値計算特有の難しさや、力強さを掴んでもらうた

---

<sup>\*1</sup> 厳密に言うと二進数のビットシフト、AND,OR,NOT 演算の組み合わせ

<sup>\*2</sup> 例えば  $f(y, t) = y^t$  のとき

めには十分だと考えている。なぜなら最も基礎的な<sup>\*3</sup>計算方法だからである。この方法を発展させた方法は Runge-Kutta 法と呼ばれ、最もメジャーな数値計算法のクラスを作っている。

Euler 法は、 $y(t)$  を  $t = 0$  の周りで Taylor 展開<sup>\*4</sup>することで得られる。

$$y(t) = y(t_0) + \left. \frac{dy}{dt} \right|_{t=t_0} (t - t_0) + \frac{1}{2} \left. \frac{d^2y}{dt^2} \right|_{t=t_0} (t - t_0)^2 + \dots \quad (2.3)$$

$t$  についての一次の項は Eq. (2.1) の  $f(y, t)$  を使い、ゼロ次の項は Eq. (2.2) を使って表すことができる。そして、二次以降の項は  $t = t_0$  の近傍では無視できるとすると、 $h \ll 1$  において Eq. (2.4) が成立する。

$$y(t_0 + h) \simeq y_0 + f(y_0, 0)h = y_1 \quad (2.4)$$

これによって、既知の値  $y_0$  から、未知の値  $y(t_0 + h) = y_1$  を得ることができた。次に  $t = t_0 + h = t_1$  の周りで Taylor 展開を行うと、Eq. (2.3) とよく似た Eq. (2.5) を得ることができる。

$$y(t) = y(t_1) + \left. \frac{dy}{dt} \right|_{t=t_1} (t - t_1) + \frac{1}{2} \left. \frac{d^2y}{dt^2} \right|_{t=t_1} (t - t_1)^2 + \dots \quad (2.5)$$

そして、Eq. (2.4) と同様にすると Eq. (2.6) が得られる。

$$y(t_1 + h) \simeq y_1 + f(y_1, t_1)h = y_2 \quad (2.6)$$

これらの操作を次々に繰り返すことで、 $t = t_n$  における  $y(t)$  の値を求めることができる。ここまでの議論で十分だが、一応常微分方程式 Eq. (2.1)、Eq. (2.2) における Euler 法の定義を書いておくと次のようになる。

Euler 法

$$y_{n+1} = y_n + f(y_n, t_n)h \quad (2.7)$$

Eq. (2.7) では、 $f(y_n, t_n)$  の計算を除くと 1 回の掛け算と 1 回の足し算しか使っていないため、有限回の加減乗除しかできない計算機でも計算することができる。この方法を使えば、どんな方程式でも理論上、計算機で解くことができる。

しかし、 $h$  の値によっては Taylor 展開の高次の項を無視することができないため、計算誤差が発生する。1 ステップでの誤差の大きさは、Euler 法では概ね  $h$  に比例する。そのため、有効数字を 1 桁上げるためには、ステップ幅  $h$  を 10 分の 1 にしなければならず、10 倍のステップ数が必要になってしまう。”理論上は”どんな方程式も任意の精度で解くことができる<sup>\*5</sup>が、それは自分が生きている間とは限らない。

Euler 法のまとめ

- 数値計算は近似的に解を求める
- どんな方程式も解くことができる
- 生きている間に計算が終わるとは限らない

<sup>\*3</sup> ”基礎的であること”と”簡単なこと”は同じではない。基礎を生み出すことは、応用することとは違った難しさがある。

<sup>\*4</sup> Taylor 展開できる関数には  $C^\infty$  級（定義域のすべての点で無限回微分可能）という制約が付いているが、物理で使う関数はだいたい微分可能なのであまり心配はいらない。

<sup>\*5</sup> 実際は丸め誤差の影響で、任意の精度で求めることはできない。



## 2.2 数値計算法決定における指標

数値計算の方法はバラエティが豊かだが、基本的に安定性、計算量の2つが重要な指標となる。これらは数値計算特有の困難を数値化したもので、安定性は計算誤差、計算量は資源の消費量を表す指標となる。

計算誤差の具体例として、丸め誤差と打ち切り誤差を挙げる\*6。

計算機は基本的に実数を直接扱うことができないため、一定の精度で近似をして計算する。このような計算機の精度による誤差を丸め誤差という。具体的には、割り算、平方根が有限桁で表せない場合に有限桁の”小数”で近似を行うため、丸め誤差が発生する。ここで言う”小数”は二進数における小数であって、十進数における小数ではない。例えば、十進数における0.1を入力した場合、計算機の内部で丸め誤差が発生する。なぜなら、0.1は二進数においてはEq. (2.8)で表せるように無限小数となるためである。

$$\begin{aligned} 0.1 \quad (\text{on decimal}) &= 2^{-4} + 2^{-5} + 2^{-8} + 2^{-9} + 2^{-12} + \dots \quad (\text{on decimal}) \\ &= 0.00011001100110011\dots \quad (\text{on binary}) \\ &= 0.0\dot{0}0\dot{1}1 \quad (\text{on binary}) \end{aligned} \quad (2.8)$$

よって、計算機で数値計算を行う際には、丸め誤差は必然的に発生する。

また、有限時間で計算を終えるためには、無限数列を有限の項で打ち切らなくてはならない。例えば、Eq. (2.9)を使って、Napier 数  $e$  を計算することができる。

$$e = \sum_{n=0}^{\infty} \frac{1}{n!} \quad (2.9)$$

しかし、計算機では  $n$  について無限に足し合わせることはできない。よって、有限項 (例えば 1000 項) で打ち切る近似を行い、Eq. (2.10) を計算する。

$$\begin{aligned} e &= \sum_{n=0}^{\infty} \frac{1}{n!} \\ &\simeq \sum_{n=0}^{1000} \frac{1}{n!} \end{aligned} \quad (2.10)$$

この計算の一行目から二行目に移るところでは高次の項の寄与\*7が打ち切られているため、真の値よりも小さくなってしまう。このように打ち切り誤差は必然的に発生する。

計算誤差は避けようがないが、制御することはできる。常微分方程式の数値解法では、積分範囲を有限区間に区切って積分を行うが、このステップ幅を小さくすれば精度は高くなる\*8。そこで、どの程度のステップ幅ならば許容できるのかが重要になってくる。これを議論するときに必要なものが安定性の概念である。

常微分方程式 Eq. (2.11)Eq. (2.12) を一定の刻み  $h$  で数値積分するとき、解が必ず真の値に収束する\*9 ような

\*6 他に数値計算に影響を与える誤差として連続量を離散量で表す際に発生する離散化誤差がある。

\*7 もちろん、1000 項ではなく 5000 項の方が高い精度を期待できるが、計算に時間がかかるため、いずれにしても Eq. (2.9) を直接計算する方法では無限の精度の計算を有限の時間で行うことはできない。

\*8 ステップ数が増えると、丸め誤差の発生が多くなるため、いくらでも精度が高くなるわけではない。よって、計算量を無視しても、やはりアルゴリズムによって適度なステップ幅が存在する。

\*9 丸め誤差を無視する

$\lambda h$  が存在する場合がある。この  $\lambda h$  が取り得る領域を絶対安定領域という。

$$\frac{df}{dx} = \lambda f(x) \quad (2.11)$$

$$f(0) = y_0 \quad (2.12)$$

安定性は、この絶対安定領域の広さによって議論することができる。絶対安定領域の広さについて経験的に知られていることとして、陰的公式<sup>\*10</sup>のほうが広く、陽的公式<sup>\*11</sup>のほうが狭い。絶対安定領域はアルゴリズム固有のもので、アルゴリズムの良し悪しを決める際の指標となる。また、どのような  $\lambda h < 0$  をとっても解が真の値に収束することを A 安定というが、A 安定だからといって実用的な時間でよい解が得られるとは限らない。<sup>\*12</sup> 収束性や計算の容易さは別の概念で測る必要がある。

ただし、絶対安定領域が狭いアルゴリズムは、高い精度で計算しなければ計算が安定しないため、自分のほしい精度と絶対安定領域との2つを管理しなければならない分、難易度は上がる。

計算量には、空間的計算資源（メモリ消費量）と時間的計算資源（計算時間）の消費量という2つの意味を持つが、2つの量はだまかに比例する傾向にある<sup>\*13</sup>ため、今回は特に計算時間のことを指して「計算量」という言葉を用いる。計算量はプログラムに入力される情報の量  $n$  に依存するが、必ずしも入力情報量が二倍になると、計算量  $\tau$  も二倍になるとは限らない。入力情報に偏りがある場合<sup>\*14</sup>は  $\tau$  を小さく抑えることができる。また、入力情報のすべての組み合わせについて計算するようなアルゴリズムの場合は、 $n$  が増えると爆発的に  $\tau$  が増える。そこで、 $\tau$  の目安として Landau の記号 Eq. (2.13) を用いることが多い。

Landau の記号

$$\tau(n) = O(f(n)) \quad (2.13)$$

この記号は Eq. (2.14) が収束することと等価である。

$$\lim_{n \rightarrow \infty} \left| \frac{\tau(n)}{f(n)} \right| \quad (2.14)$$

例えば、この記号を使うと  $\cos(x)$  は Taylor 展開を使って次のように書ける。

$$\cos(x) = 1 + O(x^2) \quad (2.15)$$

この記号を使うと具体的な関数はわからなくても、大まかな傾向を見取ることができる。計算量もアルゴリズム固有の値をとり、アルゴリズムの良し悪しを決める指標となる。

<sup>\*10</sup>  $t = t_n$  における  $f(t_n)$  を計算する際に  $f(t_n)$  の値を必要とする。つまり、すべての  $n$  についての連立方程式を解くことで解を得る方法。

<sup>\*11</sup> 陰的公式とは逆に、 $t = t_n$  における  $f(t_n)$  を計算する際に  $f(t_n)$  の値を必要としない。このおかげで、逐次代入を行うだけで解が得られる。Euler 法はこちら側の方法。

<sup>\*12</sup> 例えば Euler 法を時間について逆向きにした陰的 Euler 法は A 安定だが、局所的な誤差は Euler 法とほぼ同程度で、ろくな精度が得られない。

絶対安定領域とは、長時間計算を行っても数値解が真の値から大きくずれることはないステップ幅の領域を意味している。

<sup>\*13</sup> もちろん、メモリを大量に消費して計算時間を短くしたり、その逆をしたりすることもあるが、大規模な計算に大量の資源が必要なことには変わりはない。

<sup>\*14</sup> 極端な例では、入力がすべて "1" だとわかっている場合、入力情報を愚直に

`input(1, 1, 1, 1, 1, 1, 1, 1, 1, 1)`

と記憶するのではなく、

`input(1).times(10)`

と回数だけ記憶すれば、1000 個の入力だろうが、10000 個の入力だろうが、定数のオーダーに圧縮することができる。

基本的に計算量が少なく絶対安定領域が広いアルゴリズムが求められるが、両方が優れている方法は存在しないため、適度なものを選ぶしかない。

数値計算の指標

- 絶対安定領域：計算が収束する誤差の範囲
- 計算量：計算にかかる時間
- 絶対安定領域と計算量はトレードオフ

## 2.3 Fourier 変換法

数値的に Schrödinger 方程式を解く方法には大きく FFT 法と有限要素法が存在する。Fourier 変換法はエネルギー固有状態  $\phi_n(x)$  の重ね合わせによって系の初期状態  $\psi_0(x) = \psi(x, t)|_{t=0}$  を構成する。

$\phi_n(x)$  による  $\psi_0(x)$  の内積<sup>\*15</sup>が収束すると仮定する<sup>\*16</sup>と、Eq. (2.16) によって Fourier 係数が得られる。

$$\int_c \phi_n^*(x) \psi_0(x) dx = A_n \quad (2.16)$$

よって Fourier 逆変換によって  $\psi_0$  と  $\phi_n$  の関係が Eq. (2.17) 得られる。

$$\psi(x, t) \Big|_{t=0} = \sum_n^\infty A_n \phi_n(x) \quad (2.17)$$

時間発展を計算したければ、角振動数  $\omega_n$  をエネルギー  $\varepsilon_n$  に応じて  $\varepsilon_n/\hbar$  で与えれば良い。

Fourier 変換法

$$\psi(x, t) = \sum_n^N A_n \phi_n(x) e^{\frac{\varepsilon_n}{i\hbar} t} \quad (2.18)$$

この方法では、 $\phi_n$  がわかってしまえば、 $O(1)$  で計算することができる。

計算誤差は主に Eq. (2.18) の和を有限項で打ち切ることによって発生する。この相対誤差は低周波領域で顕著なため、 $t$  が大きな時間発展を計算する際は誤差が大きくなるが、小さい場合にはかなり効率がいい。また、あらゆる時刻の時間発展を計算する場合でも定数時間で計算が終了するという点でも非常に性質が良い。

ただし、Kronig-Penny ポテンシャルにおける定常状態の Schrödinger 方程式は、Dirac の櫛と呼ばれる極限を除いて、解析的に解を得ることに計算精度の面で困難がある。そこで、本研究では解析的に定常状態を得ることは避け、数値的に定常状態を得るだけに留めることにする。

Fourier 変換法についてのまとめ

- 定常状態を重ねあわせて計算する
- $O(1)$  で計算できる (すごい!)
- 定常状態をどうやって求めるかはわからない
- Kronig-Penny モデルでの定常状態は実質的に求められない

## 2.4 有限要素法

もうひとつの数値解法である有限要素法は積分する範囲を有限の区間に分割し、分割した区間内での解は適当な可積分関数<sup>\*17</sup>で表せると近似することで、数値解を得る方法である。

<sup>\*15</sup> 単に片方の複素共役をとって積分しているだけなのでビビらないで欲しい。

<sup>\*16</sup>  $L^2$  とよばれる関数空間の元だという仮定を置いている。

物理的に言うと、Eq. (2.16) は

”  $\psi_0$  の状態にある粒子の状態を観測したとき、その状態が  $\phi_n$  である確率が  $C_n$  である ”

ということを意味している。観測をすることができるということを仮定しているだけなので、あまり心配しないでもいい。

<sup>\*17</sup> 基本的に多項式だと考えて良い

この方法のバリエーションとして Runge-Kutta 法と線形多段法が存在する。

## 2.4.1 Runge-Kutta 法

Runge-Kutta 法は積分したい点の近傍で線形近似を行い<sup>\*18</sup>、補正を繰り返しながら何度も計算し直す方法で、Euler 法はこのクラスに属する。

例えば、二次関数  $y(t) = \alpha t^2 + \beta t + \gamma$  を  $t_a$  と  $t_b$  の地点で正しくなるよう線形近似する場合を考える。傾きを直接計算すると Eq. (2.19) のようになるが、これは  $t = (t_b + t_a)/2$ 、つまり二点の中間点での傾き Eq. (2.20) に等しい。

$$\frac{y(t_b) - y(t_a)}{t_b - t_a} = \frac{(\alpha t_b^2 + \beta t_b + \gamma) - (\alpha t_a^2 + \beta t_a + \gamma)}{t_b - t_a}$$

$$= \alpha(t_b + t_a) + \beta \quad (2.19)$$

$$= 2\alpha \left( \frac{t_a + t_b}{2} \right) + \beta \quad (2.20)$$

よって、Eq. (2.1) での  $y(t)$  について二次の項まで正しい式を作るためには区間の端点ではなく、中間地点での傾きを採用すれば良い。区間の中間地点を求めるときは、Euler 法を用いる。これが 2 階 2 次の陽的 Runge-Kutta 法<sup>\*19</sup>の一種である。

2 階 2 次の Runge-Kutta 法

$$y_{n+1} = y_n + k_2 h \quad (2.21)$$

$$k_2 = f(y_n + k_1 h/2, t_n + h/2) \quad (2.22)$$

$$k_1 = f(y_n, t_n) \quad (2.23)$$

Runge-Kutta の陰的方法には任意の次数で A 安定の方法が存在するが、実用することは難しい。陰的方法を解くには一般に非線形連立方程式を解かなくてはならない。これを解くことは一階常微分方程式を解くことと同等か、それ以上に難しいためである。

一方、Runge-Kutta の陽的方法は逐次代入によって解くことができるため非常に計算が容易<sup>\*20</sup>である。また、以前のステップの情報を使わないため、ステップ幅を自由に変えることができるのも利点のひとつである。しかし、階数をあげても次数はあまり上がらない。また、絶対安定領域が非常に狭いか、存在しない。

## 2.4.2 線形多段法

一方、線形多段法は積分したい関数を Ralston 補間などによってべき級数に近似する方法で、予測子修正子 Adams 法や Adams-Bashforth 法などが知られている。

Euler 法は過去の値  $y_n$  ひとつから未来の値  $y_{n+1}$  を予測するが、もうひとつ過去の値  $y_{n-1}$  覚えておけば改良できるのではないかという発想で線形多段法は作られている。 $y_n, y_{n-1}$  が既知だとすると、 $y_n$  の周りで

<sup>\*18</sup> つまり一階微分は定数だと近似する。

<sup>\*19</sup> 中間地点を求めるための Euler 法 (1 回め) と、中間地点の傾きを使った線形近似 (2 回め) とで、合計 2 回の線形近似を行う方法なので”2 階” (誤字ではない) の Runge-Kutta 法といい、同時に 2 次の Taylor 展開までと等しいので 2 次の Runge-Kutta 法という。今回は階数と次数が一致したが、一般には階数のほうが大きくなる。

また、あまり聞かないが、Euler 法は 1 階 1 次の Runge-Kutta 法といえる。

<sup>\*20</sup> トライアンドエラーの数をこなせるという意味で非常に重要な性質

$f(y, t)$  は Eq. (2.24) のように線形近似できる。

$$\begin{aligned} f(y, t) &\simeq \frac{t - t_{n-1}}{t_n - t_{n-1}} f(y_n, t_n) + \frac{t - t_n}{t_{n-1} - t_n} f(y_{n-1}, t_{n-1}) \\ &= \frac{f(y_n, t_n)}{h} (t - t_{n-1}) - \frac{f(y_{n-1}, t_{n-1})}{h} (t - t_n) \end{aligned} \quad (2.24)$$

$t_n - t_{n-1}$  はステップ幅に等しいので  $h$  と置いた。

Eq. (2.24) のように多項式補間する方法を **Ragrange 補間**<sup>\*21</sup> という。後に利用するので、一般形を書いておく。

Ragrange 補間

任意の  $N$  点  $(y_0, t_0), (y_1, t_1), \dots, (y_i, t_i), \dots, (y_N, t_N)$  を通る  $N - 1$  次多項式は次のように表される。

$$P(t) = \sum_{i=1}^N \frac{(t - t_1)(t - t_2) \dots (t - t_{i-1})(t - t_{i+1}) \dots (t - t_N)}{(t_i - t_1)(t_i - t_2) \dots (t_i - t_{i-1})(t_i - t_{i+1}) \dots (t_i - t_N)} y_i \quad (2.25)$$

Eq. (2.1) の  $f(y, t)$  は一般に可積分だとはいえないが、過去の  $f(y, t)$  を使って多項式近似した Eq. (2.24) を使

<sup>\*21</sup> 複雑に見えるが、分解すると非常に簡単な原理で作られている。

- 任意の  $N$  点  $(y_0, t_0), (y_1, t_1), \dots, (y_i, t_i), \dots, (y_N, t_N)$  を通る多項式  
 $\hookrightarrow t_i$  以外の  $t = t_1, t_2, \dots, t_N$  では 0 を通り、 $(y_i, t_i)$  を通る多項式を  $N$  個重ねあわせた多項式  
 $\hookrightarrow t_i$  以外の  $t_1, t_2, \dots, t_N$  では 0 を通り、 $t = t_i$  では 1 を通る多項式を  $y_i$  倍した多項式  
 $\hookrightarrow t_i$  以外の  $t_1, t_2, \dots, t_N$  で 0 を通る多項式を  $t = t_i$  で取る値で割った多項式  
 最後の下線を満たす多項式を見つけるのは難しくない。次の  $p_1$  の形であれば十分この条件を満たす。

$$p_1(t) = (t - t_1)(t - t_2) \dots (t - t_{i-1})(t - t_{i+1}) \dots (t - t_N)$$

あとは、分解した問題を順々に解いていけばいい。

$$p_2(t) = \frac{(t - t_1)(t - t_2) \dots (t - t_{i-1})(t - t_{i+1}) \dots (t - t_N)}{(t_i - t_1)(t_i - t_2) \dots (t_i - t_{i-1})(t_i - t_{i+1}) \dots (t_i - t_N)}$$

$$p_3(t) = \frac{(t - t_1)(t - t_2) \dots (t - t_{i-1})(t - t_{i+1}) \dots (t - t_N)}{(t_i - t_1)(t_i - t_2) \dots (t_i - t_{i-1})(t_i - t_{i+1}) \dots (t_i - t_N)} y_i$$

$$p_4(t) = \sum_{i=1}^N \frac{(t - t_1)(t - t_2) \dots (t - t_{i-1})(t - t_{i+1}) \dots (t - t_N)}{(t_i - t_1)(t_i - t_2) \dots (t_i - t_{i-1})(t_i - t_{i+1}) \dots (t_i - t_N)} y_i$$

最後の  $p_4$  は Ragrange 補間の公式 Eq. (2.25) と一致している。

えば Eq. (2.1) は簡単に積分できる。

$$\begin{aligned}
\int_{t_n}^{t_{n+1}} \frac{dy}{dt} dt &= \int_{t_n}^{t_{n+1}} f(y, t) dt \\
\int_{y_n}^{y_{n+1}} dy &\simeq \int_{t_n}^{t_{n+1}} \left( \frac{f(y_n, t_n)}{h} (t - t_{n-1}) - \frac{f(y_n, t_n)}{h} (t - t_{n+1}) \right) dt \\
y_{n+1} - y_n &= \left[ \frac{1}{2} \frac{f(y_n, t_n)}{h} (t - t_{n-1})^2 - \frac{1}{2} \frac{f(y_{n-1}, t_{n-1})}{h} (t - t_n)^2 \right]_{t_n}^{t_{n+1}} \\
y_{n+1} - y_n &= \frac{1}{2} \frac{f(y_n, t_n)}{h} (3h^2) - \frac{1}{2} \frac{f(y_{n-1}, t_{n-1})}{h} (h^2) \\
y_{n+1} - y_n &= \frac{3}{2} f(y_n, t_n) h - \frac{1}{2} f(y_{n-1}, t_{n-1}) h \\
y_{n+1} &= y_n + \left( \frac{3}{2} f(y_n, t_n) - \frac{1}{2} f(y_{n-1}, t_{n-1}) \right) h
\end{aligned} \tag{2.26}$$

この漸化式は 2 段 2 次の Adams-Bashforth 法<sup>\*22</sup>と呼ばれる。

Adams-Bashforth 法

$$y_{n+1} = y_n + \left( \frac{3}{2} f(y_n, t_n) - \frac{1}{2} f(y_{n-1}, t_{n-1}) \right) h \tag{2.27}$$

同様に補間公式を利用して多項式近似を行えば、 $t = t_n, t_{n-1}, \dots, t_{n-i}, \dots, t_{n-N}$  の  $N$  個の値から  $t_{n+1}$  の値を計算することができる。これらも Adams-bashfourth 法と呼ばれる。

ま  $t = t_n, t_{n-1}, \dots, t_{n-i}, \dots, t_{n-N}$  の  $N$  個の値で多項式近似を行うことで、 $t_n$  の値を求めるようにすれば陰的公式を作ることも可能で、Adams-Moulton 法と呼ばれている。

Runge-Kutta 法と違い、多段法の場合は陰的公式も重要となる。なぜなら、計算が簡単な Adams-bashfourth 法で仮の解を求めたあと、Adams-Moulton 法で解を修正することができるからである。この方法は予測子修正子 Adams 法と呼ばれる。予測子修正子 Adams 法は陽公式の利点である計算の簡単さと、陰公式の利点である絶対安定領域の広さの両方を、ある程度成立させられる点で有効な方法である。

ただし、ステップ幅を変えると積分公式を作りなおす必要があり、この計算にかなりの時間がかかる。また、初期値が段数分必要となるため、最初の数ステップは Runge-Kutta 法などをつかって別に計算する必要がある。

### 2.4.3 2つの方法の比較

Runge-Kutta 法と線形多段法を安定性と計算量の観点から評価する。

陽的 Runge-Kutta 法は絶対安定領域が非常に狭いか、存在しない。陰的 Runge-Kutta 法は絶対安定領域が広く、A 安定なものが多いが、絶対的に計算が煩雑になる。線形多段法は A 安定なものは次数の低く使い物にならないが、必ず絶対安定領域が存在する。

<sup>\*22</sup> 2 つの過去の値を使っているので 2 段、Taylor 展開の 2 次までと一致するので 2 次の Adams-bashfourth 法と呼ばれる。こちらもあり聞かないが、Euler 法は 1 段 1 次の Adams-bashfourth 法ということもできる。

また、Runge-Kutta 法の階数と同じように段数と次数も一般に一致しない。経験的に 7 次の公式を作るためには最低でも 8 段が必要になることが知られている。

よって、安定性の観点から言うと、陰的 Runge-Kutta 法が最もよく、陽的 Runge-Kutta 法が最も悪い。線形多段法はその中間となる。

陽的 Runge-Kutta 法は逐次代入によって解くことができるので、すべての点を一回ずつ計算すれば計算が終了する。特別な工夫をしなければ、点の数  $n$  に比例した  $O(n)$  の計算量で十分となる。陰的 Runge-Kutta 法は非線形の連立方程式を解かなくてはならない。この計算量について一般的なことをいうことはできないが、線形の連立方程式の解法は  $O(n^3)$  程度なので、少なくともその程度の計算量が必要になると考えられる。線形多段法では予測子と修正子で二回の計算が必要となるが、どちらも  $O(n)$  なので、予測子修正子 Adams 法全体で見ても  $O(n)$  程度となる。

計算量の観点から見ると陽的 Runge-Kutta 法と線形多段法は同程度で、陰的 Runge-Kutta 法は相当に遅いということになる。

以上の結果を踏まえると、線形多段法は陽的 Runge-Kutta 法に対して常に同程度以上の性能<sup>\*23</sup>を発揮することがわかる。また、陰的 Runge-Kutta 法は安定性が最も高いが計算量も最も多い。

よって、線形多段法の絶対安定域内のステップ幅で計算が可能であれば、線形多段法が最も適していて、線形多段法の絶対安定域に収まるステップ幅では計算が終わらないようであれば、陰的 Runge-Kutta などの陰公式を使うことが最も適している。

#### 有限要素法についてのまとめ

- Runge-Kutta 法：線形近似を何回もやる
- 線形多段法:多項式近似で陽公式と陰公式を組み合わせる
- 基本は線形多段法で計算してみる
- 線形多段法で安定しなかったら陰公式を使う

<sup>\*23</sup> 線形多段法は過去のデータを保持しなくてはならない分、メモリ消費が多いが、現代の計算機の性能ではほとんど問題にならない。



## 第3章

# 先行研究

Kronig-Penny モデルは、Bloch の定理を適用可能な実例として提案された。箱型ポテンシャルが周期的に並んでいる状況を設定したモデルで、Kronig と Penny はこのモデルの固有状態を計算することによって、周期ポテンシャルにエネルギーバンド構造が現れることを説明した。Kronig-Penny モデル中の波束の時間発展については、Zhenjun による数値計算が行われている。この中で、重心が止まった波束の緩和速度について、分散の時間発展を使うことで数値化が行われた。

### 3.1 Bloch の定理と Kronig-Penny モデルの定常解

ハミルトニアンに並進対称性を与えると定常状態の波動関数が周期関数と三角関数の積で表せるというのが Bloch の定理 Eq. (1.5) の内容である。これを今から証明する。

まず、証明のために演算子の交換関係についての有用な定理を証明する。

交換関係に関する定理

ある Hermit 演算子  $A$ 、 $B$  が交換関係  $[A, B] = 0$  を満たすとき  $A$ 、 $B$  は同時固有状態を持つ。

$A$  の固有値  $\alpha$  に対応する固有関数を  $\phi_\alpha$  とする。交換関係の仮定から Eq. (??) が成立する。

$$\begin{aligned} AB\phi_\alpha &= BA\phi_\alpha \\ AB\phi_\alpha &= B\alpha\phi_\alpha \\ AB\phi_\alpha &= \alpha B\phi_\alpha \end{aligned} \tag{3.1}$$

Eq. (??) から、 $B\phi_\alpha$  は  $A$  の  $\alpha$  に対する固有関数になっていることがわかる。 $A$  の固有状態に縮退がないとすると、 $\alpha$  と  $\phi_\alpha$  は定数倍を除いて一対一対応するので、 $B\phi_\alpha$  は  $\phi_\alpha$  の定数倍以外ありえない。よって比例定数を  $\beta$  とすれば、Eq. (3.2) が成立するが、これは  $\phi_\alpha$  が  $B$  の固有関数であることを意味している。

$$B\phi_\alpha = \beta\phi_\alpha \tag{3.2}$$

また、 $A$  の固有状態に縮退がある場合、他の固有状態を  $\phi'_\alpha$  とすると\*<sup>1</sup>  $B\phi_\alpha$  は  $\phi_\alpha$  と  $\phi'_\alpha$  との線型結合で表さ

---

\*<sup>1</sup>  $\phi'_\alpha$  以外にも  $\phi''_\alpha$  や  $\phi'''_\alpha$  があっても構わないが、簡単のために一つだけの場合を考える。

れる。適当な係数を  $\beta_1$ 、 $\beta_2$  とすれば Eq. (??) が成立する。

$$B\phi_\alpha = \beta_1\phi_\alpha + \beta_2\phi'_\alpha \quad (3.3)$$

$$B\phi'_\alpha = \beta'_1\phi_\alpha + \beta'_2\phi'_\alpha \quad (3.4)$$

ここで、 $\phi_\alpha + \beta_2/\beta_1\phi'_\alpha$  の因子は  $B$  の  $\beta_1$  に対する固有関数であることがわかる。また、 $\phi_\alpha$  と  $\phi'_\alpha$  の線型結合で表されることから、 $A$  の  $\alpha$  に対する固有関数であることもわかる。 $A$  に縮退がある場合もない場合も  $A$  と  $B$  は同時固有関数を持つことがわかったため、定理は証明された。

## 3.2 Zenjun の超拡散

## 第4章

## 結果

## 第5章

## 考察

## 第6章

# schrpy について