Copyright © 2006 Tech Science Press CMES, vol.16, no.2, pp.131-140, 2006

A Fast Space-Time BEM Method for 3D Elastodynamics

J. X. Zhou [1], T. Koziara[1] and T. G. Davies[1]

Abstract: The classical BEM approach for elastodynamics can produce poor results when high gradients are generated by impulses. High gradient areas evolve over time and their locations are unknown a priori, so they usually can not be captured by uniform meshes. In this paper, we propose a novel method which interpolates both spatial and temporal domains. A direct space-time discretization scheme is used to capture the wave fronts accurately and to forestall generation of spurious oscillations there. Some numerical examples are given to demonstrate the power and scope of the method.

keyword: Time-domain BEM, Elastodynamics, Impulse load, Fast spatial search

1 Introduction

The boundary element method is commonly used to solve acoustic and elastodynamic problems in unbounded domains. However, the algorithmic instability of the time-domain Boundary Element method for elastodynamics and scalar wave propagation has been a major numerical difficulty [Beskos (1997), Mansur and Brebbia (1985), Banerjee and Kobayashi (1992), Manolis and Beskos (1988), Pozrikidis (2002)]. To address this problem, various spatial and temporal interpolation schemes have been implemented to improve accuracy and stability, such as combinations of constant, linear and quadratic functions [Dominguez (1993), Mansur and Carrer (1998)], B-splines interpolation schemes [Rizos and Karabalis (1994)] and quadratic time interpolation schemes [Wang and Wang (1996)]. Other strategies to improve stability include the linear θ method [Yu and Mansur (1998)], the time discontinuous traction method [Mansur and Carrer (1998)], the half-step method [Birgisson, Siebrits, and Peirce (1999)], and an integral formulation combining several time steps [Marrero and Dominguez (2003)]. Some new approaches are developed to solve elastodynamics problems, such as employing nonsingular approaches for acoustic problems[Callsen, von Estorff, and Zaleski (2004), Qian, Han, Ufimtsev, and Atluri (2004)], BEM/FEM coupling to solve structural-acoustic problems [Soares and Mansur (2005), Lie, Yu, and Zhao (2001)], time-domain BEM for crack analysis [Zhang and Savaidis (2003)]. However, most of these methods do not address the problem of how to approximate wave fronts accurately. It is well known [Peirce and Siebrits (1997), Frangi and Novati (1999)] that the time-domain BEM often produces poor results because the locations of wave fronts are unknown *a priori*, and they can not be properly approximated by uniform meshes and ordinary polynomial functions. Spurious oscillations and instabilities are often observed in these regions. Alternatively, dispersion occurs if lower-order approximations or larger time steps are used to damp the oscillations. These high-gradient areas must be tracked and accurately modeled by more flexible approximation schemes.

In this paper we propose a novel approach to solve large-scale elastodynamic problems, in which the space-time domain is discretized in a true sense, combined with a fast spatial search algorithm to accelerate the generation of the influence matrices. The rest of the paper is organized as follows: in Section 2, the space-time approach for 3D elastodynamics BEM is introduced. The fast spatial search algorithm is presented in Section 3. Some illustrative examples of the application of this method are given in Section 4.

2 Space-time BEM for 3D elastodynamics

The standard BEM employs a finite difference methodology in time and a boundary element discretization in space. However, regions of high stress or high strain evolve over time and exhibit a high degree of localization at any instant. Using smaller time steps for all regions is computationally wasteful. Thus, building a BEM model in which the space-time continuum is discretized in a true sense should be more efficient.

[1] Department of Civil Engineering, Glasgow University, Glasgow, G12 8LT, UK

2.1 Boundary integral formulation for space-time BEM elastodynamics

The governing equations of elastodynamics are:

$$(\mathrm{c}_1^2-\mathrm{c}_2^2)\boldsymbol{u}_{i,ij}+c_2^2\boldsymbol{u}_{j,ii}+\boldsymbol{f}_j=\ddot{\boldsymbol{u}}_j \tag{1}$$

with initial and boundary conditions

$$\begin{array}{ll} u_i(\boldsymbol{x},0^+)|_\Gamma=u_{0i}(\boldsymbol{x}) & \dot{u}_i(\boldsymbol{x},0^+)|_\Gamma=v_{0i}(\boldsymbol{x}) \\ u_i(\boldsymbol{x},t)|_{\Gamma_1}=\overline{u}_i(\boldsymbol{x},t) & p_i(\boldsymbol{x},t)|_{\Gamma_2}=\overline{p}_i(\boldsymbol{x},t) \end{array} \tag{2}$$

where the boundary $\Gamma=\Gamma_1+\Gamma_2$, $u(\boldsymbol{x},t)$ is the displacement in 4D space-time, $p(\boldsymbol{x},t)$ are tractions on the boundary, $\boldsymbol{x}$ denotes the spatial coordinates (x,y,z), $f_j=b_j/\rho$, b_j is the body force, ρ is the density, c_1 is the dilatational wave speed and c_2 is the shear wave speed. Assuming zero body forces and initial conditions, the 3D elastodynamic BEM integral equation can be written as [Dominguez (1993)]:

$$\begin{aligned} c_{lk}^i u_k^i(\boldsymbol{x}^i,t)&=\int_\Gamma u_{lk}^*(\boldsymbol{x},t-\tau;\boldsymbol{x}^i)*p_k(\boldsymbol{x},\tau)\,d\Gamma(\boldsymbol{x},\tau)\\ &-\int_\Gamma p_{lk}^*(\boldsymbol{x},t-\tau;\boldsymbol{x}^i)*u_k(\boldsymbol{x},\tau)d\Gamma(\boldsymbol{x},\tau) \end{aligned} \tag{3}$$

where, $\boldsymbol{x^i}$ refers to a specific collocation point on Γ. The Green's function u_{lk}^* is given by Dominguez (1993)

$$\begin{aligned} u_{lk}^*(\boldsymbol{x},t;\boldsymbol{x}^i)&=\frac{1}{4\pi\rho}\{\frac{t}{r^2}\left(\frac{3r_{,l}r_{,k}}{r}-\frac{\delta_{lk}}{r}\right)\\ &[H\left(t-\frac{r}{c_1}\right)-H\left(t-\frac{r}{c_2}\right)]\}\\ &+\frac{1}{4\pi\rho}\{\frac{r_{,l}r_{,k}}{r}[\frac{1}{c_1^2}\delta(t-\frac{r}{c_1})-\frac{1}{c_2^2}\delta(t-\frac{r}{c_2})]\\ &+\frac{\delta_{lk}}{rc_2^2}\delta(t-\frac{r}{c_2})\} \end{aligned} \tag{4}$$

where $r=\left|\boldsymbol{x}-\boldsymbol{x}^i\right|$, $H(x)$ is the Heaviside function and $\delta(x)$ is the Dirac-delta function. The traction kernel function $p_{lk}^*(\boldsymbol{x},t;\boldsymbol{x}^i)$ can be obtained from the Green's function u_{lk}^* , using Hooke's law and equilibrium, which yields:

$$\begin{aligned} p_{lk}^*(\boldsymbol{x},t;\boldsymbol{x}^i)&=\frac{1}{4\pi}\{\left(\frac{\partial r}{\partial n}\delta_{lk}+r_{,k}n_l\right)A\\ &+r_{,l}r_{,k}\frac{\partial r}{\partial n}B+r_{,l}n_kC\} \end{aligned} \tag{5}$$

where the coefficients A,B,C can be found in Mansur and Brebbia (1985).

In equation (3), the integrals of the products of $p_k(\boldsymbol{x},\tau)$ and the Heaviside function $H\left(t-\frac{r}{c}\right)$ or the Dirac-delta function $\delta\left(t-\frac{r}{c}\right)$ in space-time become spatial integrals of $p_k(\boldsymbol{x},\tau)$ over the boundary. Most of the integrals of the product of $u_k(\boldsymbol{x},\tau)$ and the fundamental traction solution can be similarly reduced to spatial integrals. The integral of the product of displacement $u_k(\boldsymbol{x},\tau)$ and the time derivative of the Dirac-delta function reduces to:

$$\int\frac{\partial\delta(t-\frac{r}{c_1})}{\partial\tau}u_k(\boldsymbol{x},\tau)d\Gamma(\boldsymbol{x},\tau)=\int\frac{\partial u_k(\boldsymbol{x},t-\frac{r}{c_1})}{\partial\tau}d\Gamma(\boldsymbol{x}) \tag{6}$$

Here, we introduce a new interpolation scheme. Not only is the spatial domain interpolated via shape functions, but also both spatial and temporal domains are interpolated using $N^k(\xi,\eta,\tau)$ as a space-time interpolation function. Thus:

$$\begin{aligned} u_k&=\sum_j N^k(\xi,\eta,\tau)\,u_k^j; \qquad p_k=\sum_j N^k(\xi,\eta,\tau)\,p_k^j;\\ t_k&=\sum_j N^k(\xi,\eta,\tau)\,t_k^j; \end{aligned} \tag{7}$$

Then, distance in space-time is defined as:

$$\begin{aligned} |ct-r|&=\left|c(t_i-\sum N^k(\xi,\eta,\tau)\cdot t_j^k)-r\right|\\ &=|c(t_i-\Phi(\xi,\eta,\tau))-r| \end{aligned} \tag{8}$$

where, $\sum N^k(\xi,\eta,\tau)\cdot t_j^k=\Phi(\xi,\eta,\tau)$.

After this numerical approximation, Eq. (3) takes the form

$$\begin{aligned} c_{lk}^i u_k^i=\sum_{j=1}^N(&\int_{\Gamma_j}u_{lk}^*\cdot N^j(\xi,\eta,\tau)\,p_k^j d\Gamma(\boldsymbol{x},\tau)\\ &-\int_{\Gamma_j}p_{lk}^*\cdot N^j(\xi,\eta,\tau)\,u_k^j d\Gamma(\boldsymbol{x},\tau)) \end{aligned} \tag{9}$$

Letting

$$\begin{aligned} G_{lk}^{ij}&=\int_{\Gamma_j}u_{lk}^*\cdot N^j(\xi,\eta,\tau)\,d\Gamma(\boldsymbol{x},\tau)\\ \hat{H}_{lk}^{ij}&=\int_{\Gamma_j}p_{lk}^*\cdot N^j(\xi,\eta,\tau)\,d\Gamma(\boldsymbol{x},\tau) \end{aligned} \tag{10}$$

and adopting the notation:

$$H_{lk}^{ij}=\left\{\begin{array}{ll}\hat{H}^{ij} & i\neq j\\ \hat{H}^{ij}+c_{lk}^i & i=j\end{array}\right. \tag{11}$$

then the system of equations for all boundary nodes can be expressed in the matrix form:

$$\sum_{j=1}^{N} H_{lk}^{ij} u_k^j = \sum_{j=1}^{N} G_{lk}^{ij} p_k^j \tag{12}$$

where N is the total number of nodes in the space-time domain.

The causality law requires that boundary values at later times are only influenced by quantities at early time, but not vice versa. Thus, numerical methods constructed from these space-time boundary integral equations are global in time, i.e., it is necessary to compute the solution for all time steps from the beginning to obtain the current solution. The system is solved step-by-step: once **u** and **p** are known for the previous time steps, the solution for the nth time step is obtained from:

$$\mathbf{H}_{lk}^{(nn)ij}\mathbf{u}_k^{(n)j} = \mathbf{G}_{lk}^{(nn)ij}\mathbf{p}_k^{(n)j} + \sum_{m=1}^{n-1} \mathbf{G}_{lk}^{(nm)ij}\mathbf{q}_k^{(m)j} - \mathbf{H}_{lk}^{(nm)ij}\mathbf{u}_k^{(m)j} \tag{13}$$

The boundary consists of the spatial boundary extruded into the space-time, like a cylinder extruded from 2D to 3D space. The increase in dimensionality is however offset by special features of the problem, which are discussed in Section 2.3.

2.2 Shape functions in space-time

For illustrative purpose, piece-wise linear interpolation functions are used for tractions and displacements. Thus, eight nodes must be defined in 3D space-time for linear interpolation. Denoting the node numbers by the subscript α, we express all eight shape functions ($\alpha = 1-8$) as follows:

$$N_\alpha(\xi_1,\xi_2,\xi_3) = \frac{1}{8}(1+\xi_1^\alpha\xi_1)(1+\xi_2^\alpha\xi_2)(1+\xi_3^\alpha\xi_3) \tag{14}$$

where (ξ_1,ξ_2,ξ_3) denotes the intrinsic coordinates of the αth node, of which two represent the space dimensions and the third represents the time dimension. For quadratic interpolation in space-time, the element is defined by twenty nodes. For the eight corner nodes ($\alpha = 1-8$) , the shape functions are [Gao and Davies (2002)]

$$N_\alpha(\xi_1,\xi_2,\xi_3) = \frac{1}{8}(1+\xi_1^\alpha\xi_1)(1+\xi_2^\alpha\xi_2)(1+\xi_3^\alpha\xi_3) \times(\xi_1^\alpha\xi_1+\xi_2^\alpha\xi_2+\xi_3^\alpha\xi_3-2) \tag{15}$$

and for the twelve remaining mid-side nodes ($\alpha = 9-20$), the shape functions are

$$N_\alpha(\xi_1,\xi_2,\xi_3) = \frac{1}{4}(1+\xi_1^\alpha\xi_1)(1+\xi_2^\alpha\xi_2)(1+\xi_3^\alpha\xi_3) \times(1+[\xi_1^{\alpha 2}-1]\xi_1^2+[\xi_2^{\alpha 2}-1]\xi_2^2 +[\xi_3^{\alpha 2}-1]\xi_3^2) \tag{16}$$

2.3 Time and space integration

For the given shape functions, the influence matrices $[G]$ and $[H]$ are obtained by integrating over the boundary elements using Eq. (10). If the space-time domain $\Gamma(\boldsymbol{x},t)$ is divided into n parts along the time axis, the integrals in Eq. (9) for the mth time interval (t_{m-1},t_m) is the integral over the surface elements that lies within two concentric spherical surfaces of radius $r_m = c(t_n-t_m)$ and $r_{m+1} = c(t_n-t_{m+1})$. Here $c = c_1$ or $c = c_2$ depending on which term is being integrated (Fig. 1).

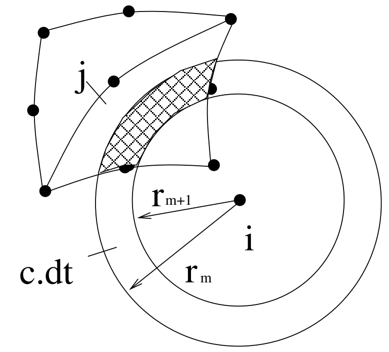

Figure 1 : Element j receives a signal from the collocation point i only during the time interval $t_n-\tau_{m+1} < \frac{r}{c} < t_n-\tau_m$.

For this reason, the system matrices are highly sparse. This special structure follows from the convolution of the Dirac delta function in time in the integral equations. The integration does not extend over the whole boundary of the space-time cylinder, but only over its intersection with the surface of the backward propagating wave cone. This means that the integrals have the same dimensionality as for the static problems, and that current response is not affected, in general, by the events that extend far into the past. These features are important for 3D scalar wave equation in three space dimensions and elastodynamics, but do not apply to the 2D wave equation, nor to the heat equation.

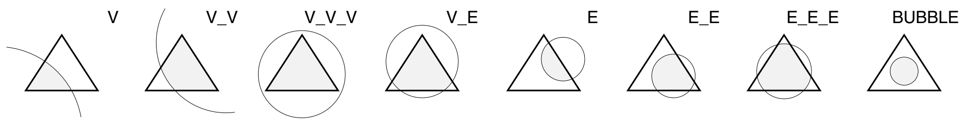

Figure 2 : Eight of the nine possible configurations of triangle-sphere intersection (triangle outside the sphere is omitted).

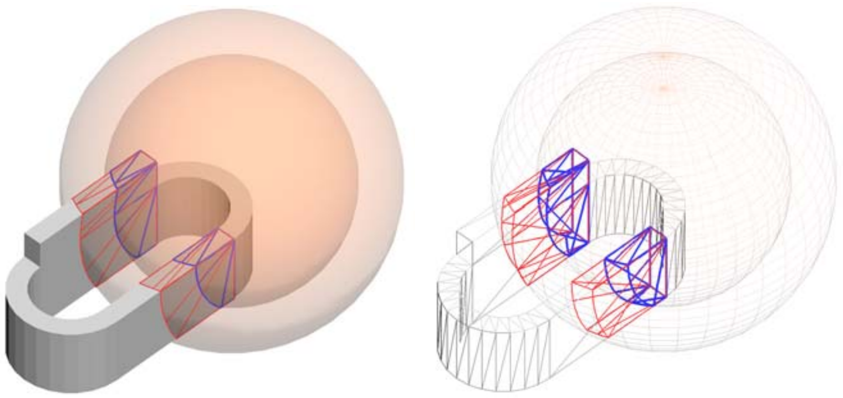

Figure 3 : Intersection of a mechanical component with a pair of spheres. The sub-triangulation boundaries are marked in gray (terms to be added) and black (terms to be subtracted).

If the source node is not within the current space-time element, standard $3 \times 3 \times 3$ Gauss quadrature is sufficient. Otherwise, the singular integrals need careful treatment. For the displacement singularity, the singularity is $O(1/r)$ and can be treated by employing an element subdivision technique to divide the original elements into several tetrahedrons. Then, each tetrahedral subelement can be mapped into a cubical intrinsic element space where the weak singularity is nullified and the integral can be performed using normal Gauss quadrature [Gao and Davies (2002)]. For the traction singularity of $O(1/r^2)$, an indirect method is used which employs the rigid body motion condition to calculate the c^i_{lk} coefficient and H^{ii}_{lk} by using Eq (11). By looping over each node and element, all terms are calculated and assembled into matrices $[G]$ and $[H]$. Once the matrices $[G]$ and $[H]$ have been obtained, the resulting matrix equation is solved by using standard techniques.

3 A fast algorithm for triangle subdivision and global intersection search

For an efficient implementation of the computational framework described here, two purely geometrical problems must be addressed. The first one is the *triangle-sphere intersection* problem, resulting from the integration over the boundary surface contained between the two spheres. The second problem concerns an effective *global search* for the intersecting triangle and sphere pairs.

3.1 Triangle subdivision

The triangle-sphere intersection problem is solved by first identifying which intersection configuration applies (Fig. 2), and sub-triangulating the area contained in the sphere. In order to minimise the number of sub-triangles, while maintaining a good approximation of the intersection boundary, six-node second-order sub-triangles are used. Further, rather than calculating the area contained between the two spheres, two separate sub-triangulations are obtained by intersecting the triangle with the bigger and the smaller sphere. The terms integrated over the first sub-triangulation are added, while those integrated over the second sub-triangulation are subtracted (Fig. 3). This approach is simpler and more efficient than an explicit derivation of the sub-triangulation contained between the two spheres.

3.2 Global intersection search

For typical benchmark problems, composed only of a small number of triangles, the issue of efficient identification of the intersecting triangle-sphere pairs can be ignored. This is not true for fine meshes, where the

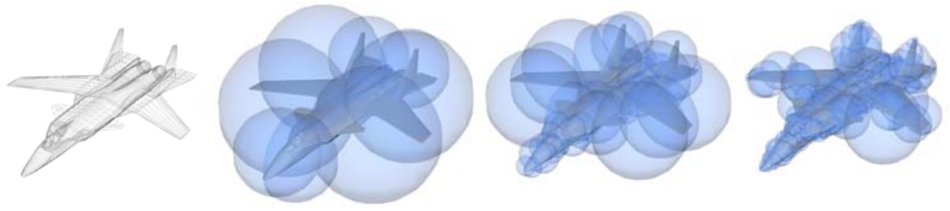

Figure 4 : Example of a sphere-tree hierarchy for an airplane represented by 10904 triangular facets.

global search for the intersecting pairs becomes a computational bottleneck. Working with n nodes and m triangles, it is evidently far from optimal to perform a brute force $O(n \cdot m)$ check, testing all possible pairs of spheres and triangles. This is a *spatial search* problem [Samet (1990)], which is common to many fields such as contact mechanics, computer graphics, etc. The specific features here can be summarised as follows:

1. The mesh geometry and connectivity does not change during the course of simulation (This assumption must be relaxed for the case of an adaptive mesh refinement).

2. The search for an intersection takes place within the volume contained between two spheres whose radii increase with time, while the difference in radii remains constant.

The sphere pairs (the *query spheres*) are centred at the mesh nodes. At each time step, the set of query spheres $\mathcal{S}$ intersects a set of surface triangles $\mathcal{T}$. This generates for each node two sub-triangle lists - one for adding and one for subtracting terms during the system matrix assembly. In order to facilitate an efficient identification of intersecting pairs from $\mathcal{S} \times \mathcal{T}$, two techniques were investigated and one of these is advocated here. The first method exploits a *multi-level range and segment tree* structures implemented in a state-of-the-art algorithm HYBRID [Zomorodian, et al. (2002)]. This approach proves to be inferior to the considerably simpler *sphere-tree* structure [Hubbard (1996)] (called SPHTREE).

To make use of HYBRID all objects from the sets $\mathcal{S}$ and $\mathcal{T}$ need to be packed into their *axis aligned bounding boxes*. The algorithm proceeds by processing the two lists of boxes (of query spheres and surface triangles) and recursively building segment and range tree structures along each of the coordinate directions [Samet (1990); Zomorodian, et al. (2002)]. Assuming the total geometry entity number $N = n + m$, the gain from the use of recursion is $O(N)$ space utilisation[2], while the runtime complexity is $O\left(N \log^3 (N) + k\right)$, where k is the number of reported box overlaps. This algorithm performs well in many practical cases, although here it cannot show its full strength. Firstly, it is not able to take advantage of the fact that the mesh geometry remains frozen - the tree structures are built partially and stored only temporarily during the recursive processing. A more serious drawback is the fact as the radii of the query spheres increases, it reports many missed sphere-triangle intersections. When query spheres reach a size comparable to the size of the overall domain, the computational overhead is significant. Thus, HYBRID performs well only as long as the size of the query spheres remains comparable to the size of the surface triangles.

The SPHTREE approach is better suited for the current case. At the initial stage of computations, the surface triangulation is wrapped into a sphere-tree hierarchy (Fig. 4), which provides the data structure which is later queried with the sphere pairs. This strategy exploits the fact that the structure remains unchanged during the computations. Here the sphere-tree is built in a simple, top-down manner by the recursive application of a median-plane coordinate bisection along the direction aligned with the longest distribution of surface triangles. At each but the last level of the tree, eight nodes are created. At the last level, nodes store no more than sixty-four leaf spheres bounding mesh triangles. This strategy is sufficient for our purposes, although further research is needed to assess the efficiency of more sophisticated approaches.

The sphere-tree also naturally addresses the issue of finding intersections with a pair of expanding spheres. At each level of the tree, the computations proceed only if

[2] Conventionally $O\left(n \log^3 n\right)$ for a self-intersection test among n boxes [Zomorodian, et al. (2002)].

Algorithm 1 Surface triangles sphere-tree (T) traversal with a pair of query spheres ($\mathbf{P}$, R_{min}, R_{max}).

```
I = query_spheres_traverse ($T$,$\mathbf{P}$,$R_{min}$,$R_{max}$)
1.  $d = \|\mathbf{P} - T.\mathbf{P}\|$
2.  if $d < (R_{max} + T.R) \wedge d >$
                        $(R_{min} - T.R)$ then
3.    if is_node ($T$) then
4.      for each $Q$ in $T.Children$
5.        query_spheres_traverse
                        ($Q$,$\mathbf{P}$,$R_{min}$,$R_{max}$)
6.      endfor
7.    else if $T.Triangle$ intersects
        ($\mathbf{P}$,$R_{min}$,$R_{max}$) /* it's a leaf */
8.      $V$ = vertex_of ($\mathbf{P}$,$R_{min}$,$R_{max}$)
9.      $T^{sub}$ = intersection
                  ($T.Triangle$,$\mathbf{P}$,$R_{min}$)
10.     $T^{add}$ = intersection
                  ($T.Triangle$,$\mathbf{P}$,$R_{max}$)
11.     $I = I \cup (V, T^{sub}, T^{add})$
12.   endif
13. endif
```

the bounding sphere of the current node passes through the volume described by the pair of query spheres. Thus nodes placed inside of the smaller, or outside of the bigger of the query spheres, are easily omitted. Query traversal is fast, as the necessary numerical tests comprise only a few inexpensive operations [Algorithm 1]. The SPHTREE algorithm performs well not only for query spheres with small radii, but also for those of a size comparable to that of the domain itself.

4 Numerical Examples

Two examples are presented in order to show the efficiency of triangle-sphere intersection algorithm and demonstrate the potential of the space-time BEM approach.

4.1 *The triangle-sphere intersection with a hourglass mesh*

Two numerical examples are presented. Although both involve surface meshes of a size still inaccessible to our current boundary element solver, this comparison is useful as a prelude to further research. In both cases the zero order approximation is assumed, so the actual "nodes" are placed at mass centers of the mesh triangles. Consequently the number of query spheres is equal to the number of triangles. In these tests, the radii of the query spheres grow from $\frac{1}{100}L$ up to L, where L is the largest dimension of the mesh bounding box. The difference between the bigger and the smaller of sphere radii is set equal to the minimal distance between the adjacent mesh nodes. Each time, the complete search is performed (including calculation of the sub-triangulations). Although the results apparently depend on the geometry of the input surfaces (Fig. 5, 6), the general performance conforms to theoretical expectations formulated in the Section 3. It is clear that the performance of HYBRID degrades for larger query spheres, while SPHTREE's performance improves. These runs were performed on a 1.7 GHz Pentium PC.

4.2 *Wave propagation in a prismatic rod*

The second problem, shown in Fig. 7, had been used as a benchmark test by several researchers [Mansur and Brebbia (1985), Banerjee and Kobayashi (1992), Marrero and Dominguez (2003)] in the field of time domain BEM. It is essentially one-dimensional in space since it is symmetric in the yz plane and has an analytical solution. Here, we solve it in 4 dimensional space-time with a boundary mesh consisting of 320 constant triangular elements in space, as shown in Fig. 7. As a benchmark, the shape functions for u and P in time are assumed to be constant as well. The time step size is defined by the dimensionless parameter $\beta = c\Delta t/\Delta h$; c is the wave velocity and Δh is equal to the radius of the circle inside the triangle. In order to find the best ratio, values of β in the range vary 0.1 to 1.5 are explored in the numerical experiment. The rod dimensions(m) are $8 \times 4 \times 4$. A Heaviside loading function is applied at the free end, where $x = 0m$. The opposite side (at $x = 8m$) is fixed. The material properties are: wave velocity $c = 200m/s$, Young's modulus $E = 10^5 N/m^2$. In Fig. 7, the displacements on the boundary at various times are shown.

The computed time history of the normal traction at the fixed end is shown in Fig. 8. Results are shown for values of $\beta = 0.3$ up to 1. Stable results were obtained for $\beta >$ 0.3. However, damping increases for larger values of β. The best results for high gradient areas are obtained for $\beta = 0.6 - 0.7$.

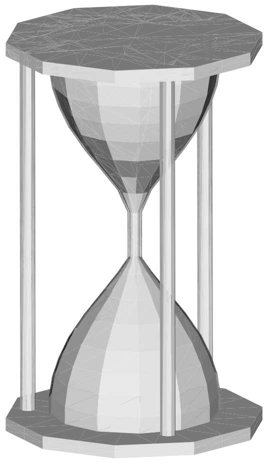
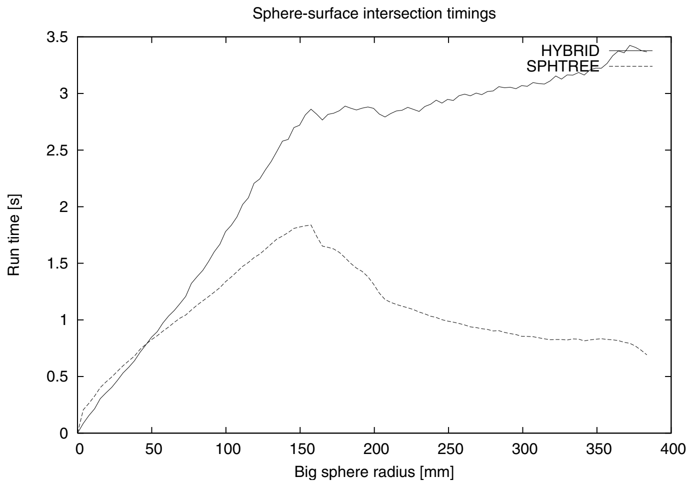

Figure 5 : Sphere-triangle intersection search. Wall-clock timings for a hourglass mesh comprising 1784 triangles.

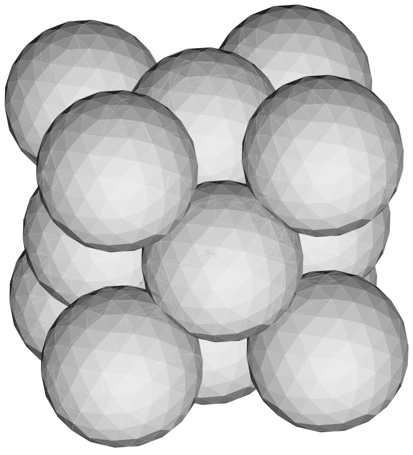
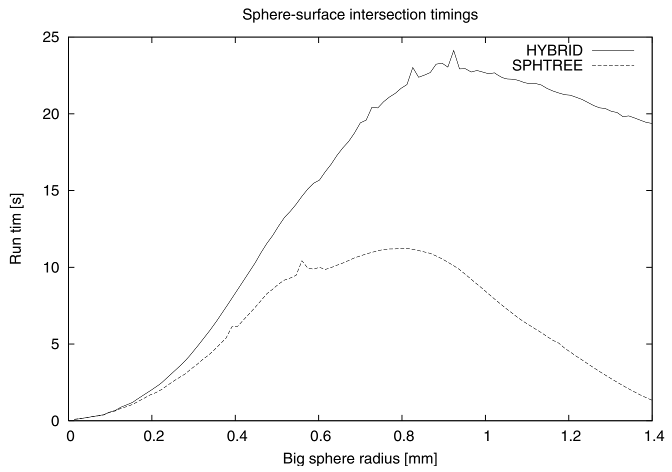

Figure 6 : Sphere-triangle intersection search. Wall-clock timings for a ball set mesh comprising 4480 triangles.

5 Conclusion

For 3D transient acoustic waves or elastodynamic problems, BEM has some advantages over other numerical methods because boundary-only discretisation means that mesh dispersion is less significant. However, the method is not immune to the difficulties of approximating high gradient areas with high accuracy while retaining computational efficiency. In order to achieve the flexibility of approximating high gradient areas in arbitrary locations, we propose a new method which employs shape functions in both space and time domain. Further research in progress focuses on implementing higher order triangular elements, adaptive algorithms and parallelization in 4D space-time acoustic waves and elastodynamics.

Acknowledgement: We would like to thank Professor Nenad Bicanic for discussing this paper and some helpful suggestions, and Professor George Manolis for access to the source code of his elastodynamics BEM.

References

Banerjee, P. K.; Kobayashi, S.(Eds): *Advanced dynamic analysis by boundary element methods*, volume 7, Developments in Boundary Element Methods. Elsevier Applied Science.

Beskos, D. E. (1997): Boundary elements in dynamic analysis, part ii. (1986-1996). *Applied Mechanics Review*, vol. 50, no. 3.

Birgisson, B.; Siebrits, E.; Peirce, A. P. (1999): Elastodynamic direct boundary element methods with en-

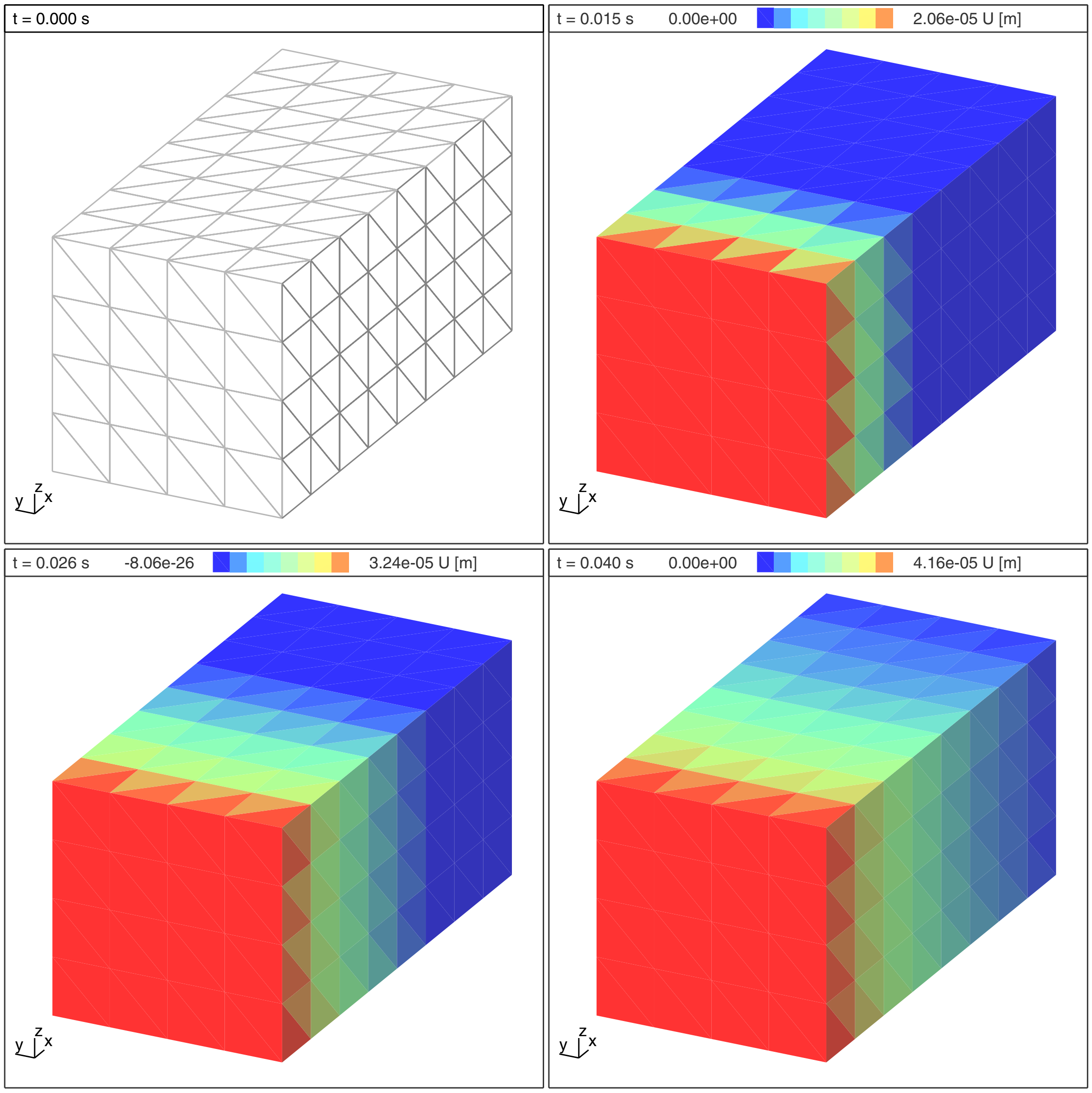

Figure 7 : A prismatic rod represented by 320 elements. Displacements at time t =0.015s, 0.026s, 0.040s

hanced numerical stability properties. *International Journal for Numerical Methods in Engineering*, vol. 46, no. 6, pp. : 871–888.

Callsen, S.; von Estorff, O.; Zaleski, O. (2004): Direct and indirect approach of a desingularized boundary element formulation for acoustical problems. *CMES: Computer Modeling in Engineering & Sciences*, vol. 6, no. 5, pp. 421–429.

Dominguez, J. (1993): *Boundary elements in dynamics*. Computational Mechanics Publications, Southampton.

Frangi, A.; Novati, G. (1999): On the numerical stability of time-domain elastodynamic analyses by BEM. *Computer Methods in Applied Mechanics and Engineering*, vol. 173, no. 3-4, pp. 403–417.

Gao, X.; Davies, T. G. (2002): *Boundary Element Programming in Mechanics*. Cambridge University Press, New York.

Hubbard, P. M. (1996): Approximating polyhedra with spheres for time-critical collision detection. *ACM Trans. Graph.*, vol. 15, no. 3, pp. 179–210.

Lie, S. T.; Yu, G.; Zhao, Z. (2001): Coupling of

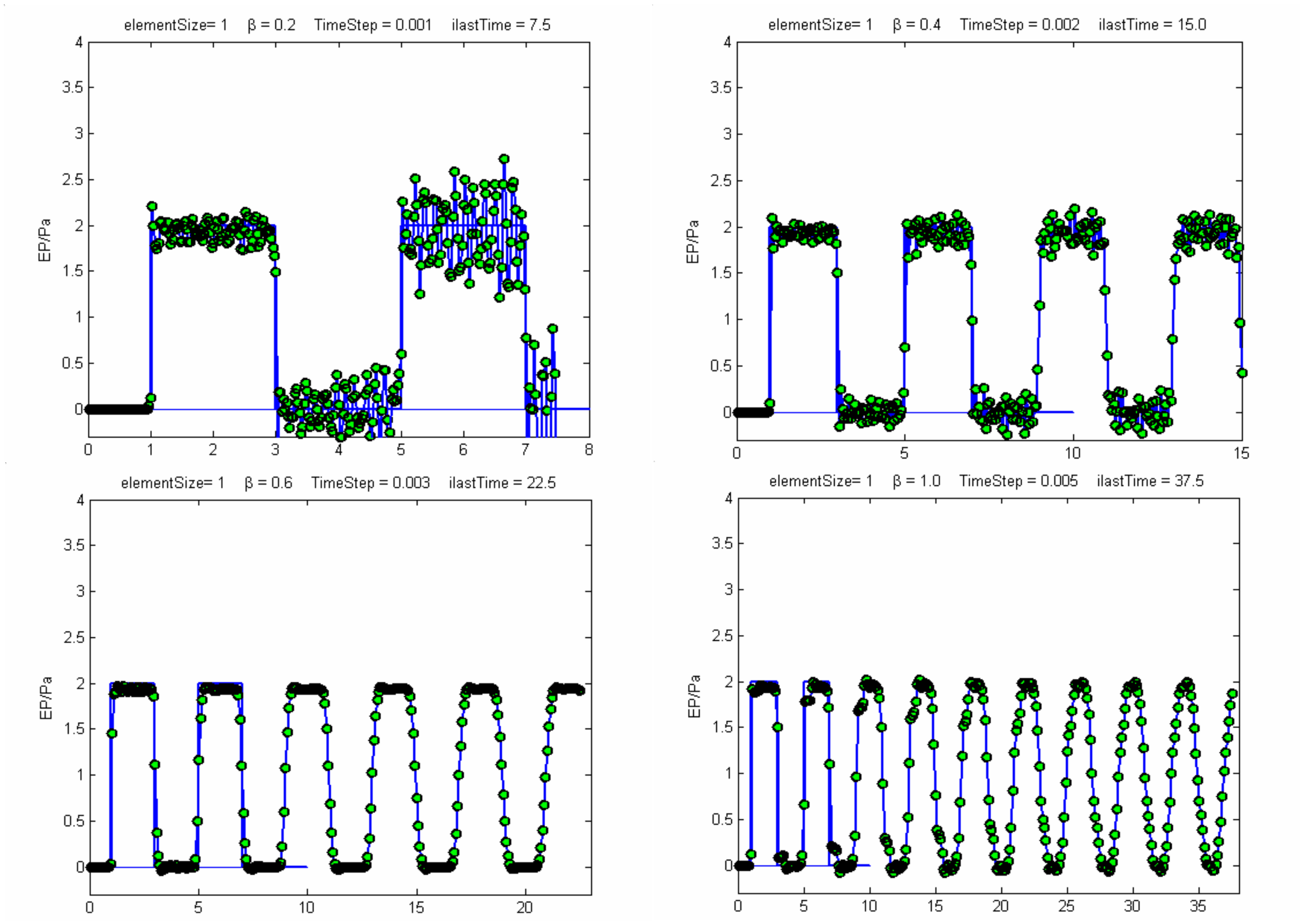

Figure 8 : Time history of traction on the fixed end, $\beta = 0.2, 0.4, 0.6, 1.0$

BEM/FEM for time domain structural-acoustic interaction problems. *CMES: Computer Modeling in Engineering & Sciences*, vol. 2, no. 2, pp. 171–181.

Manolis, G.; Beskos, D. (1988): *Boundary Element Methods in Elastodynamics*. Unwin Hyman, London.

Mansur, W.; Brebbia, C. A. (1985): *Topics in Boundary Element Research*, volume 2: Time-dependent and Vibration Problems. Springer-Verlag, Berlin.

Mansur, W. J.; Carrer, J. (1998): Time discontinuous linear traction approximation in time-domain BEM scalar wave propagation analysis. *International Journal for Numerical Methods in Engineering*, vol. 42, no. 4, pp. 667–683.

Marrero, M.; Dominguez, J. (2003): Numerical behavior of time domain bem for three-dimensional transient elastodynamic problems. *Engineering Analysis with Boundary Elements*, vol. 27, no. 1, pp. 39–48.

Peirce, A.; Siebrits, E. (1997): Stability analysis and design of time-stepping schemes for general elastodynamic boundary element models. *International Journal for Numerical Methods in Engineering*, vol. 40, no. 2, pp. 319 – 342.

Pozrikidis, C. (2002): *A Practical Guide to Boundary Element Methods with the Software Library BEMLIB*. CRC.

Qian, Z. Y.; Han, Z. D.; Ufimtsev, P.; Atluri, S. N. (2004): Non-hyper-singular boundary integral equations for acoustic problems, implemented by the collocation-based boundary element method. *CMES: Computer Modeling in Engineering & Sciences*, vol. 6, no. 2, pp. 133–144.

Rizos, D.; Karabalis, D. (1994): Advanced direct time domain bem formulation for general 3-D elastodynamic problems. *Computational Mechanics*, vol. 15, no. 3, pp. 249–269.

Samet, H. (1990): *The Design and Analysis of Spatial Data Structures*. Addison-Wesley.

Soares, Jr., D.; Mansur, W. J. (2005): An efficient time-domain BEM/FEM coupling for acoustic-elastodynamic interaction problems. *CMES: Computer Modeling in Engineering & Sciences*, vol. 8, no. 2, pp. 153–164.

Wang, C. C.; Wang, H. C. (1996): Two-dimensional elastodynamic transient analysis by QL time-domain BEM formulation. *International Journal for Numerical Methods in Engineering*, vol. 39, no. 6, pp. 951–985.

Yu, G. Y.; Mansur, W. (1998): Linear θ method applied to 2D time-domain BEM. *Communications in Numerical Methods in Engineering*, vol. 14, no. 12, pp. 1171–1179.

Zhang, C.; Savaidis, A. (2003): 3-D transient dynamic crack analysis by a novel time-domain BEM. *CMES: Computer Modeling in Engineering & Sciences*, vol. 4, no. 5, pp. 603–618.

Zomorodian, A.; E' delsbrunner, H.; De Berg, M. (2002): Fast Software for Box Intersections. *International Journal of Computational Geometry and Applications*, vol. 12, pp. 143–173.