Vol.2018-HPC-163 No.4
2018/2/28

Lanczos法を利用したTSVD法の積分方程式への応用

辺　遠[1,a)]　野寺　隆[2,b)]

概要：本稿は，非適切問題の中で代表的な第一種 Fredholm 積分方程式の数値解法について考える．最初に，このような問題の解法に用いられる TSVD の基本的な考え方について述べる．次に，TSVD 法を実行できる古典的な Lanczos 法について，重み付きの積分公式を用いた場合に精度良く近似解を計算できる修正法を提案する．最後に，数値実験を通して，提案手法の有効性について述べる．

Lanczos type iteration to TSVD method for solving Fredholm integral equation of the first kind

Yuan Bian[1,a)]　Takashi Nodera[2,b)]

Abstract: Recently, a method called TSVD (Truncated Singular Value Decomposition) has been attracting some attention for solving Fredholm integral equation of the first kind. Solving a discretized this problem is a well-known ill-conditioned problem. This paper explains why the TSVD method performs well when solving Fredholm integral equations of the first kind, and proposes a new kind of TSVD method, which is combined with the Lanczos method. The results of numerical experiments are shown to confirm the effectiveness of our proposed method.

1. はじめに

第一種 Fredholm 積分方程式:

$$\int_a^b K(x,y)f(y)dy = g(x) \tag{1}$$

を考える．ただし，積分核と呼ばれる 2 変数関数 $K(x,y)$ と右側の関数 $g(x)$ が既知で，$f(y)$ は求めたい未知関数である．ここで，$f, g \in L^2[a,b]$，$K \in L^2([a,b]\times[a,b])$ を考える．この方程式を離散化しすると，次のような線形方程が得られる．

$$A\boldsymbol{x} = \boldsymbol{b}, \quad A \in \mathbb{R}^{n\times n},\ \boldsymbol{b} \in \mathbb{R}^n \tag{2}$$

式 (2) は，数値的に簡単に解けそうに見えるが，実際，第一種 Fredholm 積分方程式を離散化して得られるので代表的な悪条件問題となり，ほとんどの場合で行列 A の条件数が非常に大きい．つまり，計算中で微小な誤差でも結果に莫大な影響を与えることになる．ゆえに，ただ $\boldsymbol{x} = A^{-1}\boldsymbol{b}$ で計算すれば大きな誤差が生じる可能性が高く，精度のよい解を求めるのはそれほど容易ではない．それ解決策の 1 つは，TSVD 法を利用することである．

TSVD(Truncated Singular Value Decomposition) 法は，悪条件の線形方程式 (2) を解くためによく使われる手法の 1 つである．特に非適切な問題として知られている第一種 Fredholm 積分方程式の数値解として有効である．

行列 A の条件数が非常に大きい場合，通常，行列 A が多数の 0 に近い特異値を持つことが多い．そのため，それらを直接 0 にすることにより，条件数を著しく減少させることができ，精度のよい解を得ることが期待できるのが TSVD 法の基本的な考え方である．具体的に言えば，行列 A が次のような特異値分解を持つとする．

$$A = U\Sigma V^*,\ \Sigma = \mathrm{diag}\{\sigma_1, \sigma_2, \cdots, \sigma_n\} \tag{3}$$

もし $\sigma_{k+1}, \cdots, \sigma_n$ が 0 に近い特異値であれば，それらを直接 0 にする．この操作で，元の行列 A が次のようになる．

$$A_k = U\Sigma_k V^*,\ \Sigma_k = \mathrm{diag}\{\sigma_1, \sigma_2, \cdots, \sigma_k\, 0, \cdots, 0\} \tag{4}$$

[1] 慶應義塾大学大学院理工学研究科
〒 223-8522 横浜市港北区日吉 3-14-1
[2] 慶應義塾大学理工学部
〒 223-8522　横浜市港北区日吉 3-14-1
[a)] vergilbian@keio.jp
[b)] nodera@math.keio.ac.jp

ここで，A_k が正則行列でないため，正確な解が必ずしも存在するとは限らない．それ故に，新しい問題を最小二乗問題として解く必要がある．

$$\boldsymbol{x}_k = \underset{\boldsymbol{x}}{\operatorname{argmin}} \|A_k\boldsymbol{x} - \boldsymbol{b}\| \tag{5}$$

この最小二乗問題を解いて，$\mathbf{x}_k$ は次の式で与えられる．

$$\boldsymbol{x}_k = V\Sigma_k^{-1}U_*\boldsymbol{b} \tag{6}$$

計算の途中で小さい特異値が使われなかったため，$\boldsymbol{x}_k$ を比較的精度よく求めることができる．しかし，TSVD 法の考え方にそのまま従って計算すると，行列 A の完全な特異値分解が必要であるため，十分な計算時間が必要となる．それだけでなく，上記の議論だけで，$\boldsymbol{x}_k$ が $\boldsymbol{x}$ に近い，即ち，特異値の切り捨ての操作が解にもたらす影響が小さいことを説明するのは至難の技である．

本稿ではそのような問題点を改善して，まず連続の視点から，第一種 Fredholm 積分方程式のための TSVD 法を導出し，積分方程式に対する有効性について述べる．次に，離散化した問題に対して，Lanczos 法を利用した TSVD 法を提案する．最後に，いくつかの数値実験を通して，提案手法の有効性を検証する．

2. 第一種 Fredholm 積分方程式の TSVD 法

式 (1) で与えられた積分方程式を考える．その積分を関数 f に掛ける作用素とみなせば，作用素の形式で書き直すことができる．

$$\mathcal{K}f = g \tag{7}$$

ここで，作用素 $\mathcal{K}$ は次のようなものである．

$$\mathcal{K}f(y) = \int_a^b K(x,y)f(y)dy \tag{8}$$

ただし，$K \in L^2([a,b]\times[a,b])$ より，$\mathcal{K}$ は $L^2[a,b]$ から $L^2[a,b]$ への作用素で，有界（ゆえに連続）線型作用素である．そのため，随伴作用素 $\mathcal{K}^*$ が存在する．また，$\mathcal{K}$ がコンパクトであることも証明できる [6]．よって，$\mathcal{K}^*\mathcal{K}$ が自己随伴な半正定値で，かつコンパクトな作用素であり，次式を満たす固有値 σ_i と固有関数 v_i が存在する．

$$\mathcal{K}^*\mathcal{K}v_i = \sigma_i v_i, \quad i = 1,2,\cdots \tag{9}$$

さらに，$\{v_i\}_{i=1}^{\infty}$ は正規直交系をなす．また，非零の σ_i の個数が有限個，または可算無限個である [5]．そこで，最初に，非零の σ_i の個数が可算無限個であるのを仮定する．有限個の場合は本節の最後で考えることにする．

2.1 非零の σ_i が加算無限個である場合

最初に，$\mathcal{K}^*\mathcal{K}$ の 1 つの固有ペア (σ_i^2, v_i) に対して，次の関数 u_i を定義する．

$$u_i = \sigma_i^{-1}\mathcal{K}v_i \tag{10}$$

u_i の定義より，内積 $\langle u_i,u_i\rangle$ と $\langle u_i,u_j\rangle$（ただし，$u_i \neq u_j$）は，次のように計算できる．

$$\begin{aligned}\langle u_i,u_i\rangle &= \langle\sigma_i^{-1}\mathcal{K}v_i,\sigma_i^{-1}\mathcal{K}v_i\rangle = \sigma_i^{-2}\langle v_i,\mathcal{K}^*\mathcal{K}v_i\rangle \\ &= \sigma_i^{-2}\langle v_i,\sigma_i^2 v_i\rangle = 1 \end{aligned}\tag{11}$$

$$\begin{aligned}\langle u_i,u_j\rangle &= \langle\sigma_i^{-1}\mathcal{K}v_i,\sigma_i^{-1}\mathcal{K}v_j\rangle = \sigma_i^{-2}\langle v_i,\mathcal{K}^*\mathcal{K}v_j\rangle \\ &= \sigma_i^{-2}\langle v_i,\sigma_i^2 v_j\rangle = 0 \end{aligned}\tag{12}$$

式 (11) と (12) より，$\{u_i\}_{i=1}^{\infty}$ も正規直交系をなす．

次に，$\mathcal{K}^*u_i$ を考える．

$$\mathcal{K}^*u_i = \mathcal{K}^*\langle\sigma_i^{-1}\mathcal{K}v_i\rangle = \sigma_i^{-1}\langle\mathcal{K}^*\mathcal{K}v_i\rangle = \sigma v_i \tag{13}$$

式 (10) と式 (13) を合わせて，次のことがわかる．

$$\mathcal{K}v_i = \sigma_i u_i,\ \mathcal{K}^*u_i = \sigma_i v_i \tag{14}$$

式 (14) より，σ_i，u_i，v_i は $\mathcal{K}$ に対して，行列の特異値と特異ベクトルと似たような役割である．それらを用いて，関数 $K(x,y)$ の行列の特異値分解と同様なものを考える．

v_i の定義より，$K(x,y)$ は v_i と内積を取れることが保証される．故に，$K(x,y)$ を y の関数とみなせば，$K(x,y) \in \operatorname{span}\{v_i\}_{i=1}^{\infty}$ である．そのため，$K(x,y)$ に対して次のような展開ができる．

$$\begin{aligned}K(x,y) &= \sum_{i=1}^{\infty}\langle K(x,y),v_i(y)\rangle v_i(y) \\ &= \sum_{i=1}^{\infty}(\mathcal{K}v_i)v_i(y) \\ &= \sum_{i=1}^{\infty}\sigma_i u_i(x)v_i(y)\end{aligned}\tag{15}$$

式 (15) より，行列の特異値分解と類似する関数 $K(x,y)$ の特異値分解が得られた．次に，この分解を用いて，元の方程式の解を考える．

式 (15) を最初の方程式 (1) に代入すると，左側が次のようになる．

$$\begin{aligned}\int_a^b K(x,y)f(y)dy &= \sum_{i=1}^{\infty}\int_a^b \sigma_i u_i(x)v_i(y)f(y)dy \\ &= \sum_{i=1}^{\infty}\sigma_i u_i(x)\int_a^b v_i(y)f(y)dy \\ &= \sum_{i=1}^{\infty}\sigma_i\langle f,v_i\rangle u_i(x)\end{aligned}\tag{16}$$

式 (16) より，式 (1) の左側が正規直交系 $\{u_i\}_{i=1}^{\infty}$ の張る空間に入っている．そのため，右側の関数 g は任意に選ぶことができない．解が存在することを保証するため，g も $\{u_i\}_{i=1}^{\infty}$ の張る空間の中にないといけない．この時，g は次の展開を持つ．

$$g = \sum_{i=1}^{\infty} \langle g, u_i \rangle u_i \tag{17}$$

式 (16) と式 (17) を合わせると，方程式 (1) は次式のようになる．

$$\sum_{i=1}^{\infty} \sigma_i \langle f, v_i \rangle u_i(x) = \sum_{i=1}^{\infty} \langle g, u_i \rangle u_i(x) \tag{18}$$

式 (18) の両側の u_i の係数を比較すると，次のことがわかる．

$$\begin{aligned} &\sigma_i \langle f, v_i \rangle = \langle g, u_i \rangle \\ \Rightarrow &\langle f, v_i \rangle = \sigma_i^{-1} \langle g, u_i \rangle \end{aligned} \tag{19}$$

最後に，f の v_i によって展開すると，σ_i，u_i，v_i を用いて解 f を表示することができる．

$$\begin{aligned} f &= \sum_{i=1}^{\infty} \langle f, v_i \rangle v_i \\ \Rightarrow f &= \sum_{i=1}^{\infty} \sigma_i^{-1} \langle g, u_i \rangle v_i \end{aligned} \tag{20}$$

前述の $\lim_{i \to \infty} \sigma_i = 0$ より，式 (20) の右側の級数の項 $\sigma_i^{-1} \langle g, u_i \rangle v_i$ が発散するように見えるので，全体の級数も発散するように見える．しかし，$K(x, y) \in L^2([a, b] \times [a, b])$ であれば，作用素 $\mathcal{K} : L^2[a, b] \to L^2[a, b]$ を定義することができ，任意の $g \in \mathcal{R}(\mathcal{K})(\subset L^2[a, b])$ に対して，必ず $\mathcal{K}f = g, f \in L^2[a, b]$ となる原像 f が存在する．上記より f は，式 (20) のような形を持つ．f が $L^2[a, b]$ の元であることより，L^2 ノルムが有限なので，式 (20) の右側の級数が必ず収束する．

式 (20) の右側が収束する時，すなわち，$k \to \infty$ の時，級数の前 k 項を除いた第 $k+1$ 項からの総和が 0 に収束する．つまり，次式が成り立つ．

$$\lim_{k \to \infty} \sum_{k+1}^{\infty} \sigma_i^{-1} \langle g, u_i \rangle v_i = 0 \tag{21}$$

そのため，式 (21) の右側の前 k 項の総和だけで，目標の解 f の近似値 f_k を求めることができる．

$$f \approx f_k = \sum_{i=1}^{k} \sigma_i^{-1} \langle g, u_i \rangle v_i \tag{22}$$

これで，代数的 TSVD 法と同様に，小さい特異値を切り捨てることによって，連続の場合の解 f の近似値を特異値と特異ベクトルで表すことができた．

2.2 非零の σ_i が有限個である場合

非零の σ_i が有限個である場合，その個数を $N \in \mathbb{N}$ とすれば，$i > N$ であるすべての $\sigma_i = 0$ になる．この時，$i = 1, 2, \cdots, N$ に対して，u_i は前述と同様に定義でき，$\mathcal{K}\mathcal{K}^*$ の σ_i に対応する固有関数であることには変わらない．N より大きい i に対して，u_i を $\mathcal{K}\mathcal{K}^*$ の 0 に対応する固有関数とすれば，次式が成り立つことになる．

$$\mathcal{K}^*\mathcal{K}v_i = \mathcal{K}\mathcal{K}^*u_i = 0, \ i > N \tag{23}$$

よって，$\langle \mathcal{K}v_i, \mathcal{K}v_i \rangle = \langle \mathcal{K}^*\mathcal{K}v_i, v_i \rangle = 0$ であることより，$\mathcal{K}v_i = 0$ であることがわかる．同様に，$\mathcal{K}^*u_i = 0$ も成り立つ．つまり，$i > N$ の時，式 (14) が成り立ち，それに続く同じ理論で，式 (19) の前半 $\sigma_i \langle f, v_i \rangle = \langle g, u_i \rangle$ が得られる．この時，式 (20) のような級数で f を表すことができないが，$i > N$ の i に対して，$\langle f, v_i \rangle$ は任意の値を取ることができる．つまり，この場合は，解が一意ではない．しかし，$i > N$ である i に対して，$\langle f, v_i \rangle = 0$ にすると，1 つの解 f が得られる．実は，これが $\mathcal{K}f = g$ を満たす f の中でノルム最小のものである．

$$f = f_N = \sum_{i=1}^{N} \sigma_i^{-1} \langle g, u_i \rangle v_i \tag{24}$$

従って，式 (24) をこの解の近似として使うことができることがわかる．

3. Lanczos 法 + TSVD 法

最も自然な離散化の考え方は，Riemann 和を使って式 (1) を離散化し，線形方程式 $A\boldsymbol{x} = \boldsymbol{b}$ を得ることである．連続の場合と同様に行列 A の特異値と特異ベクトルを用いて解 $\boldsymbol{x}$ の近似値を求める．

$$\boldsymbol{x}_k = \sum_{i=1}^{k} \sigma_i^{-1} (\boldsymbol{u}_i^* \boldsymbol{b}) \boldsymbol{v}_i \tag{25}$$

ここで，ほんの一部の特異値と特異ベクトルが必要なので，行列 A に対して Lanczos 法が使える．多数の特異値が 0 に近いので，一般に Lanczos 法の収束が速い．

Lanczos 法は任意のベクトル $\hat{\boldsymbol{v}}_1$ と $\hat{\boldsymbol{u}}_1 = A\hat{\boldsymbol{v}}_1 / \|A\hat{\boldsymbol{v}}_1\|$ から始まり，次の反復を行うことになる [3].

$$\begin{aligned} \hat{\boldsymbol{v}}_i &= A^* \hat{\boldsymbol{u}}_{i-1} - \hat{\boldsymbol{v}}_{i-1} \alpha_{i-1}, \\ \|\hat{\boldsymbol{v}}_i\| &= \beta_{i-1}, \hat{\boldsymbol{v}}_i = \hat{\boldsymbol{v}}_i / \beta_{i-1} \\ \hat{\boldsymbol{u}}_i &= A\hat{\boldsymbol{v}}_i - \hat{\boldsymbol{u}}_{i-1} \beta_{i-1} \\ \|\hat{\boldsymbol{u}}_i\| &= \alpha_i, \hat{\boldsymbol{u}}_i = \hat{\boldsymbol{u}}_i / \alpha_i \end{aligned} \tag{26}$$

ただし，ある β_m が十分小さい場合に収束したと判定する．この時，次式が得られる．

$$\begin{pmatrix} & A^* \\ A & \end{pmatrix} \begin{pmatrix} \hat{V}_m & \\ & \hat{U}_m \end{pmatrix} = \begin{pmatrix} \hat{V}_m & \\ & \hat{U}_m \end{pmatrix} \begin{pmatrix} & H_m \\ T_m & \end{pmatrix}$$

ただし，T_m は各 α_i と β_i からなる次のような 2 重対角行列で，H_m は T_m の転置である．

$$T_m = \begin{pmatrix} \alpha_1 & \beta_1 & & \\ & \alpha_2 & \ddots & \\ & & \ddots & \beta_{m-1} \\ & & & \alpha_m \end{pmatrix}, H_m = T_m^* \tag{27}$$

通常，m は大きくないので，T_m の一般の SVD： $T_m = U\Sigma_m V^*$ が簡単に計算できる．最後に $U_m = \hat{U}_m U, V_m = \hat{V}_m V$ とおけば，行列 A の部分的な SVD が得られる．

$$AV_m = U_m\Sigma_m \tag{28}$$

ここで得られた特異値と特異ベクトルを用いて，式 (25) で解 $\boldsymbol{x}$ の近似値を求めることができる．

しかし，Riemann 和は積分の離散化法として精度がそれほどよいわけではない．より精度のいい積分公式は，Boole の公式が考えられる．Boole の公式より x_0 と x_n の間の関数 $f(x)$ の積分は，2 つの端点とその間の n 等分点 $x_1, x_2, \cdots, x_{n-1}$ における関数値を使えば次のような近似ができる．ただし，$h = 1/n$.

$$\int_a^b f(x)dx \approx \frac{2}{45}h\sum_{i=0}^{n} w_i f(x_i)$$
$$w_i = \begin{cases} 7 & (i = 0 \text{ or } n) \\ 14 & (i = 4k) \\ 32 & (i = 4k+1 \text{ or } 4k+3) \\ 12 & (i = 4k+2) \end{cases} \tag{29}$$

ただし，$k \in \mathbb{N}$. 式 (29) に従って離散化を行うなら，まず，$x_0 = y_0 = a$, $x_n = y_n = b$ とし，行列 A を各 $K(x_i, y_j)(i, j = 0, 1, \cdots, n)$ からなる行列とする．そして，重みを表す行列 W を次のように定義する．

$$W = \mathrm{diag}\{w_0, w_1, \cdots, w_n\} \tag{30}$$

ここで，積分作用素 $\mathcal{K}$ の離散化は AW で表すことができ，得られた線型方程式は $AW\boldsymbol{x} = \boldsymbol{b}$ となる．

一見，このアルゴリズムの行列 A のところに AW を代入して計算していけば良いように見えるが，それを実際に実行してみると，望ましい結果を得ることができなかった．その理由は，もし行列 AW を前のアルゴリズムに直接代入すると，作用素 $\mathcal{K}^*$ に対応する行列が $(AW)^* = WA^*$ になる．しかし，$\mathcal{K}^*$ は，次式で記述できる．

$$\mathcal{K}^* g = \int_a^b K(x, y)g(x)dx \tag{31}$$

故に，もし式 (31) の積分公式で離散化を行うと，作用素 $\mathcal{K}^*$ に対応する行列は A^*W になるはずである．AW が対称ではない限り，両方が一致しない．そのため，ここで矛盾が生じる．

この問題を解決するために，Lanczos 法の修正を考える．まず，積分公式に対応できる W で定まる内積とノルムを，次のように定義する．

$$\langle \boldsymbol{u}, \boldsymbol{v}\rangle_W = \boldsymbol{u}^* W\boldsymbol{v}, \quad ||\boldsymbol{u}||_W = \sqrt{\langle \boldsymbol{u}, \boldsymbol{u}\rangle}, \quad \forall \boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$$

この新しい内積とノルムの導入によって，Lanczos 法の反復は任意のベクトル $\hat{\boldsymbol{v}}_1$ と $\hat{\boldsymbol{u}}_1 = AW\hat{\boldsymbol{v}}_1/\|AW\hat{\boldsymbol{v}}_1\|_W$ から始まり，毎回の反復は次のようになる．

$$\begin{aligned} \hat{\boldsymbol{v}}_i &= A^*W\hat{\boldsymbol{u}}_{i-1} - \hat{\boldsymbol{v}}_{i-1}\alpha_{i-1}, \\ \|\hat{\boldsymbol{v}}_i\|_W &= \beta_{i-1}, \hat{\boldsymbol{v}}_i = \hat{\boldsymbol{v}}_i/\beta_{i-1} \\ \hat{\boldsymbol{u}}_i &= AW\hat{\boldsymbol{v}}_i - \hat{\boldsymbol{u}}_{i-1}\beta_{i-1} \\ \|\hat{\boldsymbol{u}}_i\|_W &= \alpha_i, \hat{\boldsymbol{u}}_i = \hat{\boldsymbol{u}}_i/\alpha_i \end{aligned} \tag{32}$$

式 (32) で生成される各 $\hat{\boldsymbol{u}}_i$ と $\hat{\boldsymbol{v}}_i$ は，本当に正規直交なベクトルなのかをここで確かめる．あるステップですでに得られた $\hat{\boldsymbol{v}}_1, \hat{\boldsymbol{v}}_2, \cdots, \hat{\boldsymbol{v}}_i$ と $\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2, \cdots, \hat{\boldsymbol{u}}_i$ が正規直交であると仮定する．そして，式 (32) で新しい $\hat{\boldsymbol{v}}_{i+1}$ が得られる．$j = 1, 2, \cdots, i-2$ に対して，$\hat{\boldsymbol{v}}_{i+1}$ と $\hat{\boldsymbol{v}}_j$ の内積を取ると，次式のようになる．

$$\begin{aligned} \langle \hat{\boldsymbol{v}}_{i+1}, \hat{\boldsymbol{v}}_j\rangle_W &= \hat{\boldsymbol{v}}_j^* W\hat{\boldsymbol{v}}_{i+1} = \frac{1}{\beta_i}\hat{\boldsymbol{v}}_j^* W(A^*W\hat{\boldsymbol{u}}_i - \alpha_i\hat{\boldsymbol{v}}_i) \\ &= \frac{1}{\beta_i}(AW\hat{\boldsymbol{v}}_j)^* W\hat{\boldsymbol{u}}_i = \frac{1}{\beta_i}(\alpha_j\hat{\boldsymbol{u}}_j^* + \beta_{j-1}\hat{\boldsymbol{u}}_{j-1}^*)W\hat{\boldsymbol{u}}_i \\ &= 0 \end{aligned} \tag{33}$$

また，$\hat{\boldsymbol{v}}_{i+1}$ と $\hat{\boldsymbol{v}}_i$ の内積を取ると次になる．

$$\begin{aligned} \langle \hat{\boldsymbol{v}}_{i+1}, \hat{\boldsymbol{v}}_i\rangle &= \hat{\boldsymbol{v}}_i^* W\hat{\boldsymbol{v}}_{i+1} = \frac{1}{\beta_i}\hat{\boldsymbol{v}}_i^* W(A^*W\hat{\boldsymbol{u}}_i - \alpha_i\hat{\boldsymbol{v}}_i) \\ &= \frac{1}{\beta_i}((AW\hat{\boldsymbol{v}}_i)^* W\hat{\boldsymbol{u}}_i - \alpha_i) \\ &= \frac{1}{\beta_i}((\alpha_i\hat{\boldsymbol{u}}_i^* + \beta_{i-1}\hat{\boldsymbol{u}}_{i-1}^*)W\hat{\boldsymbol{u}}_i - \alpha_i) \\ &= \frac{1}{\beta_i}(\alpha_i - \alpha_i) = 0 \end{aligned} \tag{34}$$

式 (33) と式 (34) より，新しく得られた $\hat{\boldsymbol{v}}_{i+1}$ が前のすべての $\hat{\boldsymbol{v}}_j (j \le i)$ と直交する．同様な議論で，$\hat{\boldsymbol{u}}_{i+1}$ はすべての $\hat{\boldsymbol{u}}_j (j \le i)$ と直交することがわかる．従って，式 (33) で生成される $\{\hat{\boldsymbol{u}}_i\}_{i=1}^n$ と $\{\hat{\boldsymbol{v}}_i\}_{i=1}^n$ は，それぞれ正規直交列になる．

前と同じ，ある β_i が十分小さい時，アルゴリズムが収束したと判定する．収束した時，前と似たような式が得られる．

$$\begin{pmatrix} & A^*W \\ AW & \end{pmatrix}\begin{pmatrix} \hat{V}_m & \\ & \hat{U}_m \end{pmatrix} = \begin{pmatrix} \hat{V}_m & \\ & \hat{U}_m \end{pmatrix}\begin{pmatrix} & H_m \\ T_m & \end{pmatrix}$$

ただし，T_m と H_m は前述と同じものである．

次は前述と同じ T_m の通常の特異値分解 $T_m = U\Sigma_m V^*$ を計算して，$U_m = \hat{U}_m U, V_m = \hat{V}_m V$ と置くと，上記の関係式を利用して，次式が得られる．

$$\begin{aligned} A^*WU_m &= A^*W\hat{U}_m U = \hat{V}_m T_m^* U \\ &= \hat{V}_m V\Sigma_m U^* U = V_m\Sigma_m \\ AWV_m &= AW\hat{V}_m V = \hat{U}_m T_m V \\ &= \hat{U}_m U\Sigma_m V^* V = U_m\Sigma_m \\ U_m^* WU_m &= U^*\hat{U}_m^* W\hat{U}_m U = U^* U = I \\ V_m^* WV_m &= V^*\hat{V}_m^* W\hat{V}_m V = V^* V = I \end{aligned}$$

従って，ここで得られた U_m, V_m, Σ_m は確かに W に導か

れる内積とノルムの意味で行列 A の特異ベクトルと特異値からなる行列である．特に，前のように互いに矛盾することがない．残りは式 (25) に沿って近似値を求めればよい．W に導かれる内積とノルムによって，Lanczos 反復を修正して，式 (29) などの積分公式を利用して離散化した問題をうまく解けばよい．

4. 数値実験

数値実験は次のような計算環境で行った．
OS: Windows 10 Home(64-bit),
CPU: Intel Core i7-6700HQ CPU @ 2.60GHz,
Memory: 16.0GB,
Program Language: MATLAB R2016a.

4.1 例題 1

次の積分方程式を考える．

$$\int_0^1 e^{xy} f(y)dy = \frac{e^{x+1}-1}{x+1}$$

ただし，真の解は $f(y) = e^y$ である．

ここで使う重みを表す行列 W は式 (30) で定義されたものを使う．$n = 2048, h = 1/n$ とすると，A と $\boldsymbol{b}$ は以下のようになる．

$$x_i = y_i = ih, \quad i = 0, 1, \cdots, n$$

$$A = \begin{pmatrix} e^{x_0y_0} & e^{x_0y_1} & \cdots & e^{x_0y_n} \\ e^{x_1y_0} & e^{x_1y_1} & \cdots & e^{x_1y_n} \\ \vdots & \vdots & \ddots & \cdots \\ e^{x_ny_0} & e^{x_ny_1} & \cdots & e^{x_ny_n} \end{pmatrix}$$

$$\boldsymbol{b} = (b_0, b_1, \cdots, b_n)^*, \; b_i = \frac{e^{x_i+1}-1}{x_i+1}, \quad i = 0, 1, \cdots, n$$

これ以降の例題も同じような方法で離散化を行うものとする．

前節で述べた古典的な Lanczos 法を適用したアルゴリズムと修正した Lanczos を適用したアルゴリズムをそれぞれ実行して，実行結果は表 1 に示した．また，各 y_i における誤差 $e(y_i) = |f_{exact}(y_i) - f_{num}(y_i)|$ を図 1 にプロットした．

表 1 と図 1 から，古典的な Lanczos 法においても，$f(y) = e^y$ のきれいな形が得られたとはいえ，細かく誤差を計算してみると，精度がそんなにいいとは言いがたい．それは比較的に誤差が大きい Riemann 和で離散化したからである．一方，Boole の公式を利用した修正版の Lanczos は，古典的な Lanczos 法に比べて 3 倍以上の精度を実現した．n と m, k の値が特に大きくないため，2 つの方法とも実行時間がかなり短くて，無視できるほどである．

4.2 例題 2

次の方程式を考える．

$$\int_0^1 \sin(xy) f(y)dy = \frac{\sin x - x\cos x}{x^2}$$

ただし，真の解は $f(y) = y$ である．

前の例題と同じ，2 つの方法で離散化を行い，古典的な Lanczs 法と修正版の実行結果を調べた．ただし，今回の右側の関数 $g(x)$ の分母が x^2 であるため，$b_0 = g(0)$ を直接計算することができないので，次のように b_0 を 0 における $g(x)$ の極限とおいて計算した．

$$b_0 = \lim_{x\to 0} \frac{\sin x - x\cos x}{x^2} = \lim_{x\to 0} \frac{x\sin x}{2x} = \lim_{x\to 0} \frac{\sin x}{2} = 0$$

例題 2 の実行結果を表 2 に記述した．また，2 つの方法で求めた結果に含まれる誤差は図 3 に示した．例題 1 と同様に，修正した Lanczos 法は同じくらいの時間で通常の 3 倍以上の精度が得られた．

4.3 例題 3

次の方程式を考える．

$$\int_0^1 (x-y)^2 f(y)dy = \frac{5x^2-5x+3}{15}$$

ただし，真の解は $f(y) = 16y^2 - 16y + 3$ である．

2 つの計算誤差の結果を図 3 に示した．次に計算時間の比較を表 3 に記述した．今回の場合，求まった特異値の個数が少なくて，古典的な Lanczos 法による解の誤差は前の例題とそれほど変わらないが，残差の小さい解を得ることに失敗した．一方，修正した Lanczos 法を利用すると，残差が依然として小さいだけでなく，ほぼ完全に真の解と一致する計算解が得られた．

参考文献

[1] Silvia Noschese, Lothar Reichel, “A modified truncated singular value decomposition method for discrete ill-posed problems,” Numer. Linear Algebra Appl. 2014; 21: pp. 813–822.
[2] Robert Craig Schmidt, “The numerical solution of linear first kind Fredholm integral eqnarrays using an iterative method,” Iowa State University Digital Repository. 1987; pp. 73–81.
[3] Lloyd N. Trefethen, “Numerical Linear Algebra,” SIAM. 2007; pp. 414–415.
[4] Hong du, “Approximate solution of the Fredholm integral eqnarray of the first kind in a reproducing kernel Hilbert space,” An International Journal of Rapid Publication 2008: 21, pp. 617–623.
[5] Martin Hanke,“A Taste of Inverse Problems Basic Theory and Examples,” SIAM. 2017; pp. 12-13,130–132.
[6] Paul Garrett: “Compact operators, Hilbert-Schmidt operators,” March 2012. pp. 2–3.

表 1: 例題 1 の実行結果の比較

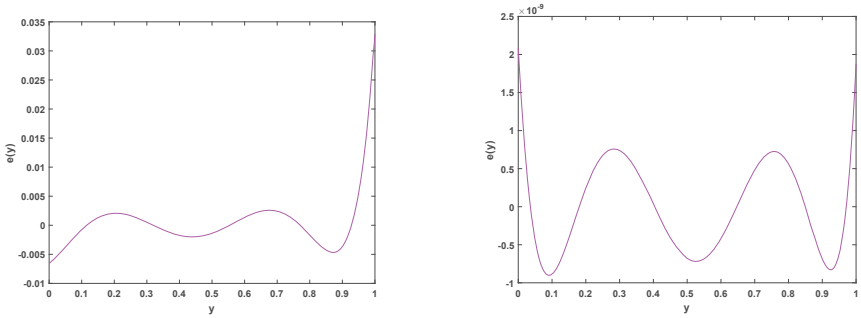
	m	k	残差	相対誤差	実行時間 (sec)
古典的な Lanczos 法	10	7	2.5300×10^{-15}	2.7100×10^{-3}	0.2331s
修正 Lanczos 法	10	6	8.9763×10^{-16}	3.3327×10^{-10}	0.2030s

表 2: 例題 2 の実行結果の比較

	m	k	残差	相対誤差	実行時間 (sec)
古典的な Lanczos 法	7	4	9.5246×10^{-14}	2.5202×10^{-3}	0.1953s
修正 Lanczos 法	7	4	1.4434×10^{-14}	2.9982×10^{-10}	0.1667s

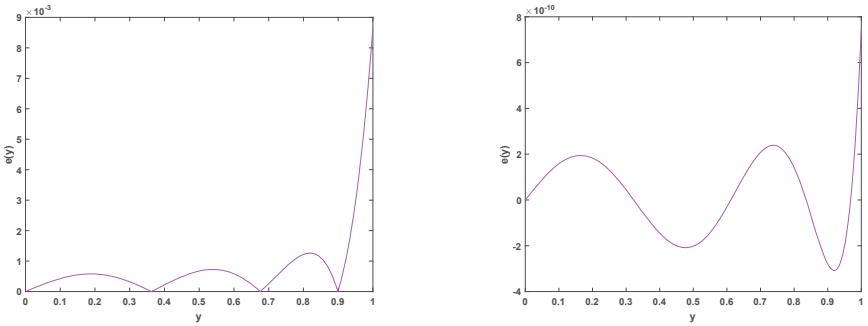
表 3: 例題 3 の実行結果の比較

	m	k	残差	相対誤差	実行時間 (sec)
古典的な Lanczos 法	2	2	3.2057×10^{-4}	1.8213×10^{-3}	0.1175s
修正 Lanczos 法	2	2	2.3977×10^{-15}	1.2275×10^{-15}	0.0766s



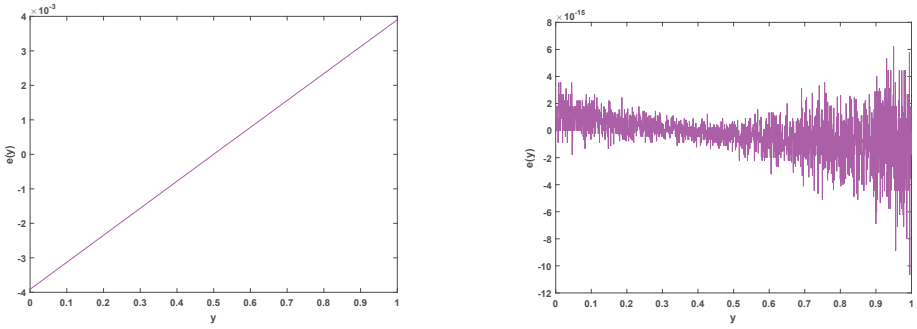
(a) 古典的な Lanczos 法による解の誤差　(b) 修正 Lanczos 法による解の誤差

図 1: 例題 1 の Lanczos 法による解の計算誤差の結果



(a) 古典的な Lanczos 法による解の誤差　(b) 修正 Lanczos 法による解の誤差

図 2: 例題 2 の Lanczos 法による解の計算誤差の結果



(a) 古典的な Lanczos 法による解の誤差　(b) 修正 Lanczos 法による解の誤差

図 3: 例題 3 の Lanczos 法による解の計算誤差の結果