

An improved implementation of time domain elastodynamic BIEM in 3D for large scale problems and its application to ultrasonic NDE

Hitoshi Yoshikawa

Dept. of Civil and Earth Resources Eng., Kyoto Univ.

Naoshi Nishimura

Academic Center for Computing and Media Studies, Kyoto Univ.

Abstract

This paper discusses a three dimensional implementation of boundary integral equation method (BIEM) for large scale time domain elastodynamic problems and its application to ultrasonic nondestructive evaluation (NDE). We improve the time integration algorithm of the BIEM in order to reduce the required computational time. We show the efficiency of the proposed method by applying it to a simple wave scattering problem and to a more realistic crack determination problem related to ultrasonic NDE.

1 Introduction

The present authors[1, 2, 3] have been investigating an inverse problem of determining the position and the shape of unknown cracks in a material using the real waveform data of the ultrasound measured with a laser interferometer. In the numerical analysis of this inverse problem, one solves direct problems for some candidate cracks using the time domain elastodynamic BIEM in 3D. It is readily seen that the size of such problems become quite large if one wishes to determine the crack geometry accurately. Indeed, the dominant wavelength of the incident wave is usually taken comparable to or smaller than the crack size, while one will have to take a few boundary elements per wavelength. If the linear dimension of the analysis domain is taken to be, say, 10 times the size of the crack, one will easily have a large scale problem with several thousands of boundary elements. Moreover, one will also need to choose a small time increment so as to be compatible with the boundary element size. Hence the number of time steps will also be quite large.

Since the publication of the early attempt by Friedman and Shaw [4, 5], however, one seldom sees applications of the time domain BIEMs to large problems. This was also the case in many subsequent works including those for non-mechanical applications. See [6, 7, 8, 9, 10, 11, 12] for example.

One of the reasons that the time domain BIEM has not been applied to large scale problems is that the solution is expressed in the form of the convolution with respect to time. This means that one has to evaluate the influence from all the past in order to obtain the solution at a certain time. Hence the computational

time as well as memory requirements for solving a large scale problem become prohibitive.

In order to fix this drawback, Demirel and Wang[13] proposed a truncation algorithm in two dimensional transient wave propagation problems, where the time integration is truncated and the influence of the far past time is neglected. They showed that the computational time is reduced to one half of the original with this improved algorithm. Walker et al.[14, 15, 16] proposed an approximate BIEM in three dimensional electromagnetics, where the electromagnetic field whose magnitude is below a certain threshold is neglected. They also proposed a ‘project forward’ method of evaluating the influence from the past, where they compute the influence of the current solutions to the future field at the current time. With these methods they estimate that the computational time scales with the fourth to third power of the frequency instead of the conventional fifth power in the wave equation in 3D. However, the size of the problems considered in these investigations are not as large as to disallow the in core storage of all the relevant influence coefficients of the discretized BIE. In many practical problems including ours in NDE, however, the memory of ordinary computer is not enough to store all the influence coefficients.

There is another possibility of dealing with large scale problems in time domain with BIEM. Indeed, the use of fast multipole and related methods in the time domain BIEM for the wave equation is possible, as has been investigated by Michielssen and his group [17]. The elastodynamic counterpart of this approach is also possible as one finds in Takahashi et al.[18] However, these approaches are still in their incipient stage of developments, and the algorithms remain quite complicated.

In view of these, we restrict our attention to the conventional BIEM in this paper and consider large scale problems in three dimensional elastodynamics in time domain where one cannot store all influence coefficients for all the relevant time differences. In order to carry out the analysis within a reasonable amount of computational time, we improve the time integration algorithm in two respects. One is related to the improvement of the algorithm for the convolution with respect to time and the other is concerned with the time interpolation function. In Section 2, we discuss these two improvements in addition to further enhancement of the efficiency of the the improved BIEM with the help of the parallelization (see [19, 20] for related developments). After solving a small test problem, we proceed to the solution of a practical large scale elastodynamic problem related to ultrasonic NDE using the parallel version of the improved BIEM. We show that the proposed approach is promising in inverse problems of determining the geometry of an unknown surface crack using real data.

2 Improved algorithms for three dimensional time domain BIEM for elastodynamics

2.1 Time domain BIE for three dimensional elastodynamics

The scattering of the elastic wave by a traction free crack S in a 3 dimensional elastic region D is considered. The unknown displacement \mathbf{u} is obtained as the solution to the following initial boundary value problem:

$$\begin{aligned} \mu\Delta\mathbf{u} + (\lambda + \mu)\nabla\nabla \cdot \mathbf{u} &= \rho\ddot{\mathbf{u}} \quad \text{in } D \setminus S \times (t > 0), \\ \mathbf{T}\mathbf{u} &= \mathbf{t} \quad \text{on } \partial D \times (t > 0), \\ \mathbf{u}|_{t=0} = \dot{\mathbf{u}}|_{t=0} &= 0 \quad \text{in } D, \\ \mathbf{T}\mathbf{u}^\pm &= 0 \quad \text{on } S \times (t \geq 0), \\ \boldsymbol{\varphi} &= \mathbf{u}^+ - \mathbf{u}^- = 0 \quad \text{on } \partial S, \end{aligned} \tag{1}$$

where λ and μ are Lamé's constants, ρ is the density, \mathbf{T} is the traction operator, the superposed $+(-)$ is the limit of a certain quantity on the crack S from the positive (negative) side, with the positive side indicating the one into which the unit normal vector \mathbf{n} points, $\boldsymbol{\varphi}$ is the crack opening displacement, and $(\dot{})$ indicates the differentiation with respect to time.

The boundary integral equations corresponding to equation (1) are written as

$$\begin{aligned} \frac{1}{2}\mathbf{u}(\mathbf{x}, t) &= \int_{\partial D} \Gamma(\mathbf{x}, \mathbf{y}, t) * \mathbf{T}\mathbf{u}(\mathbf{y}, t) dS - \text{v.p.} \int_{\partial D} \Gamma_I(\mathbf{x}, \mathbf{y}, t) * \mathbf{u}(\mathbf{y}, t) dS \\ &+ \int_S \Gamma_I(\mathbf{x}, \mathbf{y}, t) * \boldsymbol{\varphi}(\mathbf{y}, t) dS \quad \mathbf{x} \in \partial D, \end{aligned} \tag{2}$$

$$\begin{aligned} 0 &= \int_{\partial D} \mathbf{T}\Gamma(\mathbf{x}, \mathbf{y}, t) * \mathbf{T}\mathbf{u}(\mathbf{y}, t) dS - \int_{\partial D} \mathbf{T}\Gamma_I(\mathbf{x}, \mathbf{y}, t) * \mathbf{u}(\mathbf{y}, t) dS \\ &+ \text{p.f.} \int_S \mathbf{T}\Gamma_I(\mathbf{x}, \mathbf{y}, t) * \boldsymbol{\varphi}(\mathbf{y}, t) dS \quad \mathbf{x} \in S, \end{aligned} \tag{3}$$

where $\Gamma(\mathbf{x}, \mathbf{y}, t)$ and $\Gamma_I(\mathbf{x}, \mathbf{y}, t)$ are the fundamental solution and the double layer kernel of three dimensional elastodynamics, '*' indicates the convolution with respect to time, v.p. stands for the Cauchy's principal value of a singular integral, and p.f. denotes the finite part of a divergent integral, respectively.

2.2 Conventional time domain BIEM

The integral equations in (2) and (3) are discretized with the temporal and spatial interpolation functions $M^\ell(\tau)$ and $N^q(\mathbf{y})$, respectively. For the case of the Neumann boundary condition, they reduce to the following algebraic equation:

$$\sum_q A_{ij}^{pq}(\Delta t) \begin{bmatrix} u_j(\mathbf{x}^q, n\Delta t) \\ -\varphi_j(\mathbf{x}^q, n\Delta t) \end{bmatrix} + \begin{bmatrix} \frac{1}{2}u_i(\mathbf{x}^p, n\Delta t) \\ 0 \end{bmatrix}$$

$$= b_i(\mathbf{x}^p, n\Delta t) - \sum_q \sum_{\ell=1}^{n-1} A_{ij}^{pq}((n+1-\ell)\Delta t) \begin{bmatrix} u_j(\mathbf{x}^q, \ell\Delta t) \\ -\varphi_j(\mathbf{x}^q, \ell\Delta t) \end{bmatrix}. \quad (4)$$

In this formula, the matrix A_{ij}^{pq} and vector b_i are defined by

$$\begin{aligned} A_{ij}^{pq}(\ell\Delta t) &= \int_{\partial D+S} \int \Gamma_{Iij}(\mathbf{x}^p, \mathbf{y}, \tau) M^\ell(\tau) N^q(\mathbf{y}) d\tau dS_y, \\ b_i(\mathbf{x}^p, n\Delta t) &= \sum_q \sum_{\ell=1}^n (\mathbf{T}\mathbf{u})_j(\mathbf{x}^q, \ell\Delta t) \\ &\quad \int_{\partial D} \int \Gamma_{ij}(\mathbf{x}^p, \mathbf{y}, \tau) M^{n+1-\ell}(\tau) N^q(\mathbf{y}) d\tau dS_y, \end{aligned}$$

if the collocation point \mathbf{x}^p is on ∂D , and by

$$\begin{aligned} A_{ij}^{pq}(\ell\Delta t) &= \int_{\partial D+S} \int (\mathbf{T}\Gamma_I)_{ij}(\mathbf{x}^p, \mathbf{y}, \tau) M^\ell(\tau) N^q(\mathbf{y}) d\tau dS_y, \\ b_i(\mathbf{x}^p, n\Delta t) &= \sum_q \sum_{\ell=1}^n (\mathbf{T}\mathbf{u})_j(\mathbf{x}^q, \ell\Delta t) \\ &\quad \int_{\partial D} \int (\mathbf{T}\Gamma)_{ij}(\mathbf{x}^p, \mathbf{y}, \tau) M^{n+1-\ell}(\tau) N^q(\mathbf{y}) d\tau dS_y. \end{aligned}$$

if \mathbf{x}^p is on S . In the sequel, we shall call (4) the ‘algebraic equation at $t = n\Delta t$ ’.

The equation (4) shows that the influence coefficients $A_{ij}^{pq}(\ell\Delta t)$ ($\ell = 1, \dots, n$) are required when one computes the solution $u_i(\mathbf{x}^p, n\Delta t)$ and $\varphi_i(\mathbf{x}^p, n\Delta t)$. In BIEM for wave problems in time domain, the influence coefficient matrix is sparse because the influence coefficients vanish if \mathbf{x}^p is in the region not reached by the wave from the source within a certain time interval. Therefore, one has to compute and store only the non zero influence coefficients $A_{ij}^{pq}(n\Delta t)$ at each time step $t = n\Delta t$ ($n = 1, \dots, N_t$) in order to save memory, where N_t is the number of the time steps. As regards the influence coefficients $A_{ij}^{pq}(\ell\Delta t)$ for $\ell = 1, \dots, n-1$, they have already been calculated and stored at a former time step and we simply recall them from the memory. We can thus write the algebraic equation at $t = n\Delta t$ by computing the matrix vector products on the RHS of equation (4).

If all the influence coefficients up to the time step N_t can be stored, one may then compute the influence coefficient matrix with a given time difference only once at each time step. Since the computation of the influence coefficient matrix is dominant in the whole algorithm of BIEM in time domain, the computational time required for a BIE analysis is considered to scale as $O(N_t)$. In the case of large scale problems with a large number of boundary elements or time steps, however, one may not necessarily have enough memory to store all the influence coefficients. Suppose that one can store the influence coefficients $A_{ij}^{pq}(\ell\Delta t)$ for only $\ell = 1, \dots, N_m$ for some N_m smaller than N_t . With a naive approach one recalculates the influence coefficient $A_{ij}^{pq}(\ell\Delta t)$ for $\ell = N_m + 1, \dots, n$ at $t = n\Delta t$ ($n = N_m + 1, \dots, N_t$) in order to construct the algebraic equation in (4) at $t = n\Delta t$

($n > N_m$). This method will be referred to as the conventional BIEM in the sequel. One will then have to call the routine for the calculation of the influence coefficients once at time steps which satisfy $t \leq N_m \Delta t$, and $n - N_m$ times at $t \geq (N_m + 1) \Delta t$. Therefore the routine will be evoked $\frac{(N_t - N_m)(N_t - N_m + 1)}{2} + N_m$ times in total. In large scale problems, N_m is often far less than N_t . Therefore, the computational time will scale as $O(N_t^2)$ which is quite prohibitive (Figure 1).

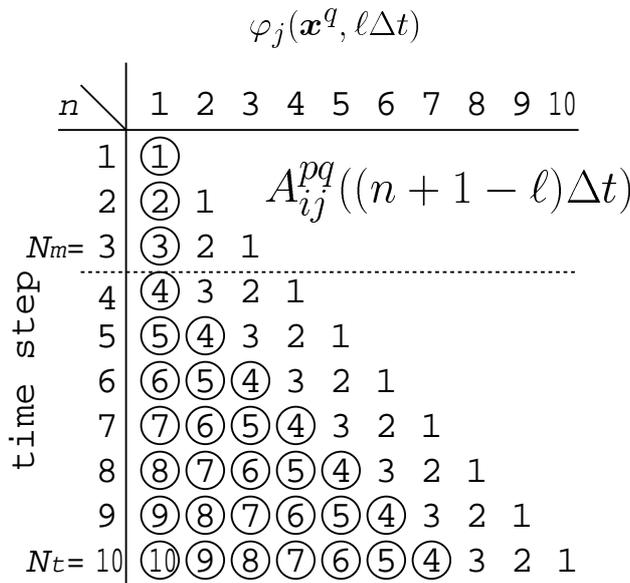


Figure 1: Evocation of the subroutine for the influence coefficients in the conventional time domain BIEM with insufficient memory ($A_{ij}^{pq}(n \Delta t)$ ($n = 1 \cdots N_m$) are stored).

In the next section we shall investigate some improvements of the algorithm in order to carry out a large scale time domain BIE analysis within an appropriate amount of computational time.

2.3 Improvements of the algorithm for time domain BIEM

2.3.1 Improvement of the algorithm for the time convolution

To solve a large scale three dimensional elastodynamic problem within an appropriate amount of time, we improve the algorithm for the time convolution and attempt to reduce the number of the recalculation of the influence coefficients.

In the conventional BIEM, the influence coefficients $A_{ij}^{pq}(n \Delta t)$ are calculated and stored at the time $t = n \Delta t$, and are reused at $t = \ell \Delta t$ ($\ell = n, \dots, N_t$) for the evaluation of the matrix vector products $A_{ij}^{pq}(n \Delta t) u_j(\mathbf{x}^q, (\ell + 1 - n) \Delta t)$

and $A_{ij}^{pq}(n\Delta t)\varphi_j(\mathbf{x}^q, (\ell + 1 - n)\Delta t)$, which are added to the RHS of the algebraic equation at $t = \ell\Delta t$ ($\ell = n, \dots, N_t$). (In the sequel, we shall suppress the terms including u_j for the purpose of simplicity.)

In the proposed implementation, however, we do the following. At $t = n\Delta t$, the solutions $\varphi_j(\mathbf{x}^q, \ell\Delta t)$ for $\ell = 1, \dots, n-1$ are already known. It is therefore possible to compute the matrix vector products $A_{ij}^{pq}(n\Delta t)\varphi_j(\mathbf{x}^q, \ell\Delta t)$ for $\ell = 1, \dots, \min(n-1, N_t + 1 - n)$ at $t = n\Delta t$. The results are then added to the RHS of the algebraic equation at $t = (\ell + n - 1)\Delta t$. This approach is similar to the ‘project forward’ technique proposed by Walker et al.[14, 15, 16]. With this idea, the influence coefficient matrix $A_{ij}^{pq}(n\Delta t)$ ($n = [\frac{N_t+3}{2}] \dots, N_t$) (where $[a]$ is the Gauss’ symbol) is calculated only once at $t = n\Delta t$ and there is no need to store it. In other words, only the influence coefficient matrix $A_{ij}^{pq}(n\Delta t)$ for $n = 1, \dots, [\frac{N_t+1}{2}]$ will be reused. Therefore, one can store all the reusable influence coefficient matrices if $N_m \geq [\frac{N_t+1}{2}]$ holds.

But N_m is often smaller than $[\frac{N_t+1}{2}]$ in large scale problems. In such cases, we propose to construct the algebraic equations as in the following:

1. For $n = 1, \dots, N_m$, $A_{ij}^{pq}(n\Delta t)$ is calculated and stored at the time $t = n\Delta t$. The terms including $A_{ij}^{pq}(\ell\Delta t)$ for $\ell = 1, \dots, N_m$ in the RHS of the algebraic equation at $t = n\Delta t$ are computed as in the conventional approach.
2. $A_{ij}^{pq}(n\Delta t)$ for $n = N_m + 1, \dots, [\frac{N_t+1}{2}]$ are required in the matrix vector products with $\varphi_j(\mathbf{x}^q, \ell\Delta t)$ for $1 \leq \ell \leq N_t + 1 - n$: these products are needed in the computation of the RHS for $t = (\ell + n - 1)\Delta t$. However, one can compute only the matrix vector products of the form $A_{ij}^{pq}(n\Delta t)\varphi_j(\mathbf{x}^q, \ell\Delta t)$ for $1 \leq \ell \leq n - 1$ at $t = n\Delta t$, since $\varphi_j(\mathbf{x}^q, \ell\Delta t)$ for other ℓ s are unknown. Therefore, we recalculate $A_{ij}^{pq}(n\Delta t)$ at time steps given by $t = \{k(n - 1) + 1\}\Delta t$ where k is a natural number such that $k(n - 1) + 1 \leq N_t$ holds. We then multiply $\varphi_j(\mathbf{x}^q, \ell\Delta t)$ for $(k - 1)(n - 1) + 1 \leq \ell \leq \min(k(n - 1), N_t + 1 - n)$ by $A_{ij}^{pq}(n\Delta t)$ and add the results to the RHS of the algebraic equation at $t = (\ell + n - 1)\Delta t$. The routine for the influence coefficient matrix $A_{ij}^{pq}(n\Delta t)$ ($n = N_m + 1, \dots, [\frac{N_t+1}{2}]$) is called $[\frac{N_t-1}{n-1}]$ times in total.
3. For $n = [\frac{N_t+3}{2}], \dots, N_t$, $A_{ij}^{pq}(n\Delta t)$ is calculated once for all and multiplied by $\varphi_j(\mathbf{x}^q, \ell\Delta t)$ ($1 \leq \ell \leq \min(n - 1, N_t + 1 - n)$) at $t = n\Delta t$. Such A_{ij}^{pq} are never stored. The results of the matrix vector products are added to the algebraic equation at $t = (\ell + n - 1)\Delta t$.

In this method, the number of calls of the routine for the influence coefficient matrix is $N_m + \left[\frac{N_t}{2}\right] + \sum_{\ell=N_m+1}^{[\frac{N_t+1}{2}]} \left[\frac{N_t-1}{\ell-1}\right]$ in total (Figure 2). This number is much less than the corresponding number of evocation in the conventional version.

$$\varphi_j(\mathbf{x}^q, \ell\Delta t)$$

$n \setminus$	1	2	3	4	5	6	7	8	9	10
1	①									
2	②	1								
$N_m=3$	③	2	1							
4	④	3	2	1						
5	⑤	4	3	2	1					
6	⑥	5	4	3	2	1				
7	⑦	6	5	④	3	2	1			
8	⑧	7	6	5	4	3	2	1		
9	⑨	8	7	6	⑤	4	3	2	1	
$N_t=10$	⑩	9	8	7	6	5	④	3	2	1

Figure 2: Evocation of the subroutine for the influence coefficients in the improved time domain BIEM.

As an example we consider a time domain elastodynamic problem with $N_t = 10$. If the influence coefficient matrix $A_{ij}^{pq}(n\Delta t)$ for $n = 1, \dots, 3$ can be stored, the number of times that the routine for the influence coefficient matrix is called is 31 with the conventional time domain BIEM as shown in Figure 1. This number is reduced to 13 with the proposed method as seen in Figure 2.

2.3.2 Improvement of the algorithm for the time interpolation function

Next we focus our attention on the algorithm for the time interpolation function used in the discretization of the boundary integral equation. In this paper, we consider piecewise linear time interpolation functions.

The piecewise linear time interpolation function $M^n(t)$ defined by

$$M^n(t) = \begin{cases} \frac{t - (n-2)\Delta t}{\Delta t} & (n-2)\Delta t \leq t < (n-1)\Delta t, \\ \frac{n\Delta t - t}{\Delta t} & (n-1)\Delta t \leq t < n\Delta t, \\ 0 & t < (n-2)\Delta t, t \geq n\Delta t, \end{cases}$$

is rewritten in the following form:

$$M^n(t) = M_{\text{tri}}^n(t) - 2M_{\text{tri}}^{n-1}(t) + M_{\text{tri}}^{n-2}(t). \quad (5)$$

with the help of the function $M_{\text{tri}}^n(t)$ defined by

$$M_{\text{tri}}^n(t) = \begin{cases} \frac{n\Delta t - t}{\Delta t} & t < n\Delta t \\ 0 & \text{otherwise} \end{cases}$$

See Figure 3.

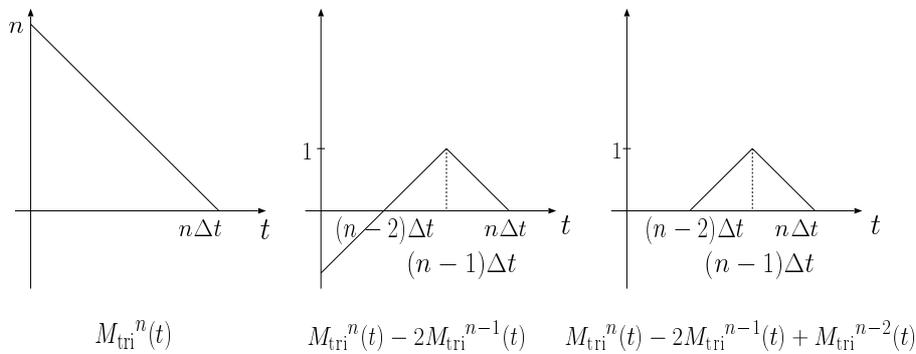


Figure 3: Decomposition of $M^n(t)$

With equation (5), one rewrites the influence coefficient $A_{ij}^{pq}(n\Delta t)$ in the following form:

$$A_{ij}^{pq}(n\Delta t) = A_{\text{tri}ij}^{pq}(n\Delta t) - 2A_{\text{tri}ij}^{pq}((n-1)\Delta t) + A_{\text{tri}ij}^{pq}((n-2)\Delta t), \quad (6)$$

where

$$A_{\text{tri}ij}^{pq}(n\Delta t) = \int_{\partial D+S} \int \Gamma_{Iij}(\mathbf{x}^p, \mathbf{y}, \tau) M_{\text{tri}}^n(\tau) N^q(\mathbf{y}) d\tau dS_y \quad \mathbf{x}^p \in \partial D,$$

$$A_{\text{tri}ij}^{pq}(n\Delta t) = \int_{\partial D+S} \int (\text{T}\Gamma_I)_{ij}(\mathbf{x}^p, \mathbf{y}, \tau) M_{\text{tri}}^n(\tau) N^q(\mathbf{y}) d\tau dS_y \quad \mathbf{x}^p \in S.$$

Obviously, it is more expensive to calculate the three integrals on the RHS of equation (6) at $t = n\Delta t$, compute $A_{ij}^{pq}(n\Delta t)$ from equation (6) and store the result in the memory than to calculate and store just $A_{\text{tri}ij}^{pq}(n\Delta t)$ at $t = n\Delta t$ to obtain $A_{ij}^{pq}(n\Delta t)$, using $A_{\text{tri}ij}^{pq}((n-1)\Delta t)$ and $A_{\text{tri}ij}^{pq}((n-2)\Delta t)$ obtained in previous time steps. We therefore calculate and store $A_{\text{tri}ij}^{pq}(n\Delta t)$, rather than $A_{ij}^{pq}(n\Delta t)$, at $t = n\Delta t$. With this method the time required for the integration at each time step is expected to be reduced to one third of the corresponding time using $A_{ij}^{pq}(n\Delta t)$ instead. The additional cost one has to pay with this approach is

that one has to perform the matrix vector product operation three times since the required product $A_{ij}^{pq}(n\Delta t)\varphi_j(\mathbf{x}^q, \ell\Delta t)$ is decomposed as follows:

$$\begin{aligned} A_{ij}^{pq}(n\Delta t)\varphi_j(\mathbf{x}^q, \ell\Delta t) &= A_{\text{tri}ij}^{pq}(n\Delta t)\varphi_j(\mathbf{x}^q, \ell\Delta t) \\ &- 2A_{\text{tri}ij}^{pq}((n-1)\Delta t)\varphi_j(\mathbf{x}^q, \ell\Delta t) \\ &+ A_{\text{tri}ij}^{pq}((n-2)\Delta t)\varphi_j(\mathbf{x}^q, \ell\Delta t). \end{aligned}$$

Notice that the three terms on the RHS of the above equation are evaluated at different time steps, i.e. $t = n\Delta t$, $t = (n-1)\Delta t$ and $t = (n-2)\Delta t$, respectively. Therefore the improvement of the computational time achieved with the proposed method is expected to be less than three times of the original.

2.3.3 Parallelization

We seek further enhancement of the efficiency of the code by parallelizing the algorithm.

The parallelization strategy used in this paper is a rather standard one which can be described as follows: Since the matrix equation to be solved is a sparse one, the solution process itself is relatively light in terms of the computational load, and can be performed with only one process called ‘master’. In this paper we use GMRES as the solver. The most significant computational load is actually found in the evaluation of the RHS which takes the form of the sum of matrix-vector products. In the present implementation we distribute boundary elements to processes, and each process computes influence coefficients of elements known to the process for all the collocation points. In other words the influence coefficients $A_{ij}^{pq}(\cdot, n\Delta t)$ are distributed column-wise to processes. The contributions to the RHS from each of the processes are then sent to the master process and summed.

In distributing boundary elements to processes one has to take the load balancing into consideration. This is particularly so because we are interested in a time domain BIEM where some of the matrices are sparse. In BIEM, however, one can distribute boundary elements quite freely to processes, thus making it relatively easy to achieve good load balancing. This is in contrast to FEM or FDM where one usually uses domain decomposition to distribute unknowns to processes. In the present implementation, we number boundary elements in a systematic manner (e.g. in a spiral manner for a penny shaped crack) and then distribute boundary elements to processes in a cyclic manner. The cyclic distribution has an effect of making the distribution of boundary elements in processes rather ‘uniform’ and thus helps to achieve a good load balancing.

2.4 Numerical example

2.4.1 Scattering of P wave by a penny shaped crack

To examine the improvement of the efficiency achieved by the modifications proposed in the previous section, we solve a simple scattering problem for a crack S

in a three dimensional infinite elastic region D . We solve this problem with the improved and the conventional methods and compare the performance.

The crack is assumed to be penny shaped with the radius of a , and the incident wave is a plane P wave propagating into the direction normal to the crack (taken to be the x_3 direction) and having a constant stress magnitude of $\sigma_{33} = p$ (Figure 4). The number of the boundary elements is 2680 (8040 DOF), the number of time steps N_t is 30 and the Poisson ratio ν is 0.25. Also, the time increment Δt is set equal to $0.09a/c_T$, where c_T is the velocity of the S wave. The computation is carried out with a cluster of 4 SMP PCs having 2 Compaq Alpha EV67 (666MHz) as CPUs and 512MB of RAM. The PCs are connected with a Myrinet network. We use MPICH to run parallel codes implemented with MPI[21].

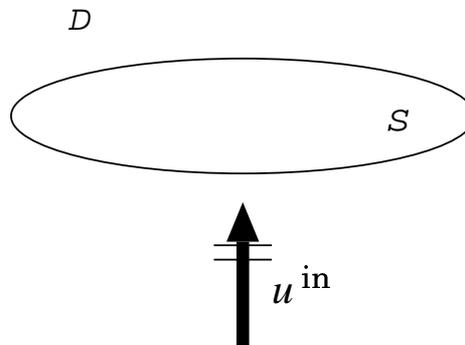


Figure 4: Wave scattering problem by a penny shaped crack S .

Figure 5 shows the time history of the x_3 component of the crack opening displacement for $t \leq 11\Delta t$. The results agree well with those obtained by Hirose and Achenbach[22]. Table 1 shows the comparison of computational time(sec) required by conventional and improved methods. In this case N_m is 7 with the sequential code. The conventional method requires the wall time of 26764 seconds, which

Table 1: Computational time (sec).

	# CUP	wall time(sec.)
conventional BIEM	1	26764
improved BIEM	1	1877
improved parallel BIEM	8	188

is reduced to 1877 seconds with the algorithm improvement. This result shows that the efficiency of the improved approach is remarkable. Also noteworthy is the superlinear reduction of the computational time achieved by the parallelization. This superlinearity is partly due to the fact that one can increase N_m in

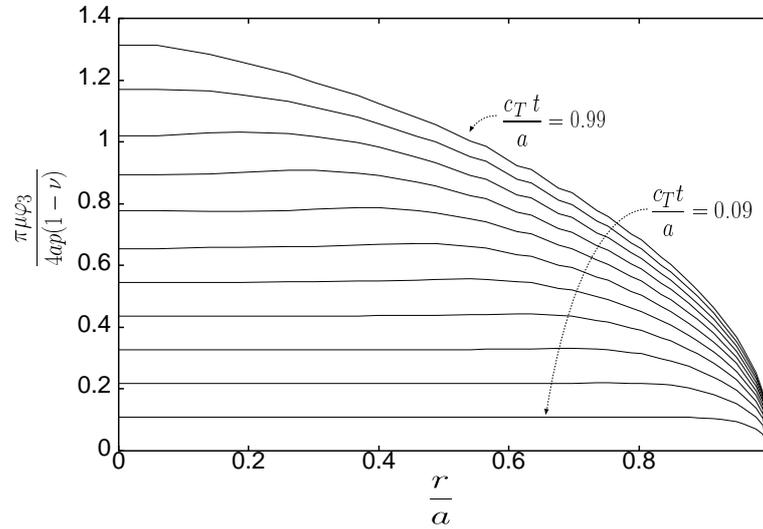


Figure 5: Opening displacement of the crack S .

the parallel code simply because more memory is available. Indeed, N_m has been increased to 15 in the 8 process case. This superlinearity is more evidently seen in Figure 6 in which the wall time \times the number of processes (N_{procs}) is plotted vs N_{procs} . We could thus reduce the computational time to 1/142 of the original with the algorithm improvements and parallelization.

3 Application to large scale elastodynamic problems related to ultrasonic NDE

In this section we apply the improved parallel code to the analysis of the real waveform data obtained in an ultrasonic NDE.

In the ultrasonic testing under consideration (See Figure 7), we set an ultrasonic transducer (central frequency = 500 KHz, diameter = 14.5mm) on the surface of the specimen made of aluminum alloy which has a rectangular surface crack S , which actually is a saw cut having an opening of 0.2mm. The location and the length (=10.5mm) of the crack are known since it is visible on the surface of the specimen, but the depth and the angle of inclination of S are unknown. To determine these quantities, we excite the transducer with an electric pulser, thus generating an incident wave which illuminates the crack. We then use a laser interferometer to measure the normal velocities at several points on the surface (L_i, R_i shown in Figure 7). The depth and the angle of inclination of the surface crack are determined from the velocimetry data.

In this NDE problem, we have to determine the elastic wave generated from

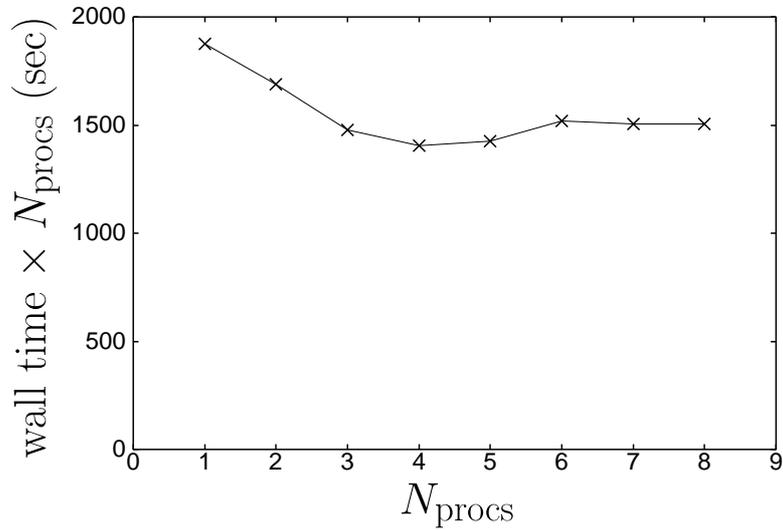


Figure 6: Parallel performance of the code (N_{procs} : number of processes)

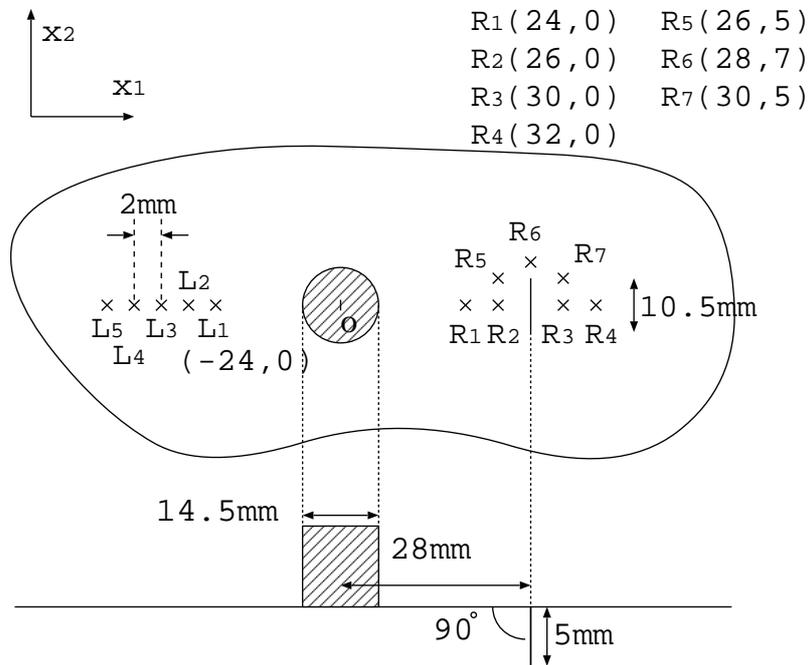


Figure 7: Specimen and the points of measurements.

the transducer, which is unknown. To determine this wave, we model the action of the transducer as the time varying load p distributed under the transducer. As we found[3], however, the spatial distribution of p is rather uniform, and a relatively simple modeling for the behavior of p is permissible. We thus model the action of the transducer by two uniform loads denoted by p_1, p_2 . These equivalent loads p_j ($j = 1, 2$) are applied on the surface area denoted by ∂D_{t_j} , where ∂D_{t_1} is the circular area having the radius of three eighths of the radius of the transducer and ∂D_{t_2} is the rest of the area under the transducer, respectively. We determine the time variations of these loads using the laser velocimetry data obtained at points, denoted by L_i ($i = 1, \dots, N$), where the observed velocities are not affected by the scattered waves from the crack. Indeed, $p_j(t)$ ($j = 1, 2$) can be obtained as solutions to a system of integral equations given by[3]

$$V_L^i(t) = \sum_{j=1}^2 \int_0^t k_j^i(t-s) \dot{p}_j(s) ds, \quad (i = 1, \dots, N) \quad (7)$$

where $k_j^i(t)$ is the normal displacement at the point L_i produced by the uniform load having a Dirac delta time variation distributed on ∂D_{t_j} , $V_L^i(t)$ is the normal velocity at L_i obtained with laser velocimetry, and N is the number of experiments. The function k_j^i can be computed numerically as one integrates the well-known point load solution by Lamb. The integral equation in (7) can be solved numerically with the help of Tikhonov's regularization and the L-curve[3]. Figure 8 shows the reconstructed \dot{p}_j used in the present investigation. The elastic wave from the transducer is now reconstructed as one solves a direct elastodynamic problem with the equivalent loads as the boundary condition.

We next solve the inverse problem of determining the depth and the angle of inclination of the crack. To this end we introduce shape parameters d and θ which indicate the depth and the angle of inclination of the crack, respectively. We then solve the following initial boundary value problem via the improved time domain BIEM using the reconstructed \dot{p}_j as part of the boundary conditions:

$$\begin{aligned} \mu v_{i,kk} + (\lambda + \mu) v_{k,ik} &= \rho \ddot{v}_i \quad \text{in } D \times (t > 0) \\ (\mathbf{T}\mathbf{v})_i &= -\dot{p}_j(t) n_i \quad \text{on } \partial D_{t_j} \times (t > 0) \\ (\mathbf{T}\mathbf{v})_i &= 0 \quad \text{on } (\partial D \setminus \partial D_{t_j}) \times (t \geq 0), \\ (\mathbf{T}\mathbf{v})_i^\pm &= 0 \quad \text{on } S \times (t \geq 0) \\ v_i|_{t=0} &= \dot{v}_i|_{t=0} = 0 \quad \text{in } D \\ \dot{\varphi}_i &= v_i^+ - v_i^- = 0 \quad \text{on } \partial S, \end{aligned}$$

where D is the domain of the specimen with the crack which is described with shape parameters (d, θ) , ∂D is the boundary of the domain D , and \mathbf{n} is the unit normal vector to ∂D . Obviously, the solution v_i to this problem is a function of (d, θ) . We then introduce the cost function J defined by

$$J = \sum_i \sum_m (V^i(m\Delta t) - \mathbf{v}^i(d, \theta, m\Delta t) \cdot \mathbf{n})^2, \quad (8)$$

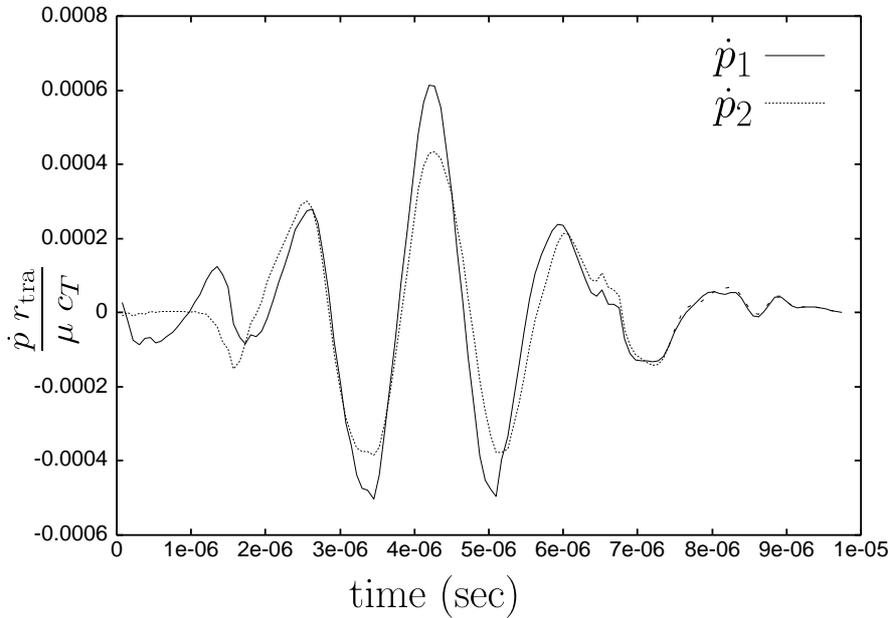


Figure 8: $\dot{p}_1(t), \dot{p}_2(t)$

where $V^i(t)$ are the measured velocities at points R_i and $v^i(t)$ are the calculated velocities at R_i . The points R_i are taken so that the velocities there are affected by the scattered wave from the crack. The solution (d, θ) to our inverse problem is found as the minimizer of this cost.

In our numerical example, the P wave and S wave velocities are 6180(m/sec) and 3180(m/sec), the time increment Δt is 0.075 (μsec), the number of the time steps is 130, and the number of the boundary elements is 10766 (DOF: 32298), respectively. The cost function J is minimized with an NLP (Quasi-Newton method), while the gradients of J are computed with the finite difference. The improved parallel BIEM is used on Fujitsu VPP800 (vector-parallel supercomputer) with 10CPUs in the computation of the cost J in the NLP process. The measured velocities at R_i (see Figure 9 for typical input data) are thus inverted to yield $(d, \theta) = (4.84\text{mm}, 89.68^\circ)$ as the minimizer of the cost function J , which is quite close to the true value $(d, \theta) = (5\text{mm}, 90^\circ)$. As a check we compared in Figure 9 the measured velocities at R_2 and R_3 and the calculated velocities at the same points obtained using the true shape parameters (d, θ) . The agreement is quite satisfactory.

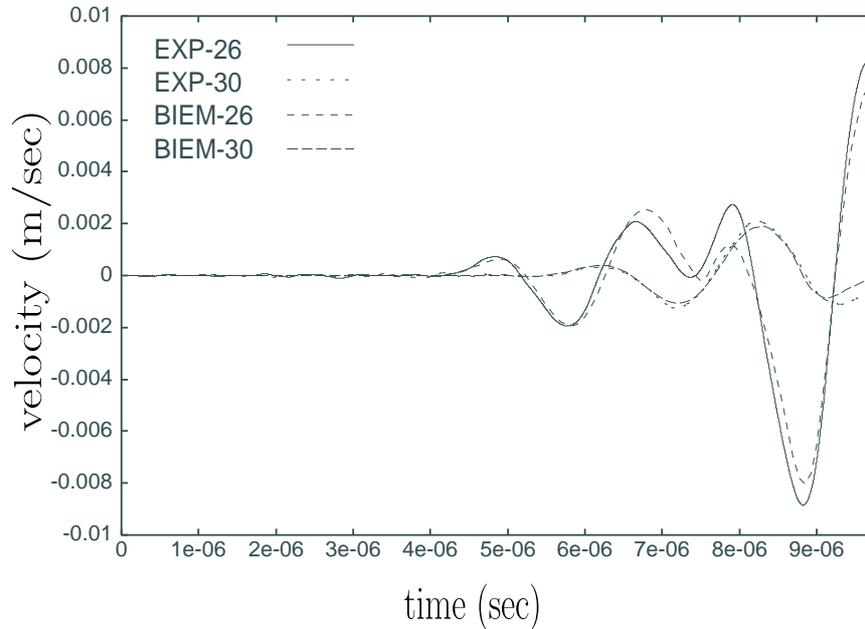


Figure 9: Measured (EXP) and calculated (BIEM) velocities at R_2 and R_3

4 Conclusion

In this paper we have presented an improved time domain BIEM for three dimensional elastodynamics. This implementation saves the computational time by avoiding recalculation of the influence coefficients in problems where one cannot store all these coefficients in the memory. We have also investigated the parallelization of the code. The efficiency of the improved code has been tested in a simple wave scattering problem, as well as in a large scale time domain problem related to ultrasonic NDE.

We have seen that the inverse problem we have considered in this paper requires a very large analysis, although it is still considered to be an academic one. One may well conclude that the real world problems will need still larger analysis. We believe that the approaches based on fast methods are the way to go, and we are trying to further improve the efficiency of the present elastodynamic BIEM code in time domain with the help of fast multipole methods.

References

- [1] Nishimura, N., A boundary integral equation method for solving elastodynamic crack determination problems, *Proc. 2nd Mint. Conf. on Math. and*

- Num. Aspects of Wave Propagation* (Eds. R. Kleinman et al.), SIAM, 390–397, 1993.
- [2] Kanbayashi, T., Yoshikawa, H., Nishimura, N. and Kobayashi, S., Verification of ultrasonic transducer characteristics determined in an inverse problem based on laser measurements, *Inverse Problems in Engineering Mechanics II*, (Eds. M. Tanaka and G. S. Dulikravich), Elsevier, 327–332, 2000.
 - [3] Yoshikawa, H., Nishimura, N. and Kobayashi, S., On the determination of ultrasonic waves emitted from transducers using laser measurements with applications to defect determination problems, *J. Appl. Mech. JSCE*, **4**, 145–152, 2001.
 - [4] Friedman, M.B. and Shaw, R.P., Diffraction of pulses by cylindrical obstacles of arbitrary cross section, *J. Appl. Mech.*, **29**, 40–46, 1962.
 - [5] Shaw, R.P., Retarded potential approach to the scattering of elastic pulses by rigid obstacles of arbitrary shape, *J. Acoust. Soc. Am.*, **44**, 745–748, 1967.
 - [6] Banerjee, P.K. and Shaw, R.P., *Developments in boundary element methods-2*, Elsevier Applied Science Publishers, 1982.
 - [7] Banerjee, P.K. and Kobayashi, S., Advanced dynamic analysis by boundary element methods, *Developments in boundary element method-7*, Elsevier Applied Science Publishers, 1992.
 - [8] Beskos, D.E., Boundary element methods in dynamic analysis, *Appl. Mech. Reviews*, **40**, 1–23, 1987.
 - [9] Beskos, D.E., Boundary element methods in dynamic analysis: Part II (1986–1996), *Appl. Mech. Reviews*, **50**, 149–197, 1997.
 - [10] Bonnet, M., *Boundary integral equation methods for solids and fluids*, Wiley & Sons, 1995.
 - [11] Kobayashi, S. et al., *Wave analysis and boundary element methods*, Kyoto University Press, 2000 (in Japanese).
 - [12] Miller, E.K., Time-domain modeling in electromagnetics, *J. Electromagn. Waves and Appl.*, **8**, 1125–1172, 1994.
 - [13] Demirel, V. and Wang, S., An efficient boundary element method for two-dimensional transient wave propagation problems, *Appl. Math. Model.*, **11**, 411–416, 1987.
 - [14] Walker, S.P., Scattering analysis via time-domain integral equations: Methods to reduce the scaling of cost with frequency, *IEEE Ant. Prop. Mag.*, **39**, 13–20, 1997.

- [15] Walker, S.P. and Lee, B.H., Reduced-cost methods for large time domain integral equation scattering analyses, *Commun. Num. Meth. Eng.*, **14**, 751–761, 1998.
- [16] Dodson, S.J., Walker, S.P. and Bluck, M.J., Costs and cost scalings in time domain integral equation analysis of electromagnetic scattering, *IEEE Ant. Prop. Mag.*, **40**, 12–21, 1998.
- [17] Ergin, A.A., Shanker, B. and Michielssen, E., Fast evaluation of three dimensional transient wave fields using diagonal translation operators, *J. Comp. Phys.*, **146**, 157–180, 1998.
- [18] Takahashi, T., Nishimura, N. and Kobayashi, S., A fast BIEM for three-dimensional elastodynamics in time domain, *Eng. Anal. Boundary Elements*, **27**, 491–506, 2003.
- [19] Dodson, S.J., Walker, S.P. and Bluck, M.J., Parallelisation issues for high speed time domain integral equation analysis, *Parallel Comp.*, **25**, 925–942, 1999.
- [20] Yoshikawa, H., Nishimura, N. and Kobayashi, S., Algorithm improvements and parallel computing of the three dimensional elastodynamic boundary integral equation method in time domain, *J. Appl. Mech. JSCE*, **5**, 199–206, 2002(in Japanese).
- [21] Gropp, W., Lusk, E. and Skjellum, A., *Using MPI: Portable Parallel Programming with Message-Passing Interface, second edition*, MIT Press, 1999.
- [22] Hirose, S. and Achenbach, J.D., Acoustic emission and near-tip elastodynamic field of a growing penny-shaped crack, *Eng. Fract. Mech.*, **39**, 21–36, 1991.