Implementation of Hierarchical Tree Structure in Fast Multipole Method in 2-D Seismic Elastic Domain

Mohammad Saffar[1] and Mohsen Kamalian[2*]

1. Ph.D. Candidate, Geotechnical Engineering Research Center, International Institute of Earthquake Engineering and Seismology (IIEES), Tehran, Iran
2. Professor, Geotechnical Engineering Research Center, International Institute of Earthquake Engineering and Seismology (IIEES), Tehran, Iran,
* Corresponding Author; email: kamalian@iiees.ac.ir

Received: 28/04/2017
Accepted: 11/09/2017

ABSTRACT

A numerical boundary element, as an appurtenance of integral equation method, has some useful characteristics that facilitate the solutions of numerical equations, but asymmetrical and sparse structure of formed stiffness matrix in large-scale boundary element method related to high degree of freedom problems make it unpractical, especially in seismic analysis of large-scale surface topographies with irregularities. Nowadays, fast algorithms such as fast multi-pole method present new media in numerical solutions with the aim of revolutionary changes in geometric definitions. In contrary with the usual node-to-node or element-to-element interconnection implementation, the cell-to-cell relation along hierarchy tree structure is applied. In most papers, the fast algorithm uses a two-level hierarchical tree structure as a part of algorithm internally without detail illustration. Therefore, a comprehensive detail of hierarchical tree structure is requested. In this paper, a multi-level (level definition is dynamic) hierarchical tree structure is presented with graphical theme and examples. This paper presents the relation between conventional boundary element method geometric structure with hierarchical tree model, and later, explains the method along with its abilities and limitations.

Keywords:
Hierarchical tree structure; Fast Boundary Element; Large-scale problem; Topographies; Degree of Freedom

1. Introduction

In technical literature, there are two applicable choices based on different volumetric and boundary methods to solve equations numerically. Both of them have their own benefits and limitations. The former solvers have symmetrical coefficient matrix where volumetric discretization definition is mandatory, while the latter solvers impose radiation condition implicitly, which eliminate the need for local absorbing boundary conditions [1]. They need to only mesh exterior surface, while MOT nature schemes of integral equation methods enforcedly are unstable and computationally expensive, as the number of DOF increases remarkably.

These days, BEM with systematic analysis nature has been developed in both frequency and time domain, especially in small strain range and linear behavior of materials. One of the first attempts in this media was established by Friedman and Shaw [2] for anti-plane elastodynamics. Boundary element was formulated in 2-D time domain by Cole et al. [3] for the first time in scalar problems. Niwa et al. [4], Manolis and Beskos [5], and Manolis

[6] used Cole formulation with some changes in analysis of 2-D elastodynamics problems, and their solutions were compared with responses obtained in the transformed domains. The new generation of full-plane time-convoluted kernels and BE formulation was presented by Mansur [7] for 2-D scalar and elastodynamics problems. Antes [8] developed the time-domain BE formulation for arbitrary initial conditions, and Spyrakos and Antes [9] changed last formulation for dynamic analysis of various problems usage. Regardless of the Heaviside functions in integration, Israil and Banerjee [10] were able to show the simpler and more tangible form of full-plane time-convoluted kernels for scalar and elastodynamics problems so that later Kamalian et al. [11] modified their in-plane kernels and implemented that in time-domain BEM algorithm in order to analyze different geotechnical earthquake engineering problems as well.

In the mid-1980s, Rokhlin and Greengard [12] established a new algorithm that reduces the time consumption and complexities of the equation system in integral base equation solvers intentionally. They called it Fast Multi-pole Method (FMM) [13] that presents innovative media in solving boundary element formulation.

This method was developed by Greengard [14] as he established FMM to solve problems with two main elements in structure, consisting of hierarchical tree architecture plus iterative solver. Iterative solvers are available as form of external subroutines based on CG, GMRES, Bi-CGSTAB methods, etc. On the contrary, the hierarchical tree architecture is applied in some papers related to fast algorithm with overall description.

The purpose of this article is two-fold:

1. Description of the tree structure with details and illustration of hierarchical algorithm graphically.
2. Solving some practical examples with detailed explanation.

Hierarchical tree algorithm utilizes in various fields in science and engineering, such as Heat conduction [15], Elasticity [16], Stokes flow [17], Acoustic [18], Astrophysics [19], Molecular dynamics [20], fluid mechanics [21], etc.

2. Methodology and Implementation Issues

The basic integral form of BIEM is mentioned in Equation (1). Traditional boundary integral equations have explained in some reference books in field of numerical methods in details, and it is not necessary to describe them here again.

$$\frac{1}{2}\cdot u_i(x,t)=\int_S (t_i(y,t)*G_{ij}-F_{ij}*u_i(y,t)).dS \quad \perp x,y\in S \tag{1}$$

The sole of fast method consists of two parts. First one is constructed by meaning of batch waves that implies source wave's integration instead of node to node interconnections, and second one is related to solving equation method. In this section, the prior one is explained analytically.

2.1. Source Wave Implementation

Despite conventional BEM that solves integral equations in each time step separately, fast BEM uses source signal as a sequence of waves with defined and constant time length; therefore, the source signal $f(t)$ is divided into z consecutive sub-signals $f_z(t)$, Figures (1) and (2).

$$z=0,1,2,...,z-1 \text{ as } f(t)=\sum_{z=0}^{z-1} f_z(t).$$

Note that the earliest time of arrival of the actual signal in the observer point with temporal signature $f_z(t)$ should meet special situation that is explained below.

Let C_S and C_O be disjoint circular domains with radius of R centered at s, o and the distance between the centers or $|o-s|$ will be denoted by $R_c(>2R)$. Besides, assume that C_S includes a part of S denoted by S_0. In the plane wave expansion for the fundamental solution of Equation (1), there is an expression that includes a non-physical ghost and

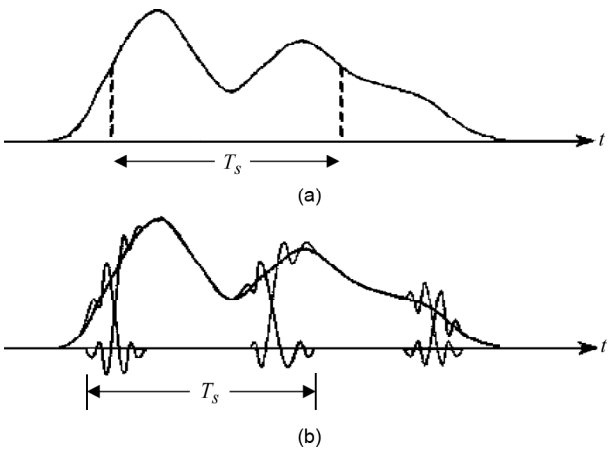

Figure 1. Sectioning of signal into sub-signals of duration T_s.

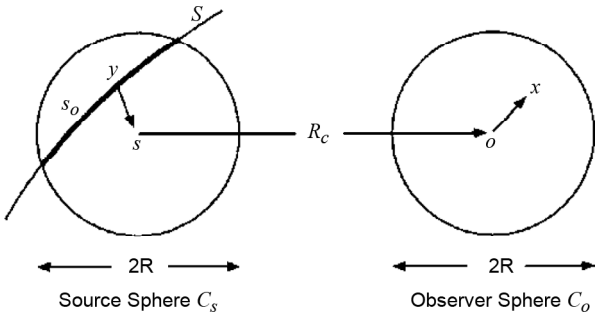

Figure 2. Source and Observer.

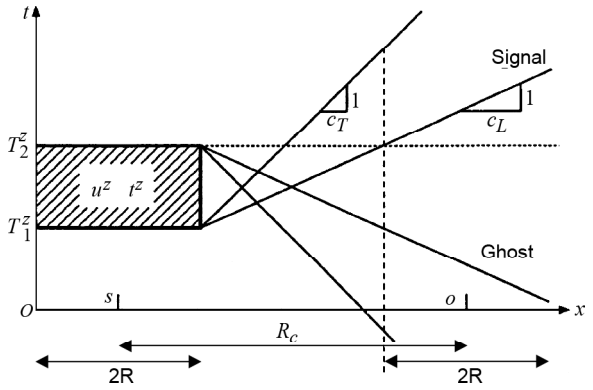

Figure 3. Signal and Ghost.

true sub-signal simultaneously. In utilizing this expansion, it is needed to develop an approach, which guarantees that the ghost does not pollute the solution. In order to obtain such an approach, it is preferable to find density functions u (displacement) and t (traction) as sums of functions u^z and t^z, $f_z(t)$, $z=0,1,2,...,z-1$ which have supports in the finite time intervals $(T_1^z, T_2^z]$.

$$\begin{aligned} &f(x,t)=\sum_{z=0} f_Z(x,t), \\ &f_Z(x,t)= f(x,t), \\ &if \ \left(T_1^z < t \le T_2^z\right) \ or \quad 0 \quad (otherwise) \end{aligned} \tag{2}$$

Here, the time interval $T_2^z - T_1^z$ is equal for all z. At this time, $c_1 > c_2$, If it is taken into consideration that,

$$t_Z^{trans} = \frac{R_c - 2R_s}{c_1} + z(T_2^z - T_1^z) \tag{3}$$

the result would be $T_S = T_2^z - T_1^z$.

Therefore, if $T_Z^{trans} > T_Z^{gost}$, all ghost fields' sub-signals in the observation circle C_O cease to exist before true signal arrives. In addition, $T_Z^{trans} > (z+1)T_S$, all source potentials related to the Z^{th} time interval ends before the true signal reaches any observer (Figure 3).

Now, because of the nature of FMM, the obtained broken series of time-gated sub-signals should be interpolated for each arbitrary time step by use of appropriate interpolation function. There are so much interpolation functions in time and space, and after comparing some of them, finally the following interpolation function (approximate prolate spherical interpolant) is a variant of the one originally proposed by Knab [22].

$$\psi(t)= \frac{\omega_o}{\omega_f} \cdot \frac{Sin(\omega_o t)}{\omega_o t} \cdot \frac{Sin\left(\Omega \cdot p_t \Delta t \sqrt{\left(t/p_t\Delta t\right)^2 - 1}\right)}{Sinh\left(\Omega \cdot p_t \Delta t\right)\sqrt{\left(t/p_t\Delta t\right)^2 - 1}} \tag{4}$$

Here, $\omega_0 = \omega_{\max}\dfrac{(\chi_1+1)}{2}$, $\Omega = \omega_{\max}\dfrac{(\chi_1-1)}{2}$ and p_t, χ_1 are parameters, which p_t is an integer equal or greater than 0 and χ_1 is a real number equal or greater than 1 too. ψ is band limited by $\omega_f = \chi_1 \cdot \omega_{\max}$ and almost vanishes (or $\psi \approx 0$) for $|t| > p_t \cdot \Delta t$.

It is assumed that $f(t)$ is very smooth, or is band limited by $\omega_{\max}$. $f(t)$ is interpolated by using an approximately time and band limited base function ψ and group consecutive M terms together to define:

$$f_Z(t) = \sum_{\alpha=zM+1}^{(z+1)M} f(\alpha\Delta t)\cdot\psi(t-\alpha\Delta t) \tag{5}$$

Then $f(t)$ splits into a sum of approximately time and band limited functions $f_z(t)$ by using M samples of $f(t)$ (Figure 4).

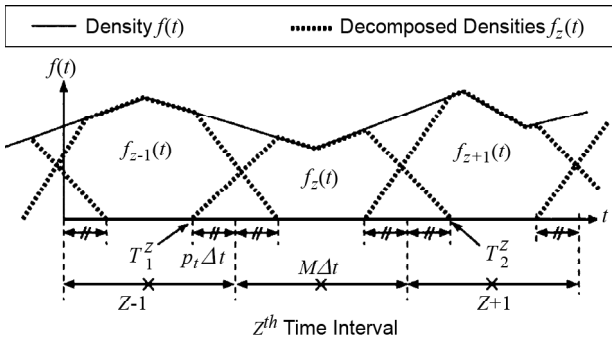

Figure 4. Decomposing of Signal to Sub-Signals.

3. Tree Structure Versus Conventional Mesh Definition

In conventional boundary element, the exterior (boundary) surfaces S is divided into N_S segments. For simplicity, It is assumed that the boundary elements to be piecewise constant. Although this assumption has no effect on problem generality, it places one node on each element that indicates variation in element properties amplitude. Therefore, in case of using constant element, the total number of nodes would be equal to the total number of elements indeed (Figure 5).

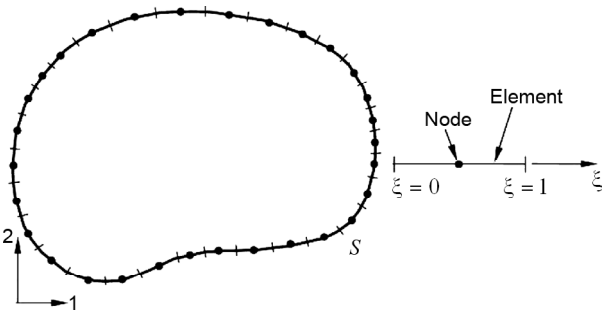

Figure 5. Discretization of boundary S using constant elements.

In the classical boundary integral elements equation, direct integrations are needed. It should be noted that the node-to-node (or element-to-element) interaction between each assumed source and field point is indispensable. On the contrary, in FMM, an innovative scenario is happened/exists that all node-to-node interconnections are replaced with cell-to-cell interactions by a hierarchical tree architecture of cells containing groups of elements.

4. Geometry of Hierarchical Tree in 2-D

First, we discretize the exterior surface of problem S into N_S elements such as conventional boundary element. Then, the geometric problem is circumscribed by a square that covers the entire boundary of problem, and we call this square is called the "cell" (a parent cell) of level zero. This cell is divided into four equal sub-square cells (child cells) of level one, whose edge length is half of the parent cell. This process should be continued as such until a cell contains elements less than a prescribed number (only for illustration, in Figure (6), this number is supposed to be 1). A cell having childless is called a leaf; thus, a quad-tree algorithm will be formed when this procedure is completed (Figure 6).

In this process, an element is assumed within a cell, if and only if, the center of the element is inside that cell. This assumption is independent of element rank accuracy [23].

It is considered that the edge length of a cell at level l is given by $L/2^l$, where L is the length of the edge of the largest cell at level zero [24].

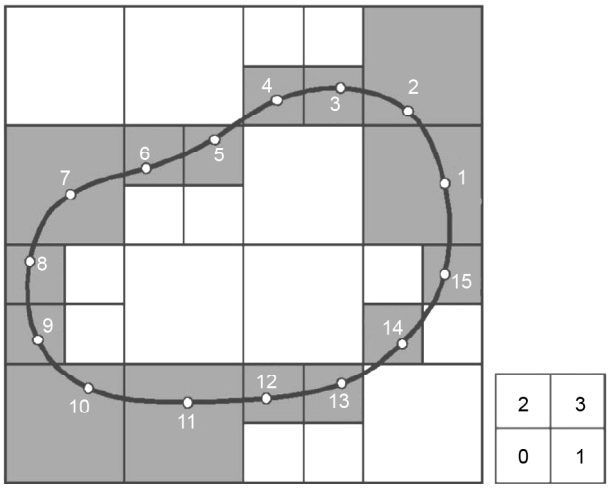

Figure 6. Demonstration of hierarchical tree (cells versus en closing elements).

5. Hierarchical Tree Structure Implementation

The main parent cell at level zero (largest cell that surrounds the exterior surface of problem) is called "cell one", and the four cells at lower level (level one) are numbered as 2, 3, 4, and 5, subsequently, according to arranged numbers from 0 to 3 at the right side of Figure (6).

Sorting of elements in the cells related to their own levels is shown in Figure (7), where squares represent legend numbers that indicate the position of each filled cell (symbols 0 to 3 in Figure 6), and the element numbers are surrounded by circles.

There are 15 elements in Figure (6) with one node at each center. According to Table (1), elements are sorted in each level by dividing the cells from upper levels to lower ones based on the legend of Figure (6) subsequently.

For some special problems with notable differences between length and width size of geometries, there are new advanced modified algorithm, such as adaptive structure. This algorithm can accelerate the speed of BEM equation solution dramatically [25]. Designing of quad-tree structure needs some new arrays to determine the position of each cell and

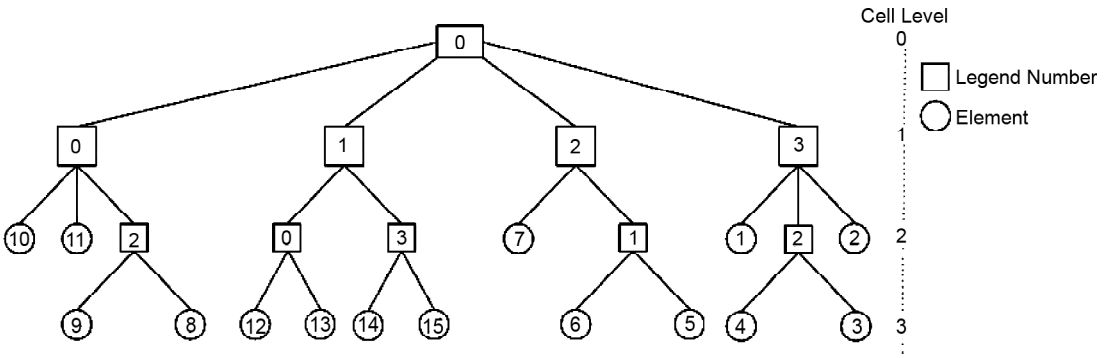

Figure 7. Demonstration of hierarchical quad-tree structure in 2-D.

Table 1. Element sequences sorting accordance with tree level.

Tree Level	Sequences of the Elements in Quad-Tree Algorithm														
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	8	9	10	11	12	13	14	15	5	6	7	1	2	3	4
2	10	11	8	9	12	13	14	15	7	6	5	1	4	3	2
3	10	11	8	9	12	13	14	15	7	6	5	1	4	3	2

element in the problem explained below:

- Array Loc_tree (i) gives i^{th} cell location within its corresponding tree level. The parent main cell (cell 0) is divided by . The numbering of each small square (cells in lower level) starts from the lower left corner, first in the *x* direction, then in the *y* direction.
- Array Elem_num (i) gives the number of elements surrounded by i^{th} cell area.
- Array cell_fath (i) gives the parent cell number of ith cell.
- Array Start_elem (i) indicates the starting place of elements included in the i^{th} cell [26].

Now, we state three numerical examples with more elements and detailed description.

6. Numerical Examples

Example One:

This simple example only tries to illustrate fast method in straight forward manner with no practical usage. The first model consists of 20 constant elements, which are located on perimeter of the semi-rectangle. All first and end points of the element's coordinates should be defined as conventional boundary elements, listed in Table (2).

Red nodes indicate the position of the center of elements, where black nodes specify the first and end points of the elements (Figure 4). It is supposed that the predefined maximum number of elements in a leaf is limited to one, it means that the process of bi-sectioning of cells is continued until the cells in the deepest level would have only one element, as shown in Table (3), and the maximum number of levels is confined to 4.

Table 2. Middle element nodes coordinate of first model.

Node Number	X Coordinate	Y Coordinate
1	0.1	0.0
2	0.3	0.0
3	0.5	0.0
4	0.7	0.0
5	0.9	0.0
6	1.0	0.1
7	1.0	0.3
8	1.0	0.5
9	1.0	0.7
10	1.0	0.9
11	0.9	1.0
12	0.7	1.0
13	0.5	1.0
14	0.3	1.0
15	0.1	1.0
16	0.0	0.9
17	0.0	0.7
18	0.0	0.5
19	0.0	0.3
20	0.0	0.1

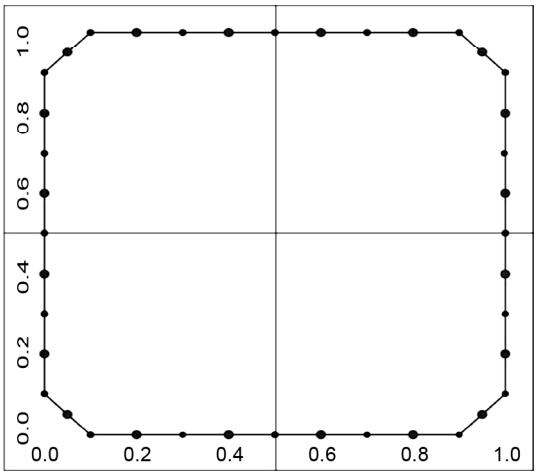

Figure 8. Demonstration of first model of hierarchical quad-tree structure in 2-D.

Table 3. Values of the arrays of first model in hierarchy tree structure.

Cell_No.	Loc_tree	Elem_num	Cell_fath	Start_elem
1	0	20	1	1
2	0	4	1	3
3	1	5	1	18
4	2	5	1	8
5	3	6	1	1
6	0	2	2	2
7	1	1	2	19
8	4	1	2	3
9	2	2	3	5
10	3	2	3	7
11	7	1	3	18
12	8	2	4	16
13	12	2	4	14
14	13	1	4	8
15	11	2	5	13
16	14	2	5	10
17	15	2	5	1
18	0	2	6	3
19	4	1	9	4
20	5	1	9	5
21	7	2	10	18
22	32	1	12	17
23	40	1	12	16
24	56	2	13	8
25	39	1	15	9
26	47	1	15	13
27	60	1	16	12
28	61	1	16	10
29	63	2	17	1
30	1	1	18	20
31	16	1	18	5
32	14	1	21	6
33	31	1	21	16
34	224	1	24	15
35	241	1	24	10
36	239	1	29	11
37	254	1	29	20

Predefined arrays (Cell_no(i) , Loc_tree(i) , Elem_num(i) , cell_fath(i), Start_elem(i)) are calculated and listed in Table (3).

For instance, cell_No. 1 surrounds the exterior boundary of the model. Therefore, the cell number is one, and its position in level 0 is zero (Loc_tree=0). For first level (level 0), it is supposed that the father's cell is also one (Cell_fath=1), and all elements are included (Elem_num=20), which starts with element number one (start_elem=1).

The process is continued to the second level (level 2). This level consists of four filled cells that are numbered from 2 to 5. The positions of these four cells are sorted by using legend, which is depicted at the left side of Figure (2) (Loc_num). Father or parent's cell is one (Cell_fath=1), and the first element located in each second level of the cell is numbered as 3, 18, 8, and 1, sequentially The bi-sectioning of each level is continued until the number of elements in the deepest level is confined to one, as mentioned in Table (3).

Example Two:

The second model is depicted in Figure (9) with 2000 constant elements, which are located on perimeter of the circle. For simplicity, the middle points of elements are shown with red nodes, and *X*, *Y*coordinates are listed in Table (4).

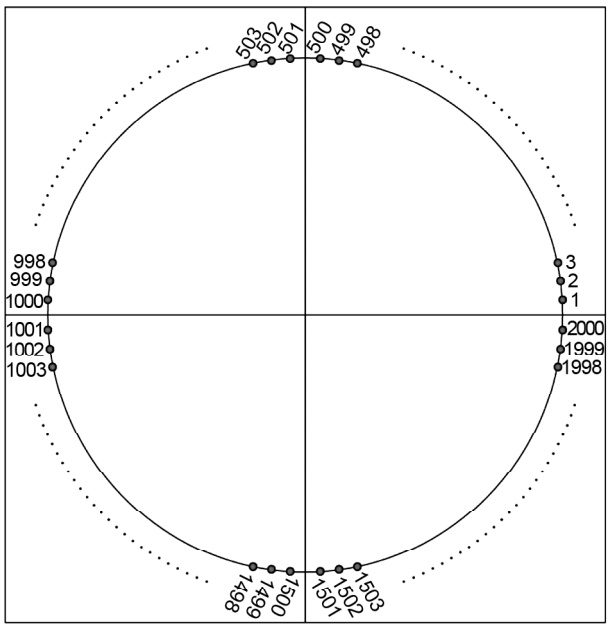

Figure 9. Demonstration of second model of hierarchical quad-tree structure in 2-D.

Table 4. Middle element nodes coordinate of second model.

Element Number	X Coordinate	Y Coordinate
1	100.00	0.16
2	100.00	0.47
3	100.00	0.79
.	.	.
.	.	.
.	.	.
498	0.79	100.00
499	0.47	100.00
500	0.16	100.00
501	-0.16	100.00
502	-0.47	100.00
503	-0.79	100.00
.	.	.
.	.	.
.	.	.
998	-100.00	0.79
999	-100.00	0.47
1000	-100.00	0.16
1001	-100.00	-0.16
1002	-100.00	-0.47
1003	-100.00	-0.79
.	.	.
.	.	.
.	.	.
1498	-0.79	-100.00
1499	-0.47	-100.00
1500	-0.16	-100.00
1501	0.16	-100.00
1502	0.47	-100.00
1503	0.79	-100.00
.	.	.
.	.	.
.	.	.
1998	100.00	-0.79
1999	100.00	-0.47
2000	100.00	-0.16

Predefined arrays such as Cell_no(i), Loc_tree(i), Elem_num(i), etc. are calculated and listed in Table (5), where the maximum number of predefined elements in the deepest level (leaf cells) is equal to 40, and the maximum level number is considered as 5. Maximum number of cells in this example is 181, and the position of those filled cells in Cartesian coordinate system is 890.

Example Three:

The Third model is displayed in Figure (10) with 308 constant elements, which are located on perimeter of the semi-four leaf flower. For simplicity, the first node is located at start point of trigonometry circle on X axes, and other nodes are arranged counter clockwise. X, Y coordinates of nodes are listed in Table (6).

Table 5. Values of the arrays of Second model in hierarchy tree structure.

Cell_no.	Loc_tree	Elem_num	Cell_fath	Start_elem
1	0	2000	1	1
2	0	500	1	1
3	1	500	1	501
4	2	500	1	1001
5	3	500	1	1501
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
30	23	91	11	825
31	31	85	11	916
32	32	85	12	1001
33	40	91	12	1086
34	48	35	13	1177
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
60	27	47	26	630
61	44	17	28	712
62	45	44	28	729
63	61	17	28	773
64	78	47	30	825
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
100	45	7	51	276
101	14	21	52	283
102	15	21	52	304
103	258	23	53	325
104	259	1	53	348
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
175	950	23	85	1763
176	951	23	85	1786
177	980	21	86	1809
178	981	23	86	1830
179	858	14	89	1887
180	859	15	89	1901
181	890	15	89	1916

Predefined arrays such as Cell_no(i), Loc_tree(i), Elem_num(i), etc. are calculated and listed in Table (7), where the maximum number of predefined elements in the deepest level (leaf cells) is equal to 40, and the maximum level number is considered as 5. Maximum number of cells in this example is 21, and the position of those filled cells in Cartesian coordinate system is 15.

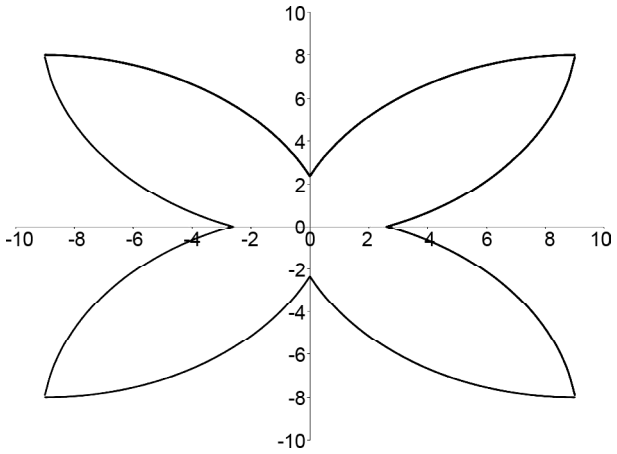

Figure 10. Demonstration of third model of hierarchical quad-tree structure in 2-D.

Table 6. Element nodes coordinate of third model.

Node Number	X Coordinate	Y Coordinate
1	2.60	0.00
2	2.80	0.09
3	3.00	0.17
.	.	.
.	.	.
.	.	.
74	0.80	3.72
75	0.60	3.43
76	0.40	3.10
.	.	.
.	.	.
.	.	.
144	-4.80	1.16
145	-4.60	1.02
146	-4.40	0.89
.	.	.
.	.	.
.	.	.
214	-3.60	-6.42
215	-3.40	-6.28
216	-3.20	-6.15
.	.	.
.	.	.
.	.	.
306	3.20	-0.25
307	3.00	-0.17
308	2.80	-0.09

Table 7. Values of the arrays of third model in hierarchy tree structure.

Cell_no.	Loc_tree	Elem_num	Cell_fath	Start_elem
1	0	308	1	1
2	0	77	1	1
3	1	77	1	78
4	2	77	1	155
5	3	77	1	232
6	0	27	2	1
7	1	15	2	28
8	4	17	2	43
9	5	18	2	60
10	2	15	3	78
11	3	27	3	93
12	6	18	3	120
13	7	17	3	138
14	8	17	4	155
15	9	18	4	172
16	12	27	4	190
17	13	15	4	217
18	10	18	5	232
19	11	17	5	250
20	14	15	5	267
21	15	27	5	282

7. Conclusion

The essential part of new revolutionary changes in traditional BEM is due to elements interconnections. In fast method, ones should integrate batch numbers of elements or nodes to establish new colony. New interconnections is provided by cell to cell relation in upper levels. This process continues until the potential of each tiny part of cells (elements) will be calculated.

Hierarchical tree structure is a new element discretization with more complexity in contrast with the conventional boundary element method. Therefore, for middle range DOF problem with less than 1000 elements, the conventional method is still more effective than the new ones.

In contrary, modeling of large-scale surface

topographies with irregularities causes increased DOF as well as the rank of sparse and asymmetric stiffness matrices, which makes the conventional method completely impractical.

Hierarchical structure, by using some predefined arrays with new discretization as well as iterative solvers, accelerates the speed of matrix equation solutions, especially when the DOF increases dramatically.

References

1. Kamalian, M., Gatmiri, B., Sohrabi-Bidar, A., and Khalaj, A. (2007) Amplification pattern of 2D semi-sine shaped valleys subjected to vertically propagating incident waves. *Commun. Numer. Methods Eng.*, **23**(10), 871-887.

2. Friedman, M.B. and Shaw, R. (1962) Diffraction of pulses by cylindrical obstacles of arbitrary cross section. *J. Appl. Mech.*, **29**(1), 40-46.

3. Cole, D.N., Kosloff, D.D., and Minster, J.B. (1978) A numerical boundary integral equation method for elastodynamics. *Bulletin of the Seismological Society of America*, **68**(5), 1331-1357.

4. Niwa, Y., Fukui, T., Kato, S., and Fujiki, K. (1980) An application of the integral equation method to two-dimensional elastodynamics. *Theor. Appl. Mech.*, **28**, 281-290.

5. Manolis, G.D. and Beskos, D.E. (1981) Dynamic stress concentration studies by boundary integrals and Laplace transforms. *Int. J. Numer. Methods Eng.*, **17**(4), 573-599.

6. Manolis, G.D. (1983) A comparative study on three boundary element method approaches to problems in elastodynamics. *Int. J. Numer. Methods Eng.*, **19**(1), 73-91.

7. Mansur, W.J. (1983) *A Time-Stepping Technique to Solve Wave Propagation Problems Using the Boundary Element Method.* Ph.D. Dissertation, University of Southampton.

8. Antes, H. (1985) A boundary elements procedure for transient wave propagation in two-dimensional isotropic elastic media. *Finite Elem. Anal. Des.*, **1**(4), 313-322.

9. Spyrakos, C.C. and Antes, H. (1986) Time domain boundary element method approaches in elastodynamics: a comparative study. *Comput. Struct.*, **24**(4), 529-535.

10. Israil, A.S.M. and Banerjee, P.K. (1990b) Advanced time-domain formulation of BEM for two-dimensional transient elastodynamics. *Int. J. Numer. Methods Eng.*, **29**(7), 1421-1440.

11. Kamalian, M., Gatmiri, B., and Sohrabi-Bidar, A. (2003) On time-domain two dimensional site response analysis of topographic structures by BEM. *J. Seism. Earthq. Eng.*, **5**(2), 35-45.

12. Greengard, L.F. and Rokhlin, V. (1987) A fast algorithm for particle simulations. *J. Comput. Phys.*, **73**, 325-348.

13. Darve, E. (2000) The fast multipole method: Numerical Implementation. *J. of computational Physics*, **160**, 195-240.

14. Greenbaum, A., Greengard, L., and McFadden, GB. (1993) Laplace's equation and Dirichlet-Neumann map in multiply connected domains. *J. Comput. Phys.*, **105**, 267-278.

15. Greengard, L. (1988) *The Rapid Evaluation of Potential Fields in Particle Systems*. MIT Press, Cambridge, MA.

16. Greengard L. and Helsing J. (1988) On the numerical evaluation of elastostatic fields in locally isotropic two-dimensional composits. *J. Mech. Phys.*, **46**, 1441-1462.

17. Greengard, L. and Kropinski, M.C. (1996) Integral equation methods for stokes flow and isotropic elasticity in the plane. *J. Comput. Phys.*, **125**, 403-414.

18. Ergin A. Michielssen E. Shanker B. (1999) Fast transient analysis of acoustic wave scattering from rigid bodies using a two-level plane wave time domain algorithm. *J. Acoust. Soc. Am.*, **106**, 2405-2416.

19. Warren, M.S. and Salmon, J.K. (1992) Astro-physical N-body simulations using hierarchical tree data structures. *Supercomputing*, **92**, 570-576.

20. Board, J.A., Causey, J.W., Leathrum, J.F.,

Windemuth, A., and Schulten, K. (1992) Accelerated molecular dynamics simulation with the parallel fast multipole method. *Chem. Phys. Lett.*, **198**, 89-94.

21. Salmon, J.K., Warren, M.S., and Winckelmans, G.S. (1994) Fast parallel tree codes for gravitational and fluid dynamical N-body problems. *Int. J. Supercomput. Appl.*, **8**, 124-142.

22. Takahashi, T., Nishimura, N., and Kobayashi, S. (2001) Fast Boundary Integral Equation method for Elastodynamic Problems in 2D in Time Domain. *Trans. JSME (A)*, **661**(67), 1409-1416.

23. Otani, Y., Takahashi, T., and Nishimura, N. (2003) A fast boundary integral equation method for elastodynamics in time domain and its parallelisation. *J. American Society of Mechanical Engineers*, **55**(4), 161-185.

24. Nishimura, N. (2002) Fast multipole accelerated boundary integral equation methods. *J. American Society of Mechanical Engineers*, **55**(4), 299-324.

25. Shen, L. and Liu Y.J. (2007) An adaptive fast multipole boundary element method for three-dimensional potential problems. *Comput. Mech.*, **39**, 681-691.

26. Liu, Y.L. and Nishimura N. (2006) The fast multipole boundary element method for potential problems: A tutorial. *Eng. Anal. Boundary Elements*, **30**, 371-381.