

偏微分方程式のアンニーリング解法

工藤 和恵

お茶の水女子大学 基幹研究院自然科学系
東北大学 大学院情報科学研究科

参考文献: [arXiv:2406.17364](https://arxiv.org/abs/2406.17364)

自己紹介

現在の所属

- お茶の水女子大学
基幹研究院 (理学部情報科学科) 教授
- 東北大学 大学院情報科学研究科 教授
(クロスアポイント)

【出身は理学部物理学科】

研究分野

- 量子多体系 (特に1次元量子スピン系)
 - 量子アニーリングの基礎研究
- 量子コンピューティング
 - イジングマシンの活用



量子コンピューティング技術

ゲート型

量子コンピュータ

- (理想的には) 汎用的な計算を高速に実行
- 量子力学の原理を利用した計算
- 現在のところ小規模なマシンしか実現していない

イジングマシン

(アニーリングマシン)

- 組合せ最適化問題に特化して高速計算
- 量子性を利用しないマシンもある
- ゲート型量子コンピュータと比較して規模が大きい

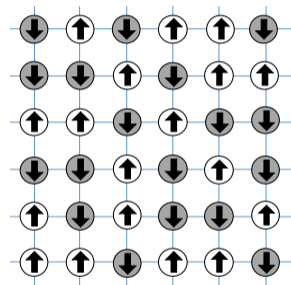
イジングマシンとイジングモデル

- イジングマシン: イジングモデルの基底状態を効率よく求めるマシン
- イジングモデル: もともと磁性体のモデル

- **スピン**を持つ粒子があり, 互いにいくつかの粒子と相互作用している.

右図ではそれぞれが上下左右の粒子と相互作用

- スピンは**上向き**または**下向き**のいずれかの状態をとる.



ポイント

スピンの向き of 最適な組合せの状態 \Leftrightarrow イジングモデルの基底状態

Outline

- 1 研究目的
- 2 方法
- 3 結果
- 4 まとめ

量子計算や量子アニーリングを科学計算に利用したい

ターゲット：偏微分方程式

- 偏微分方程式は基礎科学でも工学的応用分野でも頻出する。
- 偏微分方程式の解析解は、一般的に得るのが難しい。
→ ほとんどの場合は数値計算が必須。

量子コンピューティング解法

- 誤り耐性量子コンピュータ → まだまだ実用的でない
- 変分量子アルゴリズム → 比較的有望
- アニーリング/イジングマシン → もっと有望

目的：イジングマシンで偏微分方程式を解く

イジングマシンでできる計算は目的関数の最小化

偏微分方程式を解くためのステップ:

- 偏微分方程式を離散化 → 連立1次方程式
- 連立1次方程式 → 一般化固有値問題
- 一般化固有値問題 → 一般化レイリー商の最小化
- 最小化問題 → アニーリング

Outline

- 1 研究目的
- 2 方法
- 3 結果
- 4 まとめ

連立1次方程式

偏微分方程式の離散化によって得られた連立1次方程式:

$$Ku = f$$

K は正定値行列, u と f は実ベクトルとする.

目的: K と f を与えて, 解 u を計算する.

$Ku = f$ を $\tilde{K}u = \tilde{f}$ と書き換えておく. ($\tilde{K} = K/\|f\|$, $\tilde{f} = f/\|f\|$)

一般化固有値問題

一般化固有値問題: $A\mathbf{v} = \lambda B\mathbf{v}$

- A : 対称行列, B : 正定値対称行列
- λ : 一般化固有値, \mathbf{v} : 対応する固有ベクトル

一般化レイリー商: $R(A, B; \mathbf{w}) = \frac{\mathbf{w}^\top A \mathbf{w}}{\mathbf{w}^\top B \mathbf{w}}$

- \mathbf{w} : 単位ベクトル

- 最小の一般化固有値 λ_{\min} は, 一般化レイリー商の最小値と等しい.
- λ_{\min} に関する固有ベクトル \mathbf{v}_{\min} が一般化レイリー商を最小化する.

連立1次方程式と一般化固有値問題との対応

$$\tilde{K}\mathbf{u} = \tilde{\mathbf{f}}$$

↓

$$\alpha\tilde{\mathbf{f}} = \alpha\tilde{K}\mathbf{u}$$

$$\left\{ \begin{array}{l} \alpha = -\tilde{\mathbf{f}}^\top \mathbf{v} \\ \alpha \mathbf{u} = \lambda \mathbf{v} \end{array} \right.$$

$$-\tilde{\mathbf{f}}\tilde{\mathbf{f}}^\top \mathbf{v} = \lambda\tilde{K}\mathbf{v}$$

$$\left\{ \begin{array}{l} -\tilde{\mathbf{f}}\tilde{\mathbf{f}}^\top = A \\ \tilde{K} = B \end{array} \right.$$

$$A\mathbf{v} = \lambda B\mathbf{v}$$

$-\tilde{\mathbf{f}}\tilde{\mathbf{f}}^\top$ はランク1の行列

↓

非ゼロ固有値が1個だけ $\rightarrow \lambda_{\min}$

$$-\tilde{\mathbf{f}}\tilde{\mathbf{f}}^\top \mathbf{v}_{\min} = \lambda_{\min}\tilde{K}\mathbf{v}_{\min}$$

↓

▶ details

$$\mathbf{u} = -\frac{\lambda_{\min}\mathbf{v}_{\min}}{\tilde{\mathbf{f}}^\top \mathbf{v}_{\min}}$$

アルゴリズム

Krakoff らの提案したアルゴリズム [PLOS ONE 17, e0267954 (2022)] に基づく。

$$Av = \lambda Bv$$

- $A, B \in \mathbb{R}^{n \times n}$: 対称行列 (かつ B は正定値行列)
- 固有ベクトル $v \in \mathbb{R}^n$ を $v = (I_n \otimes p)q$ で近似
- $p = (-1, \frac{1}{2}, \frac{1}{2^2}, \dots, \frac{1}{2^{b-1}})$: 行ベクトル
- $q = \{0, 1\}^{nb}$: 二値ベクトル

2段階のアルゴリズム

- 初期推定: アニーリングで $\frac{1}{2^{b-1}}$ の精度で解の近似値を得る
- 反復降下: 解の精度を細かくしながらアニーリングを繰り返す

固有ベクトル $v = (I_n \otimes p)q$ の形:

$$v = \left[\begin{array}{c|c|c|c} -1, \frac{1}{2}, \dots, \frac{1}{2^{b-1}} & 0 & \dots & 0 \\ \hline 0 & -1, \frac{1}{2}, \dots, \frac{1}{2^{b-1}} & & \vdots \\ \hline \vdots & & \ddots & 0 \\ \hline 0 & \dots & 0 & -1, \frac{1}{2}, \dots, \frac{1}{2^{b-1}} \end{array} \right] q.$$

v の各要素は $[-1, 1 - \frac{1}{2^{b-1}}]$ の間の離散値をとる.

2段階の手法を用いることで変数の数を増やさずに精度を向上させる

初期推定

初期値 $\lambda = \mathbf{x}^\top A \mathbf{x}$ (\mathbf{x} : ランダムな単位ベクトル) を設定後, λ が収束するまで次の2つの手順を繰り返す.

(1) 暫定固有ベクトル $\mathbf{v}^* = \underset{\mathbf{v}}{\operatorname{argmin}} [\mathbf{v}^\top (A - \lambda B) \mathbf{v}]$ の計算.

- アニールングの実行: $\mathbf{q}^* = \underset{\mathbf{q}}{\operatorname{argmin}} [\mathbf{q}^\top (A - \lambda B) \otimes (\mathbf{p}^\top \mathbf{p}) \mathbf{q}]$
- 実ベクトルへの変換: $\mathbf{v} = (I_n \otimes \mathbf{p}) \mathbf{q}^*$
- ベクトルの規格化: $\mathbf{v}^* = \mathbf{v} / \|\mathbf{v}\|$

(2) 推定固有値の更新: $\lambda = \frac{\mathbf{v}^{*\top} A \mathbf{v}^*}{\mathbf{v}^{*\top} B \mathbf{v}^*}.$

反復降下

初期値 $r = r_{\text{init}}$ を設定後, $r \leq \epsilon_0$ を満たすか, λ が収束するまで以下の手順を繰り返す.

(1) $\mathbf{v}^\top (A - \lambda B) \mathbf{v}$ を最小化する方向ベクトル \mathbf{d} の計算.

- アニールングの実行: $\mathbf{q}^* = \underset{\mathbf{q}}{\operatorname{argmin}} [2r\mathbf{v}^\top C (I_n \otimes \mathbf{p}) \mathbf{q} + r^2 \mathbf{q}^\top (C \otimes \mathbf{p}^\top \mathbf{p}) \mathbf{q}]$

ただし, $C = A - \lambda B$.

- 方向ベクトルの計算: $\mathbf{d} = \mathbf{d}^* - (\mathbf{v}^\top \mathbf{d}^*) \mathbf{v}$. ただし, $\mathbf{d}^* = (I_n \otimes \mathbf{p}) r \mathbf{q}^*$.

※ \mathbf{v} は単位ベクトルなので \mathbf{d} は単位球面上の差分 $\rightarrow \mathbf{v} \perp \mathbf{d}$

(2) 暫定固有ベクトルの計算: $\mathbf{v}^* = \mathbf{v} + t\mathbf{d}$.

$$t = \begin{cases} \operatorname{sgn}(b) \max\left(\frac{|b|}{a}, 1\right) & a > 0, \\ 1 & a \leq 0, \end{cases}$$

$$\begin{aligned} a &= \mathbf{d}^\top (A - \lambda B) \mathbf{d} \\ b &= -\mathbf{v}^\top (A - \lambda B) \mathbf{d} \end{aligned}$$

- 暫定固有ベクトルの規格化: $\mathbf{v}^* \leftarrow \mathbf{v}^* / \|\mathbf{v}^*\|$

(3) 推定固有値の更新: $\lambda^* = \frac{\mathbf{v}^{*\top} A \mathbf{v}^*}{\mathbf{v}^{*\top} B \mathbf{v}^*}$.

(4a) $\lambda^* < \lambda$ の場合, 更新を受理: $\lambda = \lambda^*$, $\mathbf{v} = \mathbf{v}^*$. \rightarrow 手順 1 に戻る.

(4b) $\lambda^* \geq \lambda$ の場合, メッシュを細かくする: $r \leftarrow \eta r$. (η : 実数)
 \rightarrow 手順 1 に戻る

本研究では, 精度更新比率を $\eta = 2^{-b+1}$ とする.
(先行研究では η は一定.)

Outline

- 1 研究目的
- 2 方法
- 3 結果**
- 4 まとめ

ポアソン方程式

$$-\nabla^2 u(\mathbf{x}) = f(\mathbf{x})$$

(1D) $x \in [0, 1]$. 境界で $u(x) = 0$.

(2D) $x, y \in [0, 1]$. 境界で $u(x, y) = 0$.

● 対称 1次元: $u(x) = x^2(x - 1)^2$

$$f_{\text{sym}}(x) = -12x^2 + 12x - 2$$

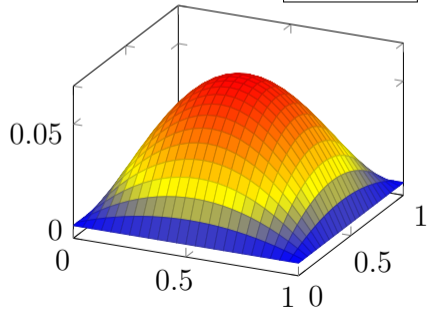
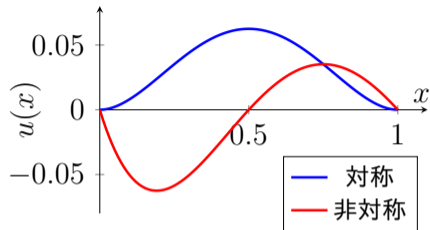
● 非対称 1次元:

$$u(x) = x(x - 1)(2x - 1)(2x - 3)/4$$

$$f_{\text{asym}}(x) = -12x^2 + 18x - 5.5$$

● 対称 2次元: $u(x, y) = xy(x - 1)(y - 1)$

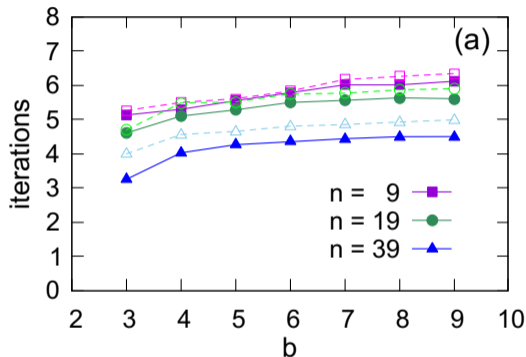
$$f_{2D}(x, y) = -2x(x - 1) - 2y(y - 1)$$



数値計算条件

- 各次元で領域 $[0, 1]$ を $n + 1$ 個に分割
→ 系のサイズ: n (1次元), n^2 (2次元)
- $r_{\text{ini}} = \eta = 2^{-b+1}$ (特に言及しないかぎり)
- $\epsilon_0 = \epsilon_1 = 10^{-8}$
- 同じ計算条件のもとで 10^3 回の試行
- **正解率**: 得られた解の二乗平均平方根誤差 (RMSE) が ϵ_1 以下である割合
- **正解** (RMSE の算出に使う): 対応する連立 1 次方程式を古典の線形ソルバーで解いた解

精度依存性 (対称1次元)



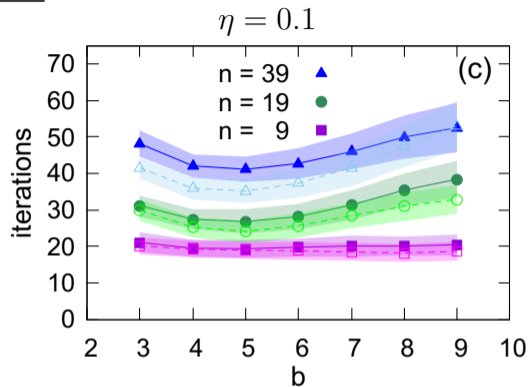
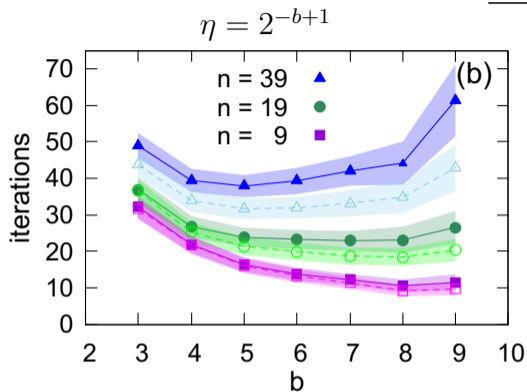
初期推定

- 実線&色塗り記号: $N_{\text{step}} = 10^3$
- 破線&白抜き記号: $N_{\text{step}} = 10^4$

N_{step} : 温度ステップ数 (アニーリング時間に対応)

アニーリング時間が短いほど、大きい系ほど、反復回数が小さい。

反復降下



- 実線 & 色塗り: $N_{\text{step}} = 10^3$. 破線 & 白抜き: $N_{\text{step}} = 10^4$
- 記号の大きさは正解率に比例

アニーリング時間を10倍にしても、反復回数は20%も減らない。

精度更新回数

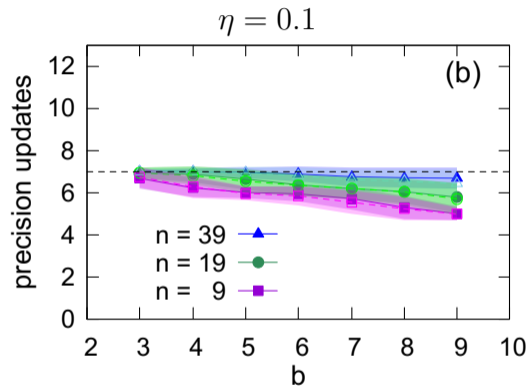
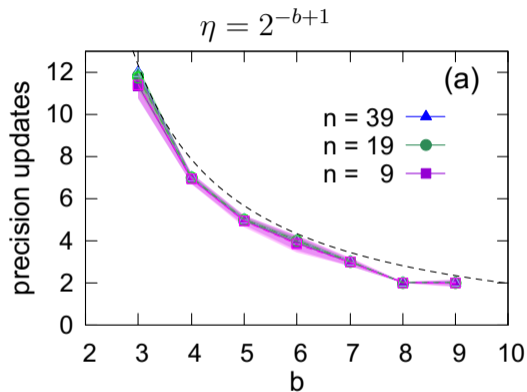
メッシュのスケール r は, n_{upd} 回更新後に $r = r_{\text{ini}}\eta^{n_{\text{upd}}}$ となる. このとき

$$n_{\text{upd}} = \frac{\log_2 r - \log_2 r_{\text{ini}}}{\log_2 \eta}.$$

とくに $\eta = r_{\text{ini}} = 2^{-b+1}$ の場合は

$$n_{\text{upd}} = \frac{\log_2 r^{-1}}{b-1} - 1.$$

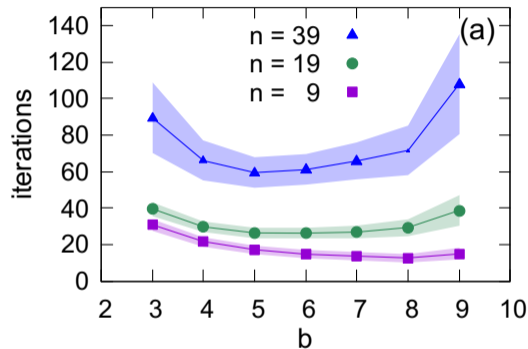
精度更新回数 (対称1次元)



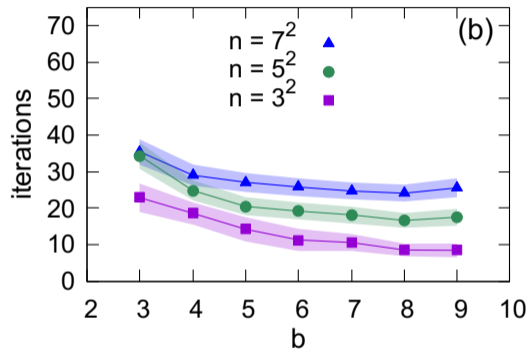
- 黒破線は理論的な見積もり
- 破線より下に点がある → 固有値 λ の収束による早期終了

問題の違いによる反復回数の違い

非対称 1次元



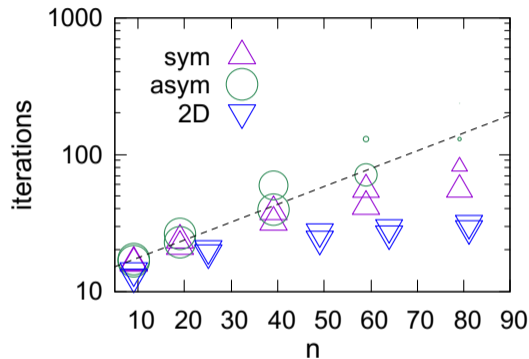
対称 2次元



記号の大きさは正解率に比例。 $N_{\text{step}} = 10^3$.

正解が対称な凸関数の場合は、最適化問題として比較的容易。

系のサイズ依存性



- $b = 5, N_{\text{step}} = 10^3, 10^4$
- 記号の大きさは正解率に比例
- 破線: $ce^{\alpha n}$
($c = 13, \alpha = 0.03$)

対称 1次元・2次元のポアソン方程式に関しては、反復回数が系のサイズ n に対して指数関数よりもゆるやかに増加する。

Outline

- 1 研究目的
- 2 方法
- 3 結果
- 4 **まとめ**

注意点

- 今回は離散化誤差を考えていない。
 - 偏微分方程式の解としての精度向上には、 n の増加が必須だが、計算コストも増大する。
 - 系のサイズ n は精度パラメタ b よりも影響が大きい。
- 指数的/劣指数的に反復回数が増える。
 - 固有値問題の典型的な数値解法の計算量は $O(n^3)$ 。
 - 提案手法での計算時間は、 $e^{\alpha n}$ で増加する非対称の場合でも α は比較的小さい。→ 高性能なイジングマシンを使えば、問題によっては従来手法よりも優位になるかもしれない。
- 実際のアニーリング時間は、反復回数だけでなく変数の数 nb にも依存する。

まとめ

- 反復回数は系のサイズとともに指数的/劣指数的に増大するが、その指数は比較的小さい。
 - ただし、大きな系の場合に短いアニーリング時間では、正解率が非常に低い。
- 問題の性質が計算性能に影響する。
 - 対称な問題では非対称な問題よりも少ない反復回数で解が得られる。
- イジングマシンを利用することで、問題によっては提案手法が従来手法よりも優位になる可能性がある。

Backup slides

一般化固有値問題

$$-\tilde{\mathbf{f}}\tilde{\mathbf{f}}^\top \mathbf{v}_{\min} = \lambda_{\min} \tilde{K} \mathbf{v}_{\min} \quad \rightarrow \quad \tilde{K} \begin{pmatrix} -\frac{\lambda_{\min} \mathbf{v}_{\min}}{\tilde{\mathbf{f}}^\top \mathbf{v}_{\min}} \end{pmatrix} = \tilde{\mathbf{f}}$$
$$\rightarrow \text{Combined with } \tilde{K} \mathbf{u} = \tilde{\mathbf{f}}, \quad \tilde{K} \left(\mathbf{u} + \frac{\lambda_{\min} \mathbf{v}_{\min}}{\tilde{\mathbf{f}}^\top \mathbf{v}_{\min}} \right) = \mathbf{0}.$$
$$\rightarrow \tilde{K} \text{ は正則行列なので, } \mathbf{u} = -\frac{\lambda_{\min} \mathbf{v}_{\min}}{\tilde{\mathbf{f}}^\top \mathbf{v}_{\min}}.$$

反復降下

$$\text{目的関数: } g(\mathbf{x}) = \mathbf{x}^\top (A - \lambda B)\mathbf{x}.$$

\mathbf{x}_0 の周りでテイラー展開すると

$$g(\mathbf{x}) = g(\mathbf{x}_0) + \mathbf{d}^\top \nabla g(\mathbf{x}_0) + \frac{1}{2} \mathbf{d}^\top (\nabla^2 g(\mathbf{x}_0)) \mathbf{d} + O(\|\mathbf{d}\|^3),$$

(ただし, $\mathbf{d} = \mathbf{x} - \mathbf{x}_0$, $\nabla g = 2(A - \lambda B)\mathbf{x}$, $\nabla^2 g = 2(A - \lambda B)$.)

- $g(\mathbf{x})$ を最小化する方向ベクトル \mathbf{d} を計算
- 解の更新: $\mathbf{x}_0 \leftarrow \mathbf{x}_0 + t\mathbf{d}$. (t : 実数)

- $g(\mathbf{x})$ を最小化する差分ベクトル:

$$\mathbf{d}^* = \operatorname{argmin}_d [2\mathbf{v}^\top (A - \lambda B)\mathbf{d} + \mathbf{d}^\top (A - \lambda B)\mathbf{d}].$$

\mathbf{v} : 暫定固有ベクトル

- \mathbf{d}^* を \mathbf{v} に対して直交化: $\mathbf{d} = \mathbf{d}^* - (\mathbf{v}^\top \mathbf{d}^*)\mathbf{v}$.
 - \mathbf{v} は単位ベクトルなので単位球面上を動く. \rightarrow 差分ベクトル \mathbf{d} は \mathbf{v} に直交.
- 固有ベクトルの更新: $\mathbf{v} \leftarrow \mathbf{v} + t\mathbf{d}$
 - t は $g(\mathbf{v} + t\mathbf{d}) = (\mathbf{v} + t\mathbf{d})^\top (A - \lambda B)(\mathbf{v} + t\mathbf{d})$ を最小化するように決める.
 - $g(\mathbf{v} + t\mathbf{d})$ は t の2次関数だから, $\mathbf{d}^\top (A - \lambda B)\mathbf{d} > 0$ のとき最小値が1つある.

- $t \geq 1$ となるように決める

$$t = \begin{cases} \operatorname{sgn}(b) \max\left(\frac{|b|}{a}, 1\right) & a > 0, \\ 1 & a \leq 0, \end{cases}$$

ただし, $a = \mathbf{d}^\top (A - \lambda B) \mathbf{d}$, $b = -\mathbf{v}^\top (A - \lambda B) \mathbf{d}$.

- $t \geq 1$ のとき $\mathbf{v} + t\mathbf{d}$ は最適解を通り越すかもしれない.
 - そのときは, メッシュサイズが解を改良するのに十分小さくないと考えられる. $\rightarrow \mathbf{v}$ を更新する代わりにメッシュを細かくする.
- メッシュサイズが $\epsilon_0/2^{b-1}$ になったらアルゴリズムを停止.
 - ϵ_0 : 許容精度

原理的には, 上記の手順で一般化固有値問題を任意精度で解くことができる.