

波動方程式の逆問題解析高速化のための順問題に対する 直接的高速数値解法の開発

Numerical method for the forward problem in linear wave equation to accelerate the
numerical resolution for the dynamical inverse problem

代田健二¹⁾
Kenji Shiota

¹⁾博(理) 愛知県立大学 情報科学部 教授 (〒 480-1189 愛知県長久手市茨ヶ廻間 1522 番 3, E-mail: shiota@ist.aichi-pu.ac.jp)

In this research, we consider the numerical method for the forward problem in linear wave equation. The aim to this study is to do a basic research for the numerical technique to get fast a solution to the dynamical inverse problem. We apply the finite difference or finite element approximation to the discretization in space. We get a linear system with the Kronecker product type coefficient matrix by using the spectral collocation method with the Gauss-Lobatto points for the time discretization. We introduce the BiCG type iterative method for the Kronecker product type coefficient matrix to solve our linear system. We show the effectiveness of our method by some numerical experiments.

Key Words : Linear wave equation, Spectral collocation method, Kronecker product type matrix, BiCG type method

1. はじめに

本研究では、波動方程式族の逆問題に対する数値的再構成法高速化の基礎研究として、スカラー波動方程式の初期値境界値問題に対する数値解法について考察する。

著者は、これまでに波動場における逆問題に対する数値解法についての研究を実施してきた。主に対象としてきた係数同定問題に対しては、その非線形性ゆえに一般には勾配法などの反復解法を用いられることが多く、著者も随伴変数法を基礎とした反復法を提案し、一定精度の同定結果を得ることができた [1]。一方、随伴変数法を基礎とした反復法を用いているため、各反復ごとに波動方程式の順問題を 2 回解く必要があり、実用問題、特に空間 3 次元問題への適用には計算速度の観点で課題が残る結果となっていた。

本研究の目的は、実用問題に対する逆問題解析の数値的再構成法高速化の基礎研究として、波動方程式の順問題に対する高速数値解法の開発を行うことである。本研究では、次のスカラー波動方程式の初期値境界値問題を対象とする。

$$\rho(\mathbf{x}) \frac{\partial^2 u}{\partial t^2} - \nabla \cdot (K(\mathbf{x}) \nabla u) = f \quad \text{in } \Omega \times (0, T). \quad (1)$$

ここで領域 $\Omega \subset \mathbb{R}^3$ は、有界かつ境界は区分的に滑らかであるとする。 $K(\mathbf{x})$ は体積弾性率、 $\rho(\mathbf{x})$ は密度であり、どちらも既知であるとする。また初期値および境界値は、次の通りに与えられていると仮定する。

$$u(0) = u_0, \quad \frac{\partial u}{\partial t}(0) = v_0 \quad \text{in } \Omega,$$

$$u = \bar{u} \quad \text{on } \Gamma_D \times (0, T),$$

$$K \frac{\partial u}{\partial \nu} = \bar{q} \quad \text{on } \Gamma_N \times (0, T).$$

ここで、 $\partial\Omega = \overline{\Gamma_D \cup \Gamma_N}$, $\Gamma_D \cap \Gamma_N = \emptyset$ とする。

本研究では、スペクトル選点法を時間方向離散化に用いることで時間方向の並列計算を可能にしたスペクトル選点-多点差分法 [2] のアイディアを採用する。先行研究では高精度化を目的としてため空間方向離散化に任意多点差分法 [3] を用いたが、本研究では差分法・有限要素法などの従来の離散化手法を採用する。導出された Kronecker 積構造の係数行列を持つ連立一次方程式を逆 **vec** 作用素により同値な行列方程式へと変換し、その方程式を数値的に解くことで近似解を得る。行列方程式を数値的に解く方法として、行列構造を維持可能な BiCG 系解法を採用し、逆 **vec** 作用素を用いることにより Kronecker 積 BiCG 系解法を導出する。数値実験により、提案手法の有効性を検証する。

2. スペクトル選点法を用いた離散化

波動方程式 (1) に対する空間方向離散化には、差分法・有限要素法などの一般的に用いられる手法を採用する。このとき得られる半離散化方程式は、次のようになる。

$$\frac{d^2 \mathbf{u}}{dt^2}(t) - A \mathbf{u}(t) = \mathbf{f}(t), \quad t \in (0, T], \quad (2)$$

$$\mathbf{u}(0) = \mathbf{u}_0, \quad \frac{d\mathbf{u}}{dt}(0) = \mathbf{v}_0.$$

ただし、

$$\begin{aligned} \mathbf{u}(t) &= (u(\mathbf{x}^{(1)}, t), \dots, u(\mathbf{x}^{(N)}, t))^T, \\ \mathbf{u}_0 &= (u_0(\mathbf{x}^{(1)}), \dots, u_0(\mathbf{x}^{(N)}))^T, \\ \mathbf{v}_0 &= (v_0(\mathbf{x}^{(1)}), \dots, v_0(\mathbf{x}^{(N)}))^T \end{aligned}$$

であり、 $\cdot$ は時間方向微分を意味している。また、 $\mathbf{x}^{(i)}$ ($i = 1, 2, \dots, N$) は空間分割による格子点または節点で

ある。

半離散化方程式 (2) の時間方向近似として、Runge-Kutta 法や Newmark 法に代表される数値積分法が一般的に用いられている。しかしこれらの方法では、逐次近似ベクトルを求めていくため、時間方向を並列化することが困難である。そこで本研究では、選点法により時間方向を離散化することで大規模連立一次方程式を導出し、その方程式を並列計算することで高速化の実現を目指すことにする。ここで選点法としては、高精度近似法として知られる Gauss-Lobatto 選点を用いた Chebyshev 多項式によるスペクトル選点法 [4] を採用する。そのために、まず (2) を 1 階常微分方程式に変換する。

$$\tilde{\mathbf{u}} = \begin{pmatrix} \mathbf{u} \\ \frac{d\mathbf{u}}{dt} \end{pmatrix}, \quad \tilde{\mathbf{f}} = \begin{pmatrix} \mathbf{0}_N \\ \mathbf{f} \end{pmatrix}, \quad \tilde{\mathbf{w}}_0 = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{v}_0 \end{pmatrix}$$

とすると、(2) は次のとおりに書き換えることができる。

$$\frac{d\tilde{\mathbf{u}}}{dt}(t) - \tilde{\mathbf{A}}\tilde{\mathbf{u}}(t) = \tilde{\mathbf{f}}(t), \quad t \in (0, T],$$

$$\tilde{\mathbf{u}}(0) = \tilde{\mathbf{w}}_0.$$

ただし、

$$\tilde{\mathbf{A}} = \begin{pmatrix} \mathbf{O}_N & \mathbf{I}_N \\ \mathbf{A} & \mathbf{O}_N \end{pmatrix}.$$

$\tilde{\mathbf{u}}(t)$ を、Gauss-Lobatto 選点を用いた Chebyshev 多項式で近似する。ここで Gauss-Lobatto 選点は、次のとおりである。

$$\hat{t}_k = \cos\left(\frac{k\pi}{N_T}\right) \quad (k = 0, 1, \dots, N_T).$$

N_T は与えられた時間方向分割数である。このとき、 $\tilde{\mathbf{u}}$ の Chebyshev 近似 $\tilde{\mathbf{u}}_{N_T}$ は、次のとおりとなる。

$$\tilde{\mathbf{u}}_{N_T}(t) = \sum_{k=0}^{N_T} T_k\left(\frac{2t-T}{T}\right) \mathbf{c}_k.$$

$T_k(t)$ は k 次 Chebyshev 多項式であり、係数ベクトル $\mathbf{c}_k$ は、 $\tilde{\mathbf{u}}_j = \tilde{\mathbf{u}}(t_j)$ とするとき、

$$\mathbf{c}_k = \sum_{j=0}^{N_T} \frac{1}{d_j} \tilde{\mathbf{u}}_j \cos(k\theta_j),$$

$$\bar{d}_j = \begin{cases} 2 & (j = 0, N_T) \\ 1 & (1 \leq j \leq N_T - 1) \end{cases},$$

$$\theta_j = \frac{j\pi}{N_T}, \quad t_j = \frac{T}{2}(\hat{t}_j + 1) \quad (j = 0, 1, \dots, N_T)$$

により求められる。このとき、 $\frac{d\tilde{\mathbf{u}}}{dt}(t)$ の近似 $\frac{d\tilde{\mathbf{u}}_{N_T}}{dt}(t)$ は

$$\frac{d\tilde{\mathbf{u}}_{N_T}}{dt}(t_l) = \frac{2}{T} \sum_{j=0}^{N_T} (D_l)_{l,j} \tilde{\mathbf{u}}_j \quad (l = 0, 1, \dots, N_T)$$

となる。ただし D_l は、各成分が

$$(D_l)_{l,j} = \begin{cases} \frac{\bar{d}_l}{\bar{d}_j} \frac{(-1)^{l+j+1}}{2 \sin \frac{(l+j)\pi}{2N_T} \sin \frac{(l-j)\pi}{2N_T}} & (l \neq j) \\ -\frac{\cos \frac{j\pi}{N_T}}{2 \sin^2 \frac{j\pi}{N_T}} & (1 \leq l = j \leq N_T - 1) \\ \frac{2N_T^2 + 1}{6} & (l = j = 0) \\ -\frac{6}{2N_T^2 + 1} & (l = j = N_T) \end{cases}$$

により定義される行列である。 $\tilde{\mathbf{u}}_{N_T}(t_l) = \tilde{\mathbf{u}}_l$ より、(2) をスペクトル選点法により近似すると、

$$\begin{aligned} \frac{d\tilde{\mathbf{u}}}{dt}(t_l) - \tilde{\mathbf{A}}\tilde{\mathbf{u}}(t_l) &\approx \frac{d\tilde{\mathbf{u}}_{N_T}}{dt}(t_l) - \tilde{\mathbf{A}}\tilde{\mathbf{u}}_{N_T}(t_l) \\ &= \frac{2}{T} \sum_{j=0}^{N_T} (D_l)_{l,j} \tilde{\mathbf{u}}_j - \tilde{\mathbf{A}}\tilde{\mathbf{u}}_l \\ &= \sum_{j=0}^{N_T} \left\{ \frac{2}{T} (D_l)_{l,j} - \delta_{lj} \tilde{\mathbf{A}} \right\} \tilde{\mathbf{u}}_j \end{aligned}$$

($l = 0, 1, 2, \dots, N_T - 1$). δ_{lj} はクロネッカーのデルタである。 $\tilde{\mathbf{u}}_{N_T}(t_{N_T}) = \tilde{\mathbf{u}}(0) = \tilde{\mathbf{w}}_0$ であることから

$$\sum_{j=0}^{N_T-1} \left\{ \frac{2}{T} (D_l)_{l,j} - \delta_{l,j} \tilde{\mathbf{A}} \right\} \tilde{\mathbf{u}}_j = \tilde{\mathbf{F}}(t_l) - \frac{2}{T} (D_l)_{l,N_T} \tilde{\mathbf{w}}_0.$$

$\hat{\mathbf{u}} = (\tilde{\mathbf{u}}_0^T, \tilde{\mathbf{u}}_1^T, \dots, \tilde{\mathbf{u}}_{N_T-1}^T)^T$ とすると、 $2NN_T$ 次連立一次方程式を得ることができる。

$$\left(\frac{2}{T} \tilde{\mathbf{D}}_l \otimes \mathbf{I}_{2N} - \mathbf{I}_{N_T} \otimes \tilde{\mathbf{A}} \right) \hat{\mathbf{u}} = \hat{\mathbf{f}}. \quad (3)$$

ただし、

$$\tilde{\mathbf{D}}_l = ((D_l)_{l,j})_{l,j=0,1,\dots,N_T-1} \in \mathbb{R}^{N_T \times N_T},$$

$$\hat{\mathbf{f}} = (\tilde{\mathbf{f}}(t_l) - \frac{2}{T} (D_l)_{l,N_T} \tilde{\mathbf{w}}_0)_{l=0}^{N_T-1} \in \mathbb{R}^{2N \cdot N_T}$$

である。また $\otimes$ は行列の Kronecker 積 [5] である: $A = (a_{ij}) \in \mathbb{R}^{m \times n}$, $B = (b_{lk}) \in \mathbb{R}^{p \times q}$ とするとき、

$$A \otimes B := \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{pmatrix} \in \mathbb{R}^{mp \times nq}.$$

3. クロネッカー積構造 BiCG 系解法

一定精度の近似解を得るための空間分割数を使用すると、離散化して得られた連立一次方程式 (3) の係数行列は、その構造ゆえに大規模となる。一方、空間方向の離散化に差分法や有限要素法を用いると、係数行列の構造は疎となる。例えば、空間方向に差分法を用いたときの係数行列は図 1 のとおりになる。

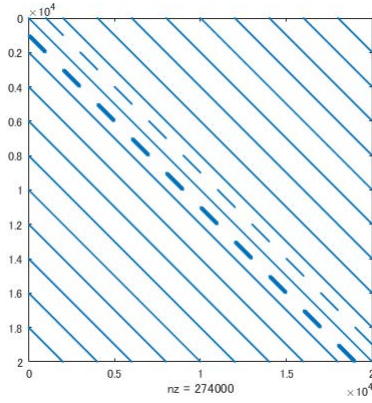


図-1 係数行列の構造例

係数行列の構造を維持可能な連立一次方程式の解法として代表的な方法としては、SOR 法などの定常反復解法や BiCG 法などの非定常反復解法がある。一方、係数行列は疎とはいえ、 D_t の要素数 $\times 2N + \tilde{A}$ の要素数 $\times N_T$ を格納するメモリが必要となり、大規模問題を対象とする場合は何らかの対処法を考えなければならない。そこで本研究では、その対処法として大橋ら [6] が提案した逆 vec 作用素を用いたアイデアを採用する。

$\mathbf{x} = (x_{11}, x_{21}, \dots, x_{n1}, x_{12}, \dots, x_{nm})^T \in \mathbb{R}^{n \cdot m}$ に対して $\text{vec}^{-1} : \mathbb{R}^{n \cdot m} \rightarrow \mathbb{R}^{n \times m}$ を

$$\text{vec}^{-1}(\mathbf{x}) = \begin{pmatrix} x_{11} & x_{21} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{pmatrix} \in \mathbb{R}^{n \times m}$$

と定義する。この vec^{-1} を、逆 vec 作用素と呼ぶ。逆 vec^{-1} 作用素を用いると、Kronecker 積構造の行列とベクトルの積は、次のように行列積へと変換することができる [5]: $A \in \mathbb{R}^{m \times m}$, $B \in \mathbb{R}^{n \times n}$ とする。このとき、

$$\text{vec}^{-1}((A \otimes B)\mathbf{x}) = BXA^T.$$

ただし、 $X = \text{vec}^{-1}(\mathbf{x})$ である。この結果と逆 vec 作用素の線形性より、(3) は vec^{-1} を用いて次の同値な行列方程式へと変換される。

$$U\tilde{D}_t^T - \tilde{A}U = F. \quad (4)$$

ただし、

$$U = \text{vec}^{-1}(\hat{\mathbf{u}}) \in \mathbb{R}^{2N \times N_T}, \quad F = \text{vec}^{-1}(\hat{\mathbf{f}}) \in \mathbb{R}^{2N \times N_T},$$

$$\tilde{D}_t = \frac{2}{T} D_t.$$

この変換により、係数行列に対する使用メモリ量を従来法と同程度にすることが可能になる。

行列方程式 (4) を解く方法として、非定常反復解法である BiCG 系解法 [7] を用いることとする。元の連立一次方程式 (3) に対して、BiCG 系解法の一つである GPBiCGSafe 法 [8] を適用したアルゴリズムは次のとおりになる。

Algorithm 1 GPBiCGSafe 法

Given $\hat{\mathbf{u}}_0$; $\tilde{G} = \tilde{D}_t \otimes I_{2N} - I_{N_T} \otimes \tilde{A}$;
 $\mathbf{r}_0 = \hat{\mathbf{f}} - \tilde{G}\hat{\mathbf{u}}_0$;
 $\mathbf{r}'_0$ を $(\mathbf{r}'_0, \mathbf{r}_0) \neq 0$ を満たすように適当に選択
 $\mathbf{p}_{-1} = \mathbf{q}_{-1} = \mathbf{z}_{-1} = \mathbf{0}$; $\beta_{-1} = 0$; $k = 0$;
repeat
 $\mathbf{p}_k = \mathbf{r}_k + \beta_{k-1}(\mathbf{p}_{k-1} - \mathbf{q}_{k-1})$;
 $\alpha_k = \frac{(\mathbf{r}'_0, \mathbf{r}_k)}{(\mathbf{r}'_0, \tilde{G}\mathbf{p}_k)}$;
if $k = 0$ **then**
 $\zeta_k = \frac{(\tilde{G}\mathbf{r}_k, \mathbf{r}_k)}{(\tilde{G}\mathbf{r}_k, \tilde{G}\mathbf{r}_k)}$; $\eta_k = 0$;
else
 $\zeta_k = \frac{\|\tilde{G}\mathbf{z}_{k-1}\|_2^2(\tilde{G}\mathbf{r}_k, \mathbf{r}_k) - (\tilde{G}\mathbf{z}_{k-1}, \mathbf{r}_k)(\tilde{G}\mathbf{r}_k, \tilde{G}\mathbf{z}_{k-1})}{\|\tilde{G}\mathbf{r}_k\|_2^2\|\tilde{G}\mathbf{z}_{k-1}\|_2^2 - (\tilde{G}\mathbf{z}_{k-1}, \tilde{G}\mathbf{r}_k)(\tilde{G}\mathbf{r}_k, \tilde{G}\mathbf{z}_{k-1})}$;
 $\eta_k = \frac{\|\tilde{G}\mathbf{r}_k\|_2^2(\tilde{G}\mathbf{z}_{k-1}, \mathbf{r}_k) - (\tilde{G}\mathbf{z}_{k-1}, \tilde{G}\mathbf{r}_k)(\tilde{G}\mathbf{r}_k, \mathbf{r}_k)}{\|\tilde{G}\mathbf{r}_k\|_2^2\|\tilde{G}\mathbf{z}_{k-1}\|_2^2 - (\tilde{G}\mathbf{z}_{k-1}, \tilde{G}\mathbf{r}_k)(\tilde{G}\mathbf{r}_k, \tilde{G}\mathbf{z}_{k-1})}$;
end if
 $\mathbf{q}_k = \zeta_k \tilde{G}\mathbf{p}_k + \eta_k(\tilde{G}\mathbf{z}_{k-1} + \beta_{k-1}\mathbf{q}_{k-1})$; $\mathbf{s}_k = \mathbf{r}_k - \alpha_k \tilde{G}\mathbf{p}_k$;
 $\mathbf{z}_k = \zeta_k \mathbf{r}_k + \eta_k \mathbf{z}_{k-1} - \alpha_k \mathbf{q}_k$;
 $\hat{\mathbf{u}}_{k+1} = \hat{\mathbf{u}}_k + \alpha_k \mathbf{p}_k + \mathbf{z}_k$; $\mathbf{r}_{k+1} = \mathbf{s}_k - \tilde{G}\mathbf{z}_k$;
 $\beta_k = \frac{\alpha_k}{\zeta_k} \cdot \frac{(\mathbf{r}'_0, \mathbf{r}_{k+1})}{(\mathbf{r}'_0, \mathbf{r}_k)}$; $k = k + 1$;
until $\|\mathbf{r}_k\|_2 \leq \varepsilon \|\hat{\mathbf{f}}\|_2$

このアルゴリズムにおける Kronecker 積構造行列とベクトルの積、そしてベクトルを逆 vec 作用素により変換することで、行列方程式 (4) に対するアルゴリズムを導出できる。

Algorithm 2 Kronecker 積 GPBiCGSafe 法

Given $U, P, Q, S, Z = O_{2N, N_T}$;
 $F = \text{vec}^{-1}(\hat{\mathbf{f}}) \in \mathbb{R}^{2N \times N_T}$;
 $R = F - (U\tilde{D}_t^T - \tilde{A}U)$; $R' = R$; $r = R'$; $\beta = 0$; $k = 0$;
repeat
 $P = R + \beta \frac{(P - Q)}{r}$;
 $\alpha = \frac{R' : (P\tilde{D}_t^T - \tilde{A}P)}{R' : (P\tilde{D}_t^T - \tilde{A}P)}$;
if $k = 0$ **then**
 $\zeta = \frac{(R\tilde{D}_t^T - \tilde{A}R) : R}{(R\tilde{D}_t^T - \tilde{A}R) : (R\tilde{D}_t^T - \tilde{A}R)}$; $\eta = 0$;
else
 $\zeta = \frac{\|Z\tilde{D}_t^T - \tilde{A}Z\|_F^2((R\tilde{D}_t^T - \tilde{A}R) : R) - ((Z\tilde{D}_t^T - \tilde{A}Z) : R)((R\tilde{D}_t^T - \tilde{A}R) : (Z\tilde{D}_t^T - \tilde{A}Z))}{\|R\tilde{D}_t^T - \tilde{A}R\|_F^2\|Z\tilde{D}_t^T - \tilde{A}Z\|_F^2 - ((Z\tilde{D}_t^T - \tilde{A}Z) : (R\tilde{D}_t^T - \tilde{A}R))((R\tilde{D}_t^T - \tilde{A}R) : (Z\tilde{D}_t^T - \tilde{A}Z))}$;
 $\eta = \frac{\|R\tilde{D}_t^T - \tilde{A}R\|_F^2\|Z\tilde{D}_t^T - \tilde{A}Z\|_F^2 - ((Z\tilde{D}_t^T - \tilde{A}Z) : (R\tilde{D}_t^T - \tilde{A}R))((R\tilde{D}_t^T - \tilde{A}R) : (Z\tilde{D}_t^T - \tilde{A}Z))}{\|R\tilde{D}_t^T - \tilde{A}R\|_F^2\|Z\tilde{D}_t^T - \tilde{A}Z\|_F^2 - ((Z\tilde{D}_t^T - \tilde{A}Z) : (R\tilde{D}_t^T - \tilde{A}R))((R\tilde{D}_t^T - \tilde{A}R) : (Z\tilde{D}_t^T - \tilde{A}Z))}$;
end if
 $Q = \zeta(P\tilde{D}_t^T - \tilde{A}P) + \eta(Z\tilde{D}_t^T - \tilde{A}Z + \beta Q)$;
 $S = R - \alpha(P\tilde{D}_t^T - \tilde{A}P)$;
 $Z = \zeta R + \eta Z - \alpha Q$;
 $U = U + \alpha P + Z$; $R = S - (Z\tilde{D}_t^T - \tilde{A}Z)$;
 $\beta = \frac{\alpha}{\zeta} \cdot \frac{R' : R}{r}$; $r = R'$; $k = k + 1$;
until $\|R\|_F \leq \varepsilon \|F\|_F$

ここで、 $\|\cdot\|_F$ は行列のフロベニウスノルムであり、 $:$ は

行列の内積

$$A : B = \sum_{i,j=1}^n a_{ij}b_{ij}$$

である。BiCG 系の他の解法 (BiCGSTAB 法, GPBiCG 法) についても, 逆 **vec** 作用素により行列方程式用アルゴリズムを導出することができる。これらのアルゴリズムは, 近似解を得るのに繰り返し計算する必要があるものの, 行列の加減算と乗算のみで構成されている。そのため, GPU での並列計算を行うことにより高速化が期待できる。

4. 数値実験

本提案手法の有効性を, 数値実験により検証する。Ω = (0, 1) × (0, 1) × (0, 1) とし, T = 1.0 とする。(1) の真の解を $u(x_1, x_2, x_3, t) = \sin 4\pi(x_1 + x_2 + x_3 - t)$ とし, ρ ≡ 1, K ≡ 1 とする。ソース項関数 f, 初期値 u_0, v_0 は与えられた真の解により求めたものを使う。境界条件は Dirichlet 境界条件のみ, すなわち $\Gamma_D = \partial\Omega, \Gamma_N = \emptyset$ とする。空間方向離散化には 7 点差分を用い, 各方向を等分割する。反復解法の収束判定定数は $\varepsilon = 1.0 \times 10^{-12}$ とし, また BiCG 系解法の停止条件は, 収束判定を満たすか, 反復回数の上限 (Kroneker 構造係数行列の次元の 2 倍: $2 \cdot (2N \cdot N_T)$) に達した場合とする。ただし, $(r'_0, r_k) = 0$ となる, すなわちアルゴリズムが破綻したときも停止する。数値計算は, ソフトウェア MATLAB (バージョン 2022b) を使用して実施する。

本数値実験では, 方法 (BiCGSTAB 法, GPBiCG 法, GPBiCGSafe 法) の収束性と, GPU を用いた並列計算による高速化可能性について検証する。なお, ここでの収束とは, 連立一次方程式に対する反復解法が正常に終了するという意味である。

(1) 反復解法の収束性

まずは, 反復解法の収束性について示す。例題に対する各反復解法の収束回数は, 表 1 のとおりである。なお, 区間分割数は各方向分割数の積の形で表しており, 表中の – は正常に終了しなかったことを意味している。N = 30 × 30 × 30, N_T = 10 としたときの, 各反復解法の残差履歴は図 2 のとおりである。

表-1 反復解法の収束性

N	N _T	BiCGSTAB	GPBiCG	GPBiCGSafe
10 × 10 × 10	10	–	608	325
20 × 20 × 20	10	–	677	575
30 × 30 × 30	10	–	1158	1057
40 × 40 × 40	10	–	–	1316
50 × 50 × 50	10	–	–	–
60 × 60 × 60	10	–	–	2528

BiCGSTAB 法はすべて場合で収束せず, また GPBiCG 法も収束しない場合が多く見られた。それに対し GPBiCGSafe 法は, 多くの場合で収束し, さらに反復回数も GPBiCG 法に比べ少ない結果となった。これらの結果

より, Kronecker 積 BiCG 系解法の中では, GPBiCGSafe 法が差分法を用いた提案手法と最も相性のよいことが示唆された。一方, GPBiCGSafe 法においても収束しない場合があり, 実用問題への適用に向けて前処理の実施など収束性の更なる向上を検討する必要がある。なお, N = 30 × 30 × 30, N_T = 10 における GPBiCG 法, GPBiCGSafe 法による近似解の相対誤差は, それぞれ 2.6002×10^{-2} , 2.6001×10^{-2} であり, 収束した場合の計算精度に違いは見られなかった。

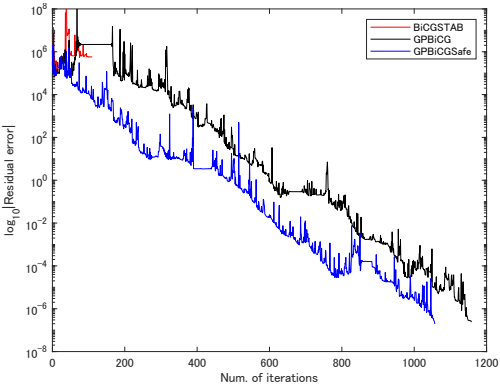


図-2 残差履歴

(2) GPU を用いた並列計算による高速化可能性

本実験では, GPU を用いた並列計算による高速化の可能性について示す。反復解法は GPBiCGSafe 法を用いることとし, CPU は Intel Xeon E5-1620 (3.5GHz), GPU は NVIDIA TITAN V (CUDA コア: 5120, FP64: 7.450TFLOPS) を搭載したワークステーションを用いて実験を実施する。このときの計算結果は, 表 2 のとおりである。本実験で行ったすべての分割数において, GPU による並列計算により計算時間を短縮することに成功した。特に分割数が多い場合, 計算時間が 1/3 から 1/4 に短縮された。このことより, GPU を用いた並列計算により高速化できる可能性が示唆された。一方, 本実験において計算終了までに 34 分程度を要したものもあり, 実用問題への適用には課題があることも示された。前処理などの反復回数を減少させるための方法検討とともに, 行列積計算の更なる高速化を検討する必要がある。

表-2 計算時間 [秒]

N	N _T	並列なし	GPU
40 × 40 × 40	10	62.60	51.58
50 × 50 × 50	20	525.18	144.73
60 × 60 × 60	10	768.70	282.49
70 × 70 × 70	25	5345.60	2047.52
80 × 80 × 80	20	4421.79	1041.00

5. まとめ

本研究では, 実用逆問題の数値的再構成法高速化の基礎研究として, 波動方程式の順問題に対する高速数

値解法について考察した。時間方向も並列計算可能なスペクトル選点-多点差分法を基礎とした離散化法により Kronecker 積構造の係数行列を持つ連立一次方程式を導出し、その方程式に逆 **vec** 作用素を適用することで、従来の方法と同程度のメモリ量で格納可能な行列方程式を導出した。導出した方程式を解くための方法として、従来の反復解法に逆 **vec** 作用素を適用することで得られる Kronecker 積構造 BiCG 系解法を導出した。数値実験により本手法の有効性は示唆されたものの、反復解法の収束性に課題が残った。また、実用問題に対する利用を考えた場合、さらなる高速化が必要なものも明らかになった。今後は、収束性の向上と高速化のため Kronecker 積構造を維持可能な前処理法の検討、GPU により並列計算の更なる高速化実現のために cuBLAS, cuSPARSE などの CUDA ライブラリを用いたプログラム開発を実施していく。

謝辞: 本研究は JSPS 科研費 18K03420 の助成を受けたものです。

参考文献

- [1] Kurashiki, D. and Shirota, K.: H^2 gradient method for the coefficient identification problem in a partial differential equation, *JSIAM Lett.*, Vol. 10, pp.37-40, 2018.
- [2] 代田健二, 曾我部知広: 高精度逆問題解析に援用可能な波動方程式の順問題に対する直接的数値解法の開発, 日本応用数理学会 2019 年度年会講演予稿集, pp.131-132, 2019.
- [3] Iijima, K. and Onishi, K.: Lattice-free finite difference method for numerical solution of inverse heat conduction problem, *Inv. Probs. Sci. and Eng.*, Vol. 15, pp.93-106, 2007.
- [4] Shen, J. et al.: Spectral Methods: Algorithms, Analysis and Applications, Springer Series in Computational Mathematics 41, Springer, 2011.
- [5] 山本哲朗, 行列解析の基礎 – Advanced 線形代数 -, SGC ライブラリ 79, サイエンス社, 2010.
- [6] Ohashi, A. and Sogabe, T.: On computing maximum/minimum singular values of a generalized tensor sum, *Electron. Trans. Numer. Anal.*, Vol. 43, pp.244-254, 2015.
- [7] 杉原正顕, 室田一雄: 線形計算の数理, 岩波書店, 2009.
- [8] 藤野清次, 降順に展開した漸化式を使用しない GPBiCGSafe 法の考案, 数理解析研究所講究録, Vol.1791, pp.31-36, 2012.