

Time Domain BIEM with CQM Accelerated with ACA and Truncation for the Wave Equation

H. Yoshikawa¹, R. Matsuura², and N. Nishimura¹

Abstract: The convolution integrals with respect to time in the time domain boundary integral equation method (TD-BIEM) are calculated approximately using the Lubich convolution quadrature method (CQM). The influence matrices in the discretized boundary integral equation are computed with the Laplace transform of the fundamental solution in TD-BIEM with the Lubich CQM. These matrices, however, are dense, and both the computational cost and memory requirements are high. In this paper, we apply Adaptive Cross Approximation (ACA) to the influence matrices to achieve a fast solver of TD-BIEM with the Lubich CQM. Moreover, we reduce the computational time of TD-BIEM with the Lubich CQM for hyperbolic PDE problems considering the arrival time of the influence from the source element to the observation point and using cast forward idea. The effect of the proposed method is confirmed with some numerical results.

Keywords: time domain BIEM, CQM, ACA, truncation, cast forward

1 Introduction

Time domain boundary integral equation methods(TD-BIEM) are efficient in solving wave scattering problems because BIEM can easily treat exterior problems and radiation conditions in particular[Friedman and Shaw (1962)],[Mansur and Brebbia (1982)], [Tsinopoulos et al. (2012)],[Wei et al. (2012)]. In TD-BIEM, one discretizes the boundary values with the spatial and temporal interpolation functions, and derives algebraic equations at each time step. The RHS of the algebraic equation at certain time step consists of the influence from the past solutions. The influence matrices in the algebraic equations are sparse in problems governed by hyperbolic PDEs because the components are zero before the arrival time of the influence of the past solutions.

¹ Kyoto University, Kyoto, Japan.

² NTT R& D, Japan.

In the last few years, several authors have solved time-dependent PDE problems with TD-BIEM with the Lubich convolution quadrature method (CQM) [Schanz (1997)], [Monegato et al. (2010)]. This is mainly in order to resolve the problem of the stability of TD-BIEM for hyperbolic PDEs, which has been a long standing issue for the BEM community. The Lubich CQM provides a stable time domain method of numerical computation of the convolution integrals using the Laplace transform of the kernel of the integral equation [Lubich (1988a)], [Lubich (1988b)], [Lubich (2004)]. In TD-BIEM with the Lubich CQM, one computes the convolution integrals with respect to time which appear in RHS of the boundary integral equation using the Laplace transform of the fundamental solution. Carrying out Laplace transforms with respect to time, one removes the time dependence of the fundamental solution. In contrast to the ordinary TD-BIEM which deals with sparse matrices, the Lubich CQM computes the influence matrices in TD-BIEM using dense matrices (i.e. the Laplace transform of the fundamental solution) even for hyperbolic PDEs. By this reason the computational time and the memory requirements for TD-BIEM with Lubich CQM are quite large.

Several approaches have been developed in order to reduce the computational loads of TD-BIEM with the Lubich CQM. Banjai and Sauter [Banjai and Sauter (2008)] proposed a reformulated CQM. Also, Messner and Schanz applied Adaptive Cross Approximation (ACA) to TD-BIEM with the reformulated Lubich CQM for elastodynamics [Messner and Schanz (2010)]. In the reformulated Lubich CQM, they calculate the Laplace transform of the boundary values and transform the time stepping procedure to another of superposing the solution of Laplace domain solutions, thus reducing the Lubich CQM to a variant of the classical Laplace-domain approaches which date back at least to 1960s [Cruse and Rizzo (1968)]. One may therefore say that the reformulated CQM achieved the memory reduction at the cost of losing the time domain nature of the CQM.

In this paper we stick to the original version of TD-BIEM with the Lubich CQM and consider techniques for memory reduction and acceleration of its time stepping procedure. We apply ACA and a truncation to the influence matrices of the algebraic equations in time stepping procedure in the wave equation in 3D. We present several numerical results to confirm the effectiveness of the proposed method.

2 TD-BIEM with CQM

2.1 CQM

The Lubich CQM [Lubich (1988a)], [Lubich (1988b)] is a method of numerical calculations for convolution integrals. This method computes the convolution integral

$$(f * g)(t) = \int_0^t f(t - \tau)g(\tau)d\tau, \tag{1}$$

as

$$(f * g)(n\Delta t) \simeq \sum_{m=0}^n \omega_{n-m}(\Delta t)g(m\Delta t), \quad n = 1, 2, 3, \dots, \tag{2}$$

$$\omega_n(\Delta t) := \frac{1}{2\pi i} \int_{|z|=R} \widehat{f}\left(\frac{\gamma(z)}{\Delta t}\right) z^{-(n+1)} dz \tag{3}$$

$$\simeq \frac{R^{-n}}{L} \sum_{\ell=0}^{L-1} \widehat{f}\left(\frac{\gamma(R e^{i\frac{2\pi\ell}{L}})}{\Delta t}\right) e^{-in\frac{2\pi\ell}{L}}, \tag{4}$$

$$\widehat{f}(s) = \int_0^\infty f(t)e^{-st} dt, \tag{5}$$

where Δt is the time increment, $i^2 = -1$, R is the radius of the circular integration path and L is the division number of the integration path. Also, $\gamma(z)$ is the quotient of the generating polynomials of a linear multistep method. For the K -step backward differential formula, we have

$$\gamma(z) = \sum_{p=1}^K \frac{1}{p}(1-z)^p, \quad K \leq 6. \tag{6}$$

2.2 TD-BIEM with CQM

We consider the following initial-boundary values problem for the wave equation with the Neumann boundary condition,

$$\ddot{u}(x,t) - u_{,ii}(x,t) = 0 \quad \text{in } D, \quad t > 0, \tag{7}$$

$$u(x,0) = 0, \quad \text{in } D, \tag{8}$$

$$\dot{u}(x,0) = 0 \quad \text{in } D, \tag{9}$$

$$\frac{\partial u}{\partial n_x}(x,t) = q(x,t) \quad \text{on } \partial D, \quad t > 0, \tag{10}$$

where D is the domain, ∂D is the boundary of the domain D , $\frac{\partial}{\partial n_x}$ is the normal derivative, $n(x)$ is the outward unit normal vector on the boundary ∂D , $q(x,t)$ is the given function, and $(\cdot)_{,i}$ and $(\dot{\cdot})$ denote $\frac{\partial}{\partial x_i}$ and $\frac{\partial}{\partial t}$. The corresponding boundary integral equation (BIE) for Eq.(7) is obtained as

$$0 = \int_{\partial D} \left[\Gamma(x-y, \cdot) * \frac{\partial u}{\partial n_y}(y, \cdot) \right](t) dS - \lim_{\xi \rightarrow x} \int_{\partial D} \left[\frac{\partial \Gamma}{\partial n_y}(\xi - y, \cdot) * u(y, \cdot) \right](t) dS, \tag{11}$$

$x \text{ on } \partial D, \xi \in D^e,$

where D^e is the exterior domain of D and $\Gamma(x, t)$ is the fundamental solution for the wave equation in time domain given by

$$\Gamma(x, t) = \frac{\delta(t - |x|)}{4\pi|x|}. \tag{12}$$

Using the Lubich CQM, the convolution integrals in the RHS of the BIE(11) are discretized as

$$\begin{aligned} 0 \simeq & \int_{\partial D} \sum_{m=0}^n \omega_{n-m}^B(x-y, \Delta t) \frac{\partial u}{\partial n_y}(y, m\Delta t) dS \\ & - \lim_{\xi \rightarrow x} \int_{\partial D} \sum_{m=0}^n \frac{\partial \omega_{n-m}^B}{\partial n_y}(\xi - y, \Delta t) u(y, m\Delta t) dS, \end{aligned} \tag{13}$$

where

$$\omega_n^B(x-y, \Delta t) := \frac{R^{-n}}{L} \sum_{\ell=0}^{L-1} \hat{\Gamma}\left(x-y, \frac{\gamma(Re^{i\frac{2\pi\ell}{L}})}{\Delta t}\right) e^{-in\frac{2\pi\ell}{L}}, \tag{14}$$

$$\hat{\Gamma}(x, s) = \frac{e^{-|x|s}}{4\pi|x|}. \tag{15}$$

The boundary ∂D is discretized with the boundary elements $S_j, j = 1, \dots, N$ and the boundary values are also discretized as

$$u(x, t) \simeq \sum_{j=1}^N M_S^j(x) u(x^j, t), \tag{16}$$

$$q(x, t) \simeq \sum_{j=1}^N M_S^j(x) q(x^j, t), \tag{17}$$

where $M_S^j(x)$ is the spatial interpolation functions and the point x^j is the collocation point on the boundary element S_j .

Using Eqs.(16) and (17), we obtain the following algebraic equations at $t = n\Delta t, n =$

$1, \dots, N_T$.

$$0 \simeq \sum_{m=1}^n \Phi_{n-m} q_m - \sum_{m=1}^n \Psi_{n-m} u_m, \tag{18}$$

$$\{u_m\}_i := u(x^i, m\Delta t), \tag{19}$$

$$\{q_m\}_i := q(x^i, m\Delta t), \tag{20}$$

$$\{\Phi_n\}_{ij} \simeq \sum_{\ell=0}^{L-1} C_\ell^n \{U_\ell\}_{ij}, \tag{21}$$

$$\{\Psi_n\}_{ij} \simeq \sum_{\ell=0}^{L-1} C_\ell^n \{W_\ell\}_{ij}, \tag{22}$$

$$C_\ell^n := \frac{R^{-n}}{L} e^{-in\frac{2\pi\ell}{L}}, \tag{23}$$

$$\{U_\ell\}_{ij} := \int_{\partial D} \omega_B(x^i - y, \Delta t) M_S^j(y) dS, \tag{24}$$

$$\{W_\ell\}_{ij} := \lim_{x \in D^e \rightarrow x^i} \int_{\partial D} \frac{\partial \omega_B(x - y, \Delta t)}{\partial n_y} M_S^j(y) dS. \tag{25}$$

In the Lubich CQM, one may choose $L = N_T, R^{N_T} = \sqrt{\varepsilon}$ if one needs the accuracy of $\mathcal{O}(\varepsilon)$ [Lubich (1988b)].

In the conventional TD-BIEM with the Lubich CQM, one derives the algebraic equations in Eq.(18) and solves them step by step.

As shown in Eqs.(21) and (22), the influence matrices Φ_n and Ψ_n are calculated as sums of products of the time-dependent coefficients C_ℓ^n and the time-independent matrices U_ℓ and W_ℓ . The algebraic equations in Eq.(18) can be rewritten as

$$0 \simeq \sum_{m=1}^n \sum_{\ell=0}^{L-1} C_\ell^{n-m} U_\ell q_m - \sum_{m=1}^n \sum_{\ell=0}^{L-1} C_\ell^{n-m} W_\ell u_m. \tag{26}$$

In the Neumann problems, the time-independent matrices W_ℓ are stored in the memory in this study. Because W_ℓ are dense, the memory requirements for W_ℓ are $\mathcal{O}(N^2 N_T)$.

3 Accelerations

3.1 \mathcal{H} -matrix with ACA

3.1.1 \mathcal{H} -matrix

We consider an approximation of the the time-independent dense matrices W_ℓ for the reduction of the memory requirements. $\{W_\ell\}_{ij}$ is calculated as in Eqs.(25),(14)

and (15). Because the Laplace transform of fundamental solution $\widehat{\Gamma}$ is nothing other than the fundamental solution of the Yukawa equation, we can apply the \mathcal{H} -matrix approach with ACA to W_ℓ [Bebendorf (2000)],[Rjasanow and Steinbach (2007)]. The procedure goes as in the following.

1. The boundary is divided into clusters Q_i^{lev} ($i = 1, \dots, 2^{\text{lev}}$, $\text{lev} = 0, \dots, \text{maxlev}$) using a binary tree, where “lev” denotes the hierarchical level of the cluster tree.
2. The boundary elements are renumbered for the ease of assembling them into clusters.
3. If the distance between the clusters Q_x and Q_y fulfills the following admissibility condition

$$\min\{\text{diam}Q_x, \text{diam}Q_y\} \leq \eta \text{dist}(Q_x, Q_y), \quad 0 < \eta < 1, \tag{27}$$

the corresponding submatrices of the hierarchized matrices W^H_ℓ are approximated by low-rank matrices with ACA. Here, we define the diameters of the clusters Q_x and Q_y and the distance between them by

$$\text{diam}Q_x = \max_{i1, i2 \in Q_x} |x^{i1} - x^{i2}|, \tag{28}$$

$$\text{diam}Q_y = \max_{j1, j2 \in Q_y} |y^{j1} - y^{j2}|, \tag{29}$$

$$\text{dist}(Q_x, Q_y) = \min_{i \in Q_x, j \in Q_y} |x^i - y^j|. \tag{30}$$

3.1.2 Adaptive Cross Approximation

Let $A_{\text{adm}} \in \mathbb{C}^{n \times m}$ be a submatrix of the hierarchized matrices W^H_ℓ which fulfills the admissibility condition (Eq.(27)). Using ACA, the submatrix A_{adm} is approximated by a low-rank matrix A_r as

$$A_{\text{adm}} \simeq A_r = PQ^*, \tag{31}$$

$$|A_{\text{adm}} - A_r|_F \leq \varepsilon_{\text{ACA}} |A_{\text{adm}}|_F, \tag{32}$$

where $P \in \mathbb{C}^{n \times r}$, $Q \in \mathbb{C}^{m \times r}$, $(\cdot)^*$ denotes the Hermitian conjugate and $|\cdot|_F$ denotes the Frobenius norm defined by

$$|A_{\text{adm}}|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |\{A_{\text{adm}}\}_{ij}|^2} = \sqrt{\sum_{j=1}^{\min(m,n)} \sigma_j^2}, \tag{33}$$

where σ_j is the singular value of A_{adm} . The detail for the ACA algorithm can be found in [Rjasanow and Steinbach (2007)].

3.2 Cast forward and Truncation

3.2.1 Cast forward

A cast forward method of evaluating the influence from the past is proposed by Walker [Walker (1997)] (see also [Yoshikawa and Nishimura (2003)]). At $t = n\Delta t$, the past solutions u_m , ($m = 1, \dots, n - 1$) have already been computed in the time stepping procedure. In the cast forward method, one computes Ψ_{n-1} and the matrix-vector products $\Psi_{n-1}u_m$ and $\Psi_m u_{n-1}$, ($m = 1, \dots, n - 1, n + m - 1 \leq N_T$) at $t = n\Delta t$ and the results are subtracted from the RHS of the algebraic equation at $t = (n + m - 1)\Delta t$. In the cast forward method, the matrix Ψ_k for $k \geq N_{T_H}$ appear just once and need not be stored, where $N_{T_H} = \lceil \frac{N_T + 1}{2} \rceil$ and $\lceil \cdot \rceil$ denotes the Gauss symbol.

The following example shows the algebraic equations in the TD-BIEM for $N_T = 6$.

$$\begin{aligned} \text{At } t = \Delta t, & \quad \Psi_0 u_1 = b^S_1, \\ \text{at } t = 2\Delta t, & \quad \Psi_0 u_2 = b^S_2 - \Psi_1 u_1, \\ \text{at } t = 3\Delta t, & \quad \Psi_0 u_3 = b^S_3 - \Psi_2 u_1 - \Psi_1 u_2, \\ \text{at } t = 4\Delta t, & \quad \Psi_0 u_4 = b^S_4 - \underline{\Psi_3 u_1} - \underline{\Psi_2 u_2} - \Psi_1 u_3, \\ \text{at } t = 5\Delta t, & \quad \Psi_0 u_5 = b^S_5 - \underline{\Psi_4 u_1} - \underline{\Psi_3 u_2} - \Psi_2 u_3 - \Psi_1 u_4, \\ \text{at } t = 6\Delta t, & \quad \Psi_0 u_6 = b^S_6 - \underline{\Psi_5 u_1} - \underline{\Psi_4 u_2} - \underline{\Psi_3 u_3} - \Psi_2 u_4 - \Psi_1 u_5, \end{aligned}$$

where $b^S_n = \sum_{m=1}^n \Phi_{n-m} q_m$. Because these underlined matrix-vector products can be computed at $t = 4\Delta t$ with the cast forward method, Ψ_3, Ψ_4, Ψ_5 only have to be calculated once with Eq.(22) at $t = 4\Delta t$.

3.2.2 Truncation

Here again, we consider the meaning of the influence coefficient $\{\Psi_{n-m}\}_{ij}$. The influence coefficient $\{\Psi_{n-m}\}_{ij}$ denotes the influence between the source element S_j at $t = m\Delta t$ and the observation point x_i at $t = n\Delta t$. $\{\Psi_{n-m}\}_{ij}$ are actually zero till the arrival time of the influence from the source element S_j at the observation point x_i for hyperbolic PDE problems, although the influence coefficient $\{\Psi_{n-m}\}_{ij}$ may be calculated to be non-zero values from Eq.(22). In this paper using the \mathcal{H} -matrix approach we set

$$\{\Psi^H_n\}_{ij} = 0 \quad \text{for} \quad \{i \in Q_x, j \in Q_y \mid \text{dist}(Q_x, Q_y) > (n + 1)\Delta t\}, \tag{34}$$

instead of using

$$\{\Psi^H_n\}_{ij} = \sum_{\ell=0}^{L-1} C_\ell^n \{W^H_\ell\}_{ij}. \tag{35}$$

From Eq.(34), the number of non-zero $\{\Psi^H_n\}_{ij}$ increases with the increasing index n .

3.2.3 Memory reduction with cast forward and truncation

The influence matrices Ψ^H_n are calculated from the time-independent matrices W^H_ℓ with Eq.(35). As shown above, the influence matrices Ψ^H_n , ($n = N_{T_H}, \dots, N_T - 1$) are calculated only once at $t = (N_{T_H} + 1)\Delta t$ with cast forward. We therefore store only the component of the time-independent matrices $\{W^H_\ell\}_{ij}$ corresponding to non-zero component of $\Psi^H_{N_{T_H}-1}$.

4 Numerical results

The following wave scattering problem by a sphere having the radius 1 shown in Fig.-1 is considered,

$$\begin{cases} \ddot{u}(x,t) - u_{,ii}(x,t) = 0 & \text{in } D, t > 0, \\ u(x,0) = \dot{u}^{\text{in}}(x,0), & \text{in } D, \\ \dot{u}(x,0) = \dot{u}^{\text{in}}(x,0) & \text{in } D, \\ \frac{\partial u}{\partial n}(x,t) = \frac{\partial u^{\text{in}}}{\partial n}(x,t) & \text{on } \partial D, t > 0. \\ u(x,t) - u^{\text{in}}(x,t) & \text{satisfies radiation condition,} \end{cases} \quad (36)$$

where the incident plane wave is given as follows:

$$u^{\text{in}}(x,t) = \{1 - \cos\{\pi(t - x_1)\}\} \{H(t - x_1) - H(t - x_1 - 2)\}. \quad (37)$$

The solution of this problem is $u(x,t) = u^{\text{in}}(x,t)$.

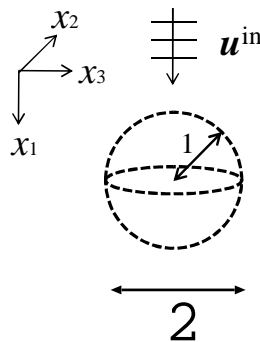


Figure 1: The wave scattering problems by a sphere.

The boundary integral equation for this problem is

$$0 = u^{in}(x, t) + \int_{\partial D} \sum_{m=0}^n \omega_{n-m}^B(x - y, \Delta t) \frac{\partial u}{\partial n_y}(y, m\Delta t) dS - \lim_{\xi \rightarrow x} \int_{\partial D} \sum_{m=0}^n \frac{\partial \omega_{n-m}^B}{\partial n_y}(\xi - y, \Delta t) u(y, m\Delta t) dS, \quad x \text{ on } \partial D, \xi \in D^e. \tag{38}$$

We solve this BIE with

- the conventional TD-BIEM with the Lubich CQM (“conv”),
- TD-BIEM with the Lubich CQM accelerated by ACA (“ACA”) and
- TD-BIEM with the Lubich CQM accelerated by ACA and truncation (“ACA + trunc”).

In each method, we have $N = 980$, $\Delta t = 0.04$ and $N_T = \frac{2}{\Delta t} = 50$. In the \mathcal{H} -matrix approach with ACA, we set $\eta = 0.9$ and $\epsilon_{ACA} = 1.0 \times 10^{-3}$. In this paper we use GMRES as the solver for linear equations and all computation are executed with one CPU(Intel Xeon E5 2.6GHz).

The CPU time, memory requirements and relative error for each of these methods are shown in **Tab.-1**, where the relative error is calculated as

$$(\text{relative error}) = \left(\frac{\sum_{i=1}^N \sum_{n=1}^{N_T} (\{u_n\}_i - u^{in}(x^i, n\Delta t))^2}{\sum_{i=1}^N \sum_{n=1}^{N_T} u^{in}(x^i, n\Delta t)^2} \right)^{\frac{1}{2}}. \tag{39}$$

Table 1: The comparisons in the case of one scatterer. ($N_T = 50$.)

	conv	ACA	ACA+trunc
CPU time(sec)	22.8	38.2	32.4
mem. reduction(%)	100 (381MB)	85.6	57.4
relative error(%)	1.72	1.74	1.74

In this numerical example, ACA is not effective because the boundary, i.e., the surface of the sphere, is not very large and N is small. However, the truncation is effective in memory reduction. This is because the matrices are not compressed much by ACA(**Fig.-2**).

We also solve the scattering problem by three spheres shown in **Fig.-3**. In this problem, we have $N = 3 \times 980 = 2940$, $N_T = \frac{7}{\Delta t} = 175$. The results are shown

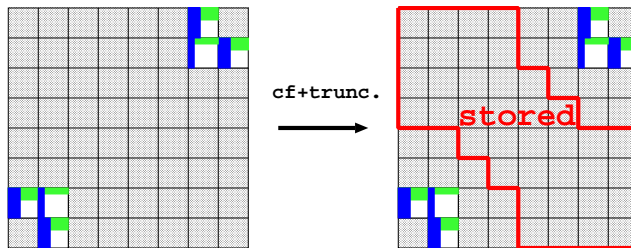


Figure 2: The truncation for the compressed matrix.

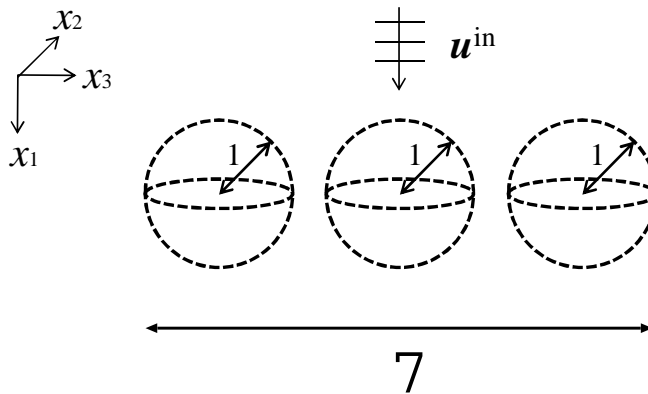


Figure 3: The wave scattering problems by three spheres.

Table 2: The comparisons in the case of three scatterers. (175 time steps.)

	conv	ACA	ACA+trunc
cpu time(sec)	2331	1113	856
mem. reduction(%)	100 (11.3GB)	49.6	49.6
relative error(%)	4.09	4.76	4.76

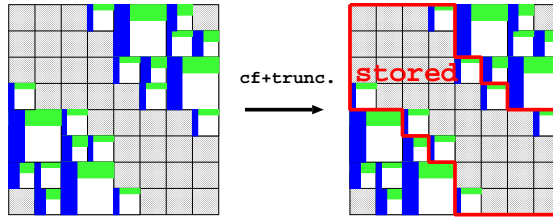


Figure 4: The truncation for the well-compressed matrix.

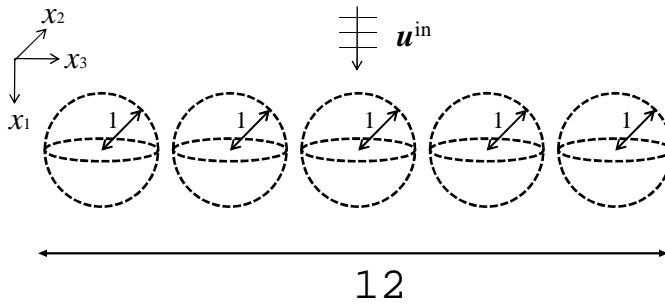


Figure 5: The wave scattering problems by five spheres.

in **Tab.-2**. ACA is effective because the boundary is large in this problem. We have little difference between “ACA” and “ACA+trunc” in the memory reduction, although. “ACA+trunc” is faster than “ACA”. This is because the compressed submatrix by ACA (Eq.(27)) and the truncated submatrix (Eq.(34)) are overlapped considerably in this problem. Indeed, if the matrix has been well-compressed with ACA, the truncation is ineffective in reducing memory (**Fig.-4**). Additionally in “ACA+trunc”, the computational time of matrix-vector product

$$\Psi_n^H u_m = \sum_{\ell=0}^{N_T-1} C_\ell^n W_\ell^H u_m \tag{40}$$

decreases, because the products in the RHS of Eq.(40) are computed only for non-zero components of the truncated Ψ_n^H .

Tab.-3 shows the comparisons of the effectiveness for the case of five scatterers (**Fig.-5**). In this problem, we have $N = 5 \times 980 = 4900$, $N_T = \frac{12}{\Delta t} = 300$. The boundary is large in this problem which means that the reduction methods are more effective as seen in **Tab.-3**.

Table 3: The comparisons in the case of five scatterers. (300 time steps.)

	conv	ACA	ACA+trunc
cpu time(sec)	21906	4792	3936
mem. reduction(%)	100 (54.0GB)	34.5	34.1
relative error(%)	3.62	4.96	5.37

5 Conclusions

In this paper we have applied ACA to the time stepping procedure in TD-BIEM with the Lubich CQM and reduced the CPU time and memory requirements. We have confirmed that ACA is effective for the problems having large boundaries. We have also truncated the dense influence matrices of TD-BIEM with the Lubich CQM for the wave equation considering the arrival time of influences between the source element and the observation point and using cast forward. The truncation is effective in memory reduction for problems where ACA is not effective. With the truncation, we can further reduce the CPU time of TD-BIEM with the Lubich CQM accelerated with ACA for hyperbolic PDE problems. We will try to apply these accelerations to wave attenuation problems in future works.

References

- Banjai L. and Sauter S.** (2008): Rapid Solution of the Wave Equation in Unbounded Domains. *SIAM J. Numer. Anal.*, vol.47(1), pp.227–249.
- Bebendorf M.** (2000): Approximation of Boundary Element Matrices. *Numerische Mathematik*, vol.86(4), pp.565–589.
- Cruse T. A. and Rizzo F. J.** (1968): A Direct Formulation and Numerical Solution of the General Transient Elastodynamic Problem. I. *Journal of Mathematical Analysis and Applications*, vol.22, pp.244–259.
- Friedman M. B. and Shaw R. P.** (1962): Diffraction of Pulses by Cylindrical Obstacles of Arbitrary Cross-section. *J. Appl. Mech.*, vol.29, pp.40–46.
- Hackbusch W.** (1999): A Sparse Matrix Arithmetic Based on H-Matrices. Part I: Introduction to H-Matrices. *Computing.*, vol.62, pp.89–108.
- Lubich C.** (1988a): Convolution quadrature and discretized operational calculus. I. *Numerische Mathematik*, vol.52, no.2, pp.129–145.
- Lubich C.** (1988b): Convolution quadrature and discretized operational calculus. II. *Numerische Mathematik*, vol.52, no.4, pp.413–425.

- Lubich C.** (2004): Convolution quadrature revisited. *BIT Numerical Mathematics*, vol.44, pp.503–514.
- Mansur W. J. and Brebbia C. A.** (1982): Numerical Implementation of the Boundary Element Method for Two Dimensional Transient Scalar Wave Propagation Problems. *Appl. Math. Model.*, vol.6, pp.299–306.
- Messner M. and Schanz M.** (2010): An accelerated symmetric time-domain boundary element formulation for elasticity. *Engineering Analysis with Boundary Elements*, vol.34, issue 10, pp. 944–955.
- Monegato G., Scuderi L., and Stanic M. P.** (2010): Lubich convolution quadratures and their application to problems described by space-time BIEs. *Numerical Algorithms*, vol.56, no.3, pp.405–436.
- Rjasanow S. and Steinbach O.** (2007): *The Fast Solution of Boundary Integral Equations*. Springer.
- Schanz M.** (2001): A New Visco- and elastodynamics Time Domain Boundary Element Formulation. *Comp. Mech.*, vol.20, pp.452–459.
- Tsinopoulos S. V. Polyzos, D., and Beskos, D. E.** (2012): Static and Dynamic BEM Analysis of Strain Gradient Elastic Solids and Structures. *Computer Modeling in Engineering and Sciences*, vol.86, no.2, pp.113–144.
- Walker S. P.** (1997): Scattering analysis via time-domain integral equations: Methods to reduce the scaling of cost with frequency. *IEEE Ant. Prop. Mag.*, vol.39, pp.13–20.
- Wei Y., Wang Q., Wang Y., and Huang Y.** (2012): Optimizations for Elastodynamic Simulation Analysis with FMM-DRBEM and CUDA. *Computer Modeling in Engineering and Sciences*, vol.86, no.3, pp.241–273.
- Yoshikawa H. and Nishimura N.** (2003): 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, no.2, pp.201–217.

