

2次元スカラー波動問題における時間域境界積分方程式法の ACAを用いた高速解法

A FAST METHOD OF TIME DOMAIN BIEM FOR SCALAR WAVE PROPAGATION IN 2D USING ACA

吉川 仁¹⁾, 山本 悟詞²⁾

Hitoshi YOSHIKAWA and Satoshi YAMAMOTO

- 1) 京都大学大学院情報学研究科 (〒 606-8501 京都市左京区吉田本町, E-mail: yoshikawa.hitoshi.5u@kyoto-u.ac.jp)
2) フラットシステムソリューションズ (〒 160-0023 東京都新宿区西新宿 3-1-5 新宿嘉泉ビル 8F)

Time domain boundary integral equation method (TD-BIEM) for 2-D scalar wave problems is considered. In scalar wave problems in 2-D, the coefficient matrices of the discretized boundary integral equation are initially sparse but they become dense with time step since the waves from the source elements propagate with the certain wave velocity. In this study, TD-BIEM with H-matrix is considered. The boundary of the domain is divided into clusters using a binary tree. We approximate the dense submatrices of the hierarchical matrices as low-rank matrices using adaptive cross approximate (ACA). We solve some numerical examples and confirm the reduction of CPU time and required memory for 2-D TD-BIEM with H-matrix and ACA.

Key Words: TD-BIEM, wave equation, fast method, H-matrix, ACA

1. 序論

2次元波動問題を時間域境界積分方程式法で解くことを考える。時間、空間を離散化し代数方程式を構成するが、特定の時間ステップにおいて、ある境界要素からの影響が観測点に到達していない場合、対応する影響係数行列の成分はゼロとなる⁽¹⁾。これは、波動方程式の基本解の形から明らかである。そこで、時間域の境界積分方程式法では、使用メモリ量を削減するために影響係数行列の非ゼロ成分のみを記憶することが多い。しかし、影響係数の非ゼロ成分のみを記憶するためには、各時間ステップで、全ての成分について非ゼロであるかどうかについての判定を行う必要がある。さらに、2次元波動問題では、時間ステップが進む程、境界要素からの影響が観測点に到達するため、影響係数の非ゼロ成分の割合が大きくなる。

影響係数行列の記憶に必要なメモリ量やそれに伴う計算コストを削減する手法として、非ゼロ成分を圧縮する手法が提案されている。本論文では、影響係数行列の非ゼロ成分の判定にかかる計算時間を削減するために、Hackbusch⁽²⁾らによる \mathcal{H} -matrix (\mathcal{H} 行列) の考えを導入する。 \mathcal{H} -matrix を用いた非ゼロ成分の判定では、影響係数の各成分ごとに判定を行

うのではなく、距離が近い境界要素を一纏めにしたクラスタを作成し、クラスタ間の影響を考える。つまり、ある2つのクラスタの組に対応する影響係数行列の部分行列がゼロ行列か否かを判定する⁽³⁾。

さらに、Adaptive Cross Approximation (ACA) を用いて、非ゼロの部分行列を低ランク行列で近似する。ACA を用いることで、低ランク行列を高速に求めることが可能になり、また、圧縮された低ランク行列により行列ベクトル積の演算を高速化することができる⁽⁴⁾。しかし、ACA は影響係数行列全体に対して用いることはできないため、非ゼロ成分の判定の場合と同様に \mathcal{H} -matrix を利用する。具体的には、各時間ステップ毎に、クラスタの大きさとクラスタ間の距離を計算し、クラスタの大きさに対して時刻と距離の差が十分に大きいか否かの判定を行うことで、クラスタに対応する部分行列に ACA による低ランク近似を適用できるかどうかを判定する。

なお、3次元波動問題では基本解の形から明らかのように、一定の時間が経過すれば境界要素からの影響は観測点を通り過ぎる。このため、影響係数行列は時間ステップが大きくなっても疎行列なままであり ACA による行列圧縮の効果が見込めない。本提案手法は2次元波動問題にのみ有効と考えられる。また、時間域の基本解を Laplace 変換域の解で構成し時間域の境界積分方程式を解く手法 (Convolution Quadrature

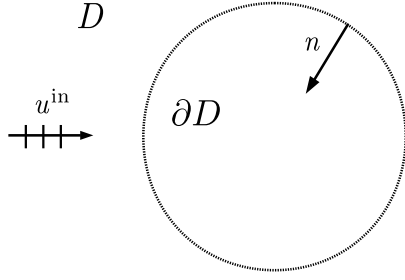


Fig. 1 The wave scattering problem in 2D.

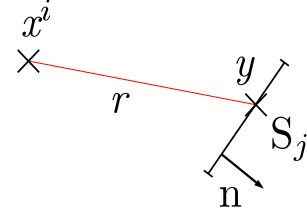


Fig. 2 The distance r between \mathbf{x}^i and S_j .

Method) に対して、影響係数行列に ACA を施し圧縮を試みた研究は既に存在するが⁽⁵⁾、時間域の基本解の波動性を考慮して ACA による高速化を試みた研究は筆者の知る限り存在しない。

2. 時間域境界積分方程式法による波動場解析とその特徴

2次元無限領域に存在する散乱体による次のスカラー波 u^{in} の波動散乱問題を考える (Fig. 1).

$$\ddot{u}(\mathbf{x}, t) - c^2 u_{,ii}(\mathbf{x}, t) = 0 \quad \text{in } D, \quad t > 0, \quad (1)$$

$$u(\mathbf{x}, 0) = \dot{u}(\mathbf{x}, 0) = 0 \quad \text{in } D, \quad (2)$$

$$\frac{\partial u}{\partial n}(\mathbf{x}, t) = \bar{q}(\mathbf{x}, t) \quad \text{on } \partial D, \quad t > 0, \quad (3)$$

$$u(\mathbf{x}, t) \rightarrow u^{\text{in}}(\mathbf{x}, t), \quad |\mathbf{x}| \rightarrow \infty, \quad (4)$$

ここに、 $(\cdot)_{,i}$ は $\frac{\partial}{\partial x_i}$ を、 $(\dot{\cdot})$ は $\frac{\partial}{\partial t}$ を表す。また、 ∂D は領域 D の境界である。 $\frac{\partial}{\partial n}$ は法線微分で $n_i(\mathbf{x}) \frac{\partial}{\partial x_i}$ 、 $\mathbf{n}(\mathbf{x})$ は境界上の点 \mathbf{x} における領域の外向き法線ベクトル、 c は波速を表す。2次元波動方程式の時間域の基本解

$$\Gamma(\mathbf{x}, t) = \frac{H(ct - |\mathbf{x}|)}{2\pi\sqrt{t^2 - |\mathbf{x}|^2/c^2}}, \quad (5)$$

を用いて、式 (1) に対応する境界積分方程式は次式で得られる。

$$\begin{aligned} -u^{\text{in}}(\mathbf{x}, t) &= \int_{\partial D} \left[\Gamma(\mathbf{x} - \mathbf{y}, \cdot) * \frac{\partial \mathbf{u}}{\partial n_{\mathbf{y}}}(\mathbf{y}, \cdot) \right] (t) dS \\ &\quad - \lim_{\boldsymbol{\xi} \rightarrow \mathbf{x}} \int_{\partial D} \left[\frac{\partial \Gamma}{\partial n_{\mathbf{y}}}(\boldsymbol{\xi} - \mathbf{y}, \cdot) * \mathbf{u}(\mathbf{y}, \cdot) \right] (t) dS, \\ &\quad \mathbf{x} \text{ on } \partial D, \quad \boldsymbol{\xi} \in D^e, \quad (6) \end{aligned}$$

$$[f * g](t) = \int_0^\infty f(t-s)g(s)ds, \quad (7)$$

ここに、 D^e は領域 D の補領域であり、 H は Heaviside 関数である。また、 $\frac{\partial}{\partial n_{\mathbf{y}}}$ は法線微分で $n_i(\mathbf{y}) \frac{\partial}{\partial y_i}$ である。

境界積分方程式 (6) を数値的に解くために、境界 ∂D を境界要素 $S_j, j = 1, \dots, N$ に分割し、さらに境界量 $u(\mathbf{x}, t), \frac{\partial u}{\partial n}(\mathbf{x}, t)$ を空間内挿関数 $M_S^j(\mathbf{x})$ 、時間内挿関数 $M_T^m(t)$ を用いて離散化する。

$$u(\mathbf{x}, t) \simeq \sum_{m=1}^{N_T} \sum_{j=1}^N u(\mathbf{x}^j, m\Delta t) M_S^j(\mathbf{x}) M_T^m(t), \quad (8)$$

$$\bar{q}(\mathbf{x}, t) \simeq \sum_{m=1}^{N_T} \sum_{j=1}^N \bar{q}(\mathbf{x}^j, m\Delta t) M_S^j(\mathbf{x}) M_T^m(t), \quad (9)$$

ここに、点 \mathbf{x}^j は境界要素 S_j の代表点であり、 Δt は時間増分で、 N_T は時間ステップ数である。時刻 $t = n\Delta t, n = 1, \dots, N_T$ における離散化された境界積分方程式は次式で得られる。

$$-\mathbf{u}_n^{\text{in}} \simeq \sum_{m=1}^n \mathbf{U}_{n-m+1} \bar{\mathbf{q}}_m - \sum_{m=1}^n \mathbf{W}_{n-m+1} \mathbf{u}_m, \quad (10)$$

$$\{\mathbf{u}_n^{\text{in}}\}_i := u^{\text{in}}(\mathbf{x}^i, n\Delta t), \quad (11)$$

$$\{\mathbf{u}_m\}_i := u(\mathbf{x}^i, m\Delta t), \quad (12)$$

$$\{\bar{\mathbf{q}}_m\}_i := \bar{q}(\mathbf{x}^i, m\Delta t), \quad (13)$$

$$\begin{aligned} &\{\mathbf{U}_{n-m+1}\}_{ij} \\ &:= \int_{\partial D} \int_0^\infty \Gamma(\mathbf{x}^i - \mathbf{y}, n\Delta t - s) M_S^j(\mathbf{y}) M_T^m(s) ds dS, \quad (14) \end{aligned}$$

$$\begin{aligned} &\{\mathbf{W}_{n-m+1}\}_{ij} \\ &:= \int_{\partial D} \int_0^\infty \frac{\partial \Gamma}{\partial n_{\mathbf{y}}}(\mathbf{x}^i - \mathbf{y}, n\Delta t - s) M_S^j(\mathbf{y}) M_T^m(s) ds dS. \quad (15) \end{aligned}$$

簡単のため、区分一定の空間内挿関数 $M_S^j(\mathbf{x})$ 、区分線形の時間内挿関数 $M_T^m(t)$ を考える。

$$M_S^j(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in S_j, \\ 0, & \text{otherwise,} \end{cases} \quad (16)$$

$$M_T^m(t) = \begin{cases} \frac{t}{\Delta t} - m + 1, & (m-1)\Delta t \leq t < m\Delta t, \\ \frac{t}{\Delta t} + m + 1, & m\Delta t \leq t < (m+1)\Delta t, \\ 0, & \text{otherwise.} \end{cases} \quad (17)$$

このとき、影響係数は次式となる。

$$\{\mathbf{U}_m\}_{ij} = \int_{S_j} \int_{(m-2)\Delta t}^{m\Delta t} \frac{H(c\tau - r)}{2\pi\sqrt{\tau^2 - r^2/c^2}} M_T^{m-1}(\tau) d\tau dS, \quad (18)$$

$$\begin{aligned} &\{\mathbf{W}_m\}_{ij} \\ &= \int_{S_j} \int_{(m-2)\Delta t}^{m\Delta t} \frac{\partial}{\partial n_{\mathbf{y}}} \left\{ \frac{H(c\tau - r)}{2\pi\sqrt{\tau^2 - r^2/c^2}} \right\} M_T^{m-1}(\tau) d\tau dS, \quad (19) \end{aligned}$$

ここに、 $r = |\mathbf{x}^i - \mathbf{y}|$ である。

式 (18),(19) 右辺の被積分関数に Heaviside 関数が含まれ

ていることから明らかなように,

$$\{U_m\}_{ij} \neq 0, \{W_m\}_{ij} \neq 0 \quad \text{for} \quad \left\{i, j \mid \inf_{\mathbf{y} \in S_j} r < cm\Delta t\right\}, \quad (20)$$

$$\{U_m\}_{ij} = 0, \{W_m\}_{ij} = 0 \quad \text{for} \quad \left\{i, j \mid \inf_{\mathbf{y} \in S_j} r \geq cm\Delta t\right\}, \quad (21)$$

となる.

2次元波動方程式を従来の時間域境界積分方程式法で解くとき, 非ゼロの係数の積分計算に必要な計算コストは, 係数行列式(18),(19)より, $\mathcal{O}(N^2 N_T)$ となる. また, 非ゼロ成分の行列ベクトル積演算にかかる計算コストは, 式(10)より, $\mathcal{O}(N^2 N_T^2)$ となる. また, 係数行列 U_m, W_m のどの成分が非ゼロであるかを判定するコストは $\mathcal{O}(N^2 N_T)$ となる. よって, 2次元波動方程式の時間域境界積分方程式法に必要な計算コストは, 行列ベクトル積演算にかかる計算コストが支配的になり $\mathcal{O}(N^2 N_T^2)$ となる. 一方, メモリ使用量については, 式(3)のような Neumann 境界条件の場合, 記憶すべき係数行列は $W_m, m = 1, \dots, N_T$ のみである. また, 逐次的に解かれる代数方程式(10)において, 実行可能な行列ベクトル積演算を, 予め計算しておく cast forward のアイデア⁽⁶⁾⁽⁷⁾ を用いれば, $W_m, m = 1, \dots, \lfloor \frac{N_T+1}{2} \rfloor$ を記憶するだけでよい. ここに, $\lfloor \cdot \rfloor$ はガウス記号である. 非ゼロの $W_m, m = 1, \dots, \lfloor \frac{N_T+1}{2} \rfloor$ の記憶に必要なメモリ容量は $\mathcal{O}(N^2 N_T)$ である. そこで, 次節で ACA と \mathcal{H} -matrix を用いて計算コストやメモリ使用量を削減する手法について述べる.

3. \mathcal{H} -matrix 及び ACA を用いた時間域境界積分方程式法による解法

影響係数行列の非ゼロ成分の判定にかかる計算コストを減らすために, また, 行列の低ランク近似のために \mathcal{H} -matrix を導入する. 具体的には, 距離の近い境界要素同士をある程度纏めたクラスタを作成し, クラスタ単位で対応する部分行列がゼロ行列か否かを判別する. さらに, 非ゼロ行列の低ランク近似を, ACA を用いて行う. \mathcal{H} -matrix は下記の手順で作成する.

3.1. クラスタリングと \mathcal{H} -matrix

まず, 境界全体を囲むような長方形を作成し, レベル0のクラスタ Q_1^0 とする. クラスタ Q_1^0 の最も長い辺を2等分した線分で境界を分割し, 分割した境界を囲む新たなレベル1の二つのクラスタ, Q_1^1, Q_2^1 を作成する. 以下同様に, それぞれのクラスタに含まれる境界を分割していき, クラスタ $Q_i^{\text{lev}}, (i = 1, \dots, 2^{\text{lev}})$ を作成する. ここに, lev はクラスタのレベルを表す. また, クラスタのレベルが予め設定した値 maxlev に至るか, クラスタに含まれる境界要素数が leafmax 以下になった時点で分割を停止するように設定する. そして, 分割を停止したクラスタを leaf と呼ぶことにする. 参考のため, クラスタリングの様子を Fig.3 に示す.

次に, 境界要素番号をクラスタごとに纏まるように付け直し, 影響係数行列 U_m, W_m を新しい要素番号に対応するように並び替えることで, Fig.4 に示すような階層行列 U_m^H, W_m^H を作る.

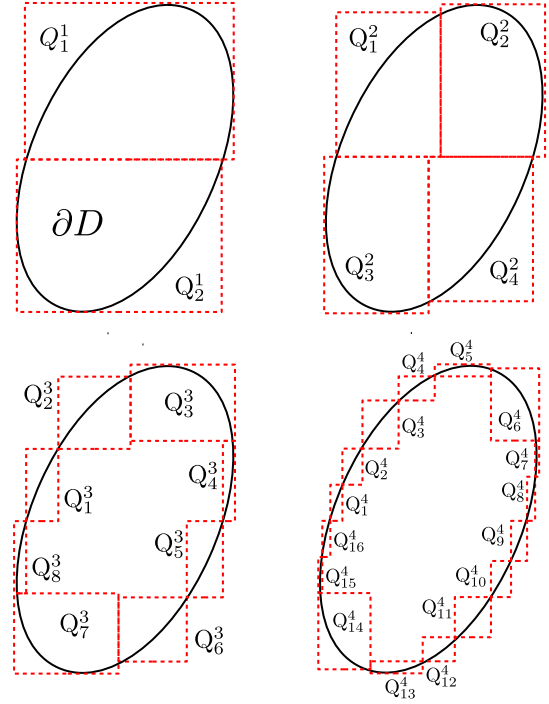


Fig. 3 Clustering.

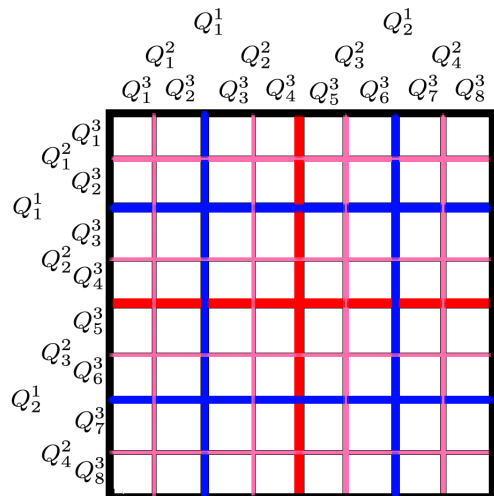


Fig. 4 \mathcal{H} -matrix: W_m^H .

3.2. \mathcal{H} -matrix と低ランク近似

時間ステップが進むとともに、階層化された影響係数行列 U_m^H, W_m^H の非ゼロ成分の割合が大きくなっていくため、非ゼロ成分の判定を高速化する効果が小さくなる。そこで、各成分を求める際に必要となる積分計算や行列ベクトル積の演算を高速化するために、ACA による低ランク近似を適用する。行列を低ランク近似するには行列の各成分がある程度近い値を持っていないと行けないので、影響係数行列全体に対して ACA による低ランク近似を行うことはできない。そこで、観測点 x^i が含まれるクラスタと境界要素 S_j が含まれるクラスタの組み合わせに対応する部分行列に対して ACA を適用できるかどうかを判断する。式 (18), (19) より明らかなように、影響係数行列の各成分は、時刻 $m\Delta t$ と境界要素間の距離 r との差が大きいほど小さくなり、その差と比較して十分に小さいサイズのクラスタであれば、対応する部分行列では、行列成分はおおむね同じような値を持つことになる。よって、ある二つのクラスタ間において、クラスタの大きさに対して時刻とクラスタ間の距離との差が大きいとき、そのクラスタに対応する階層行列 U_m^H, W_m^H の部分行列に対して ACA を適用することにする。つまり、時間が経過するほど、より多くのクラスタに対応する部分行列に対して ACA を適用することが可能になる。

3.3. Adaptive Cross Approximation

ACA は低ランク行列を高速に求める手法であり、行列の成分の記憶に必要なメモリ量を軽減できる。ある行列 $M \in \mathbb{R}^{n \times m}$ に部分ピボットによる ACA を適用することを考える。

$$(v_r)_l := (\gamma)^{-1}(M - S_{r-1})_{il}, \quad (l = 1, \dots, n), \quad (22)$$

$$(u_r)_l := (M - S_{r-1})_{lj}, \quad (l = 1, \dots, m), \quad (23)$$

$$S_r := S_{r-1} + (u_r)(v_r^T), \quad (24)$$

ここに、 i, j は選択されたピボット行と列であり、 S_r は ACA による近似で得られる行列である。また、 $S_0 = 0$ とする。 $r = 1$ のときには $i = 1$ とし、 γ として $M - S_{r-1}$ の i 行成分のうち絶対値最大のものを選び、 j をその列番号とする。 S_r が条件、

$$\|u_r\|_F \|v_r\|_F < \varepsilon \|S_r\|_F, \quad (25)$$

を満たすなら S_r を行列 M の低ランク近似とし、近似のランクを $k = r$ として反復を終了する。満たさないときには、 u_r の成分で絶対値最大となっている行を新たに i とする。そして、 $r = r + 1$ として、さらなる近似を行い、 r が一定値 maxrank になるまで反復を続ける。ここに、 $\|\cdot\|_F$ はフロベニウスノルムであり、 $0 < \varepsilon < 1$ は近似の許容パラメータである。これにより行列 M を、

$$M \simeq S_r = \sum_{r=1}^k u_r v_r^T,$$

のように近似することができる。ACA を用いることにより、メモリ使用量を $O(k(n+m))$ 、行列の積分計算や、行列ベクトル積の計算量を $O(k(n+m))$ にすることができる。

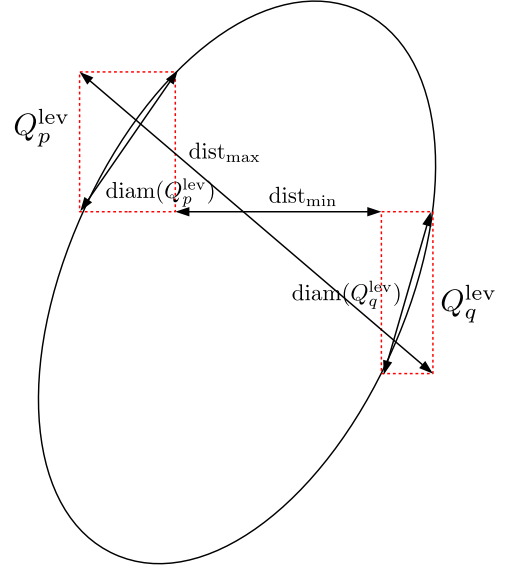


Fig. 5 The size and distance of clusters.

3.4. \mathcal{H} -matrix と ACA を用いた 2 次元波動方程式の解法

影響係数行列の非ゼロ判定を階層行列 U_m^H, W_m^H に対してクラスタ毎に纏めて行い、同時に、クラスタに対応する部分行列に対して、ACA による低ランク近似を行うことができるか否かの判定を行う。二つのクラスタ $Q_p^{\text{lev}}, Q_q^{\text{lev}}$ 間の最短距離を $\text{dist}_{\min}(Q_p^{\text{lev}}, Q_q^{\text{lev}})$ 、 $Q_p^{\text{lev}}, Q_q^{\text{lev}}$ 間の最長距離を $\text{dist}_{\max}(Q_p^{\text{lev}}, Q_q^{\text{lev}})$ とする。また、各クラスタの大きさを対角線の長さとし、 $\text{diam}(Q_p^{\text{lev}})$ 、 $\text{diam}(Q_q^{\text{lev}})$ と表す (Fig.5)。時間域境界積分方程式法の逐次計算の各時間ステップ $m = 1, \dots, N_T$ において、

- クラスタ間の最短距離が条件

$$\text{dist}_{\min}(Q_p^{\text{lev}}, Q_q^{\text{lev}}) > cm\Delta t, \quad (26)$$

を満たす場合、クラスタに対応する部分行列はゼロ行列と判定できる。

$$U_m^H(Q_p^{\text{lev}}, Q_q^{\text{lev}}) = W_m^H(Q_p^{\text{lev}}, Q_q^{\text{lev}}) = 0. \quad (27)$$

また、観測点側のクラスタと積分点側のクラスタを入れ替えても同様のことが言える。

$$U_m^H(Q_q^{\text{lev}}, Q_p^{\text{lev}}) = W_m^H(Q_q^{\text{lev}}, Q_p^{\text{lev}}) = 0. \quad (28)$$

- 条件 (26) を満たさず、かつクラスタの大きさが条件

$$\min(\text{diam}(Q_p^{\text{lev}}), \text{diam}(Q_q^{\text{lev}})) < \eta \left\{ c(m-2)\Delta t - \text{dist}_{\max}(Q_p^{\text{lev}}, Q_q^{\text{lev}}) \right\}, \quad (29)$$

を満たす場合、クラスタに対応する部分行列に対して ACA による低ランク近似を行う。ここで、 $0 < \eta < 1$ は許容パラメータである。 η を大きくするほど、多くのクラスタに対して近似を行うことができるので圧縮率が向上するが、近似を行った際の精度が低下する。

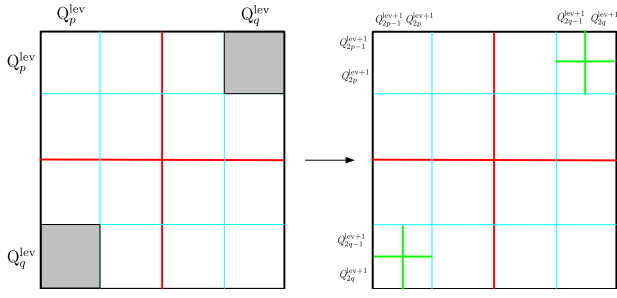


Fig. 6 The division of the sub-matrix.

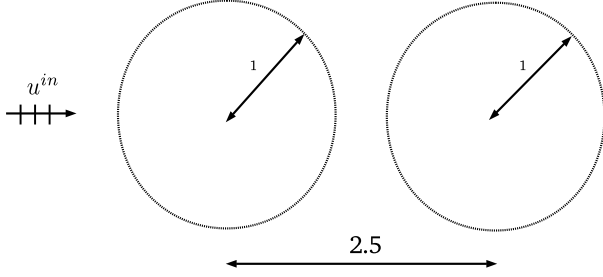


Fig. 7 The wave scattering problem with two scatterers.

- クラスタが条件 (26), (29) を満たさず、かつ leaf でもない場合は、それぞれのクラスタを二つに分割する (Fig.6).

$$Q_p^{\text{lev}} \rightarrow Q_{2p-1}^{\text{lev}+1}, Q_{2p}^{\text{lev}+1}, \quad Q_q^{\text{lev}} \rightarrow Q_{2q-1}^{\text{lev}+1}, Q_{2q}^{\text{lev}+1}. \quad (30)$$

- クラスタ $Q_p^{\text{lev}}, Q_q^{\text{lev}}$ が leaf ならば、対応する部分行列には ACA を適用せずに、各成分の非ゼロ判定は式 (20) を用いて行う。

3.5. ACA により圧縮された \mathcal{H} -matrix の行列ベクトル積演算

従来法では行列ベクトル積の計算コストは $\mathcal{O}(N^2 N_T^2)$ であったが、ACA を用いた手法では、時間ステップが進むほど ACA を適用できる範囲が大きくなっていく、つまり、各時間ステップでの非ゼロ成分を $\mathcal{O}(N \log N)$ に圧縮できるので、行列ベクトル積の計算コストを $\mathcal{O}(N \log N N_T^2)$ に減らすことができる。

4. 数値解析例

2次元無限領域に半径1の球状の散乱体が2つ存在する領域 D における式 (1)-(4) の初期値境界値問題を考える。Fig.7 のように、散乱体は中心間距離を 2.5 とし x_1 方向に一列に並んでいる。また、入射波や境界条件は以下のとおりとする。

$$u^{\text{in}}(\mathbf{x}, t) = \left\{ 1 - \cos 2\pi \left(t - \frac{\mathbf{x} \cdot \mathbf{d}}{c} \right) \right\} H \left(t - \frac{\mathbf{x} \cdot \mathbf{d}}{c} \right), \quad (31)$$

$$\bar{q}(\mathbf{x}, t) = \frac{\partial u^{\text{in}}}{\partial n}(\mathbf{x}, t), \quad (32)$$

ここに、 $c = 1, \mathbf{d} = (1, 0)^T$ とする。なお、この問題の解析解は入射波 $u^{\text{in}}(\mathbf{x}, t)$ となる。今回は入射波が左端の散乱体

達してから右端の散乱体を抜ける、 $t = 4.5$ までの計算を行った。このとき、 $N_T = \mathcal{O}(N)$ となる。この問題を、従来の時間域境界積分方程式法と \mathcal{H} -matrix と ACA を用いた時間域境界積分方程式法の2通りの手法で数値解析を行った。境界要素分割数 N を変えて両手法で解析したメモリ使用量と計算時間の比較を Table 1, Fig.8,9 に示す。表中の"err"は解析解との相対誤差

$$\text{err} := \frac{\left(\sum_{n=1}^{N_T} \sum_{i=1}^N (u(\mathbf{x}^i, n\Delta t) - u^{\text{in}}(\mathbf{x}^i, n\Delta t))^2 \right)^{\frac{1}{2}}}{\left(\sum_{n=1}^{N_T} \sum_{i=1}^N u^{\text{in}}(\mathbf{x}^i, n\Delta t)^2 \right)^{\frac{1}{2}}} \quad (33)$$

を示す。なお、全ての問題において \mathcal{H} -matrix と ACA を用いた時間域境界積分方程式法では、 $\eta = 0.5, \varepsilon = 1.0 \times 10^{-3}, \text{maxlev} = 12, \text{leafmax} = 40, \text{maxrank} = 10$ とした。これらの値は、ACA による圧縮が境界要素法の離散化誤差に影響しない程度とした。

Table 1 The comparison of the required memory and CPU time.

N	Δt	N_T	Conv.		
			mem (MB)	time (sec)	err
500	0.025	179	179	25.2	1.4×10^{-3}
1000	0.013	358	1432	322.7	3.8×10^{-4}
1500	0.0084	537	4833	1699	1.6×10^{-4}
2000	0.0062	716	11456	5805	9.7×10^{-5}
3000	0.0042	1074	38664	33061	4.3×10^{-5}

N	Δt	N_T	\mathcal{H} -mat and ACA		
			mem (MB)	time (sec)	err
500	0.025	179	72.0	27.4	1.4×10^{-3}
1000	0.013	358	406.9	188.2	3.8×10^{-4}
1500	0.0084	537	1130	625.5	1.6×10^{-4}
2000	0.0062	716	2089	1359	9.7×10^{-5}
3000	0.0042	1074	5330	4388	4.3×10^{-5}
4000	0.0031	1432	10042	11303	2.3×10^{-5}
6000	0.0021	2148	25150	38377	1.0×10^{-5}

Fig.8,9 から明らかなように、従来の時間域境界積分方程式法では2次元波動方程式を解くために $\mathcal{O}(N^4)$ の計算時間と、 $\mathcal{O}(N^3)$ の記憶容量が必要だったが、 \mathcal{H} -matrix と ACA による低ランク近似を用いた手法では、計算時間を $\mathcal{O}(N^3 \log N)$ 、記憶容量を $\mathcal{O}(N^2 \log N)$ まで減らすことができた。

参考文献

- (1) 小林昭一 他: 波動解析と境界要素法, 京都大学学術出版会, 2000.
- (2) W. Hackbusch: A sparse matrix arithmetic based on \mathcal{H} -matrices. Part I: Introduction to \mathcal{H} -Matrices, Computing., Vol. 62, pp.89-108, 1999.
- (3) 吉川仁: 3次元波動方程式の \mathcal{H} -matrix を用いた時間域 BIEM による解法, 計算数理工学論文集, Vol. 14, pp.91-94, 2014.
- (4) S. Rjasanow and O. Steinbach: The fast solution of boundary integral equations, pp.126-129, 2007.
- (5) 吉川仁, 松浦亮介, 西村直志: Lubich の CQM を用いた時間域境界積分方程式法の ACA による高速化について, 土木学会論文集 A2 分冊 (応用力学) 特集号, Vol. 69(2), pp.187-193, 2013.
- (6) S. Walker: Scattering analysis via time-domain integral equations: Methods to reduce the scaling of cost with frequency, *IEEE Ant. Prop. Mag.*, Vol. 39, pp.1320, 1997.
- (7) H. Yoshikawa and N. Nishimura: An improved implementation of time domain elastodynamic BIEM in 3D for large scale problems and its application to ultrasonic NDE, *Electronic Journal of Boundary Elements*, Vol. 1, Issue 2, pp.201217, 2003.

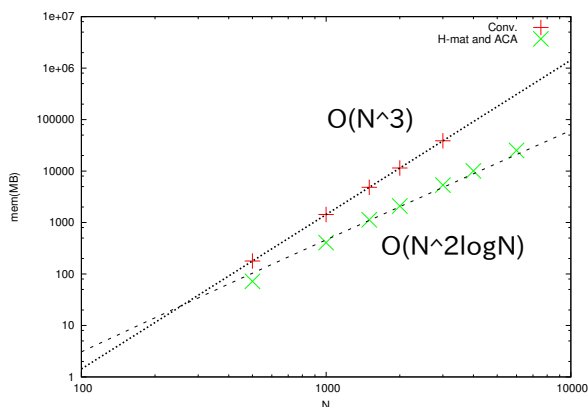


Fig. 8 The comparison of the required memory.

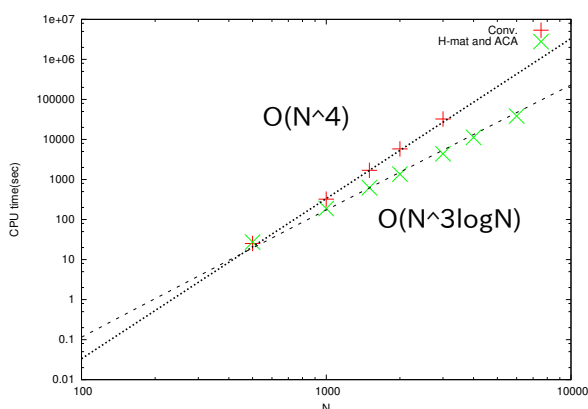


Fig. 9 The comparison of CPU time.

5. 結論

2次元波動方程式を従来の時間域境界積分方程式法で解く場合, 影響係数行列の非ゼロ成分の割合は時間ステップが進むとともに大きくなっていく. 非ゼロ成分の記憶に必要な容量は $\mathcal{O}(N^2 N_T)$ であり, さらに, 影響係数の積分計算と非ゼロ成分の判定に必要な計算量は $\mathcal{O}(N^2 N_T)$ であるが, 各時間ステップでの行列ベクトル積の計算に $\mathcal{O}(N^2 N_T^2)$ の計算量が必要であった. しかし, \mathcal{H} -matrix と ACA を用いることにより, 影響係数行列の非ゼロ成分を圧縮して計算を行うことができ, 非ゼロ成分の判定に必要な計算量が $\mathcal{O}(N(\log N) N_T)$, 行列ベクトル積にかかる計算量が $\mathcal{O}(N(\log N) N_T^2)$ となり, 求解にかかる時間を $\mathcal{O}(N(\log N) N_T^2)$, メモリ使用量を $\mathcal{O}(N(\log N) N_T)$ に軽減できた.