Fast Solution of 3D Elastodynamic Boundary Element Problems by Hierarchical Matrices

I. Benedetti[1], M.H. Aliabadi[2]

[1] On leave from DISAG – Dipartimento di Ingegneria Strutturale Aerospaziale e Geotecnica, Università di Palermo, Viale delle Scienze, Edificio 8, 90128, Palermo, Italy, i.benedetti@unipa.it

[2] Department of Aeronautics, Imperial College London, South Kensington Campus, Roderic Hill Building, Exhibition Road, SW72AZ, London, UK, m.h.aliabadi@imperial.ac.uk

Keywords: Elastodynamic BEM, Laplace Transform Method, Adaptive Cross Approximation, Hierarchical Matrices, Fast BEM solvers

Abstract. In this paper a fast solver for three-dimensional elastodynamic BEM problems formulated in the Laplace transform domain is presented, implemented and tested. The technique is based on the use of hierarchical matrices for the representation of the collocation matrix for each value of the Laplace parameter of interest and uses a preconditioned GMRES for the solution of the algebraic system of equations. The preconditioner is built exploiting the hierarchical arithmetic and taking full advantage of the hierarchical format. An original strategy for speeding up the overall analysis is presented and tested. The reported numerical results demonstrate the effectiveness of the technique.

Introduction

The analysis of elastic dynamic problems through reliable numerical techniques is a subject of great relevance in many fields of science and engineering. The Boundary Element Method (BEM) has been effectively employed for the analysis of dynamic problems using several different strategies, like the time domain formulation, the Fourier and Laplace transform techniques or the dual reciprocity method [1-3].

In the context of Fracture Mechanics, to mention a field of interest to the authors, dynamic crack problems have been successfully solved using the Dual Boundary Element Method in the time domain [4] (Time Domain Method, TDM), in the Laplace transform domain [5,6] (Laplace Transform Method, LTM) and in conjunction with the dual reciprocity method [7] (Dual Reciprocity Method, DRM) and the performances of the three approaches have been compared, in terms of analysis time and memory consumption, for both two-dimensional [8] and three-dimensional [9] cases.

In these works it was found that the Laplace Transform Method, although very accurate, is computationally expensive, in terms of computational time, in comparison to the other techniques. Moreover, although the storage memory required for the analysis in the transform domain is less than that required by the other strategies for dynamics, it is however larger than that needed by the corresponding static problem. On the other hand, it is well known that the BEM produces fully populated matrices whose storage and direct solution are of order $O(n^2)$ and $O(n^3)$ respectively, if n is the order of the problem. Such considerations limit the size of the problems that can be effectively tackled on common computers using the standard BEM. This circumstance hindered for many years the industrial development of the method and has limited its use to the analysis of small or medium size problems.

However, in the recent years, a considerable effort has been devoted to the development of strategies aimed at reducing the computational complexities of the BEM, reducing both memory requirements and time consumption.

Many investigations have been carried out to overcome such limitations and different techniques have been developed such as the fast multipole method (FMM) [10,11], the panel clustering method [12], the mosaic-skeleton approximation [13] and the methods based on the use of hierarchical matrices [14]. The general aim of such techniques is to reduce the computational complexity of the matrix-vector multiplication which is the core operation in iterative solvers for linear systems. However while FMMs and panel clustering tackle the problem from an analytical point of view and require the knowledge of some kernel expansion in advance to carry out the integration, mosaic-skeleton approximations and hierarchical matrices provide purely algebraic tools for the approximation of the boundary element matrices, thus proving particularly suitable for problems where analytic closed form expressions of the kernels are not available or difficult to expand.

The analysis of elastodynamic problems through FMM BEM has been addressed both in the time[15] and frequency domain [16-18], with a special attention to seismology and soil-structure interaction problems. Although the reported results show a noticeable reduction in both memory and time requirements, the implementation of FM strategies requires heavy and *ad hoc* recoding of available packages. On the other hand, fewer works have been devoted to the use of hierarchical matrices for the analysis of elastodynamic problems. In particular, the authors are aware of only one application of the Adaptive Cross Approximation (ACA) to a symmetric elastodynamic Galerkin boundary element formulation [19].

In this work the use of hierarchical matrices for the rapid solution of 3D BEM elastodynamic problems in the Laplace transform domain is presented and investigated for the first time, extending the work previously developed by the authors for the fast solution of 3D static Dual BEM problems [20,21]. To obtain an accurate solution through any inverse transform technique, the solution in the transform domain for a sufficient number of Laplace parameters has to be computed. The hierarchical format is used for representing and storing the collocation matrix for each value of the Laplace parameter. The coefficient matrices are built by Adaptive Cross Approximation (ACA) and the final system for each parameter is solved through a preconditioned GMRES iterative solver, in which the matrix-vector product is sped up by the hierarchical representation itself. Also the preconditioner is built taking advantage of the hierarchical format and an original strategy to further speed up the overall analysis is presented an tested.

Elastodynamic BEM in the Laplace transform domain

The boundary integral equation governing the dynamic behavior of an elastic body in the Laplace transform domain can be written

$$c_{ij}(\boldsymbol{x}_0)\tilde{u}_j(\boldsymbol{x}_0) + \fint_\Gamma \tilde{T}_{ij}(\boldsymbol{x}_0,\boldsymbol{x},s)\tilde{u}_j(\boldsymbol{x})d\Gamma = \int_\Gamma \tilde{U}_{ij}(\boldsymbol{x}_0,\boldsymbol{x},s)\tilde{t}_j(\boldsymbol{x})d\Gamma \tag{1}$$

where the tilde indicates transformed quantities and s is the Laplace parameter. The boundary integral representation of the elastodynamic problem in the Laplace domain has the same form as that of the elastostatic problem. Eq.(1) is to be used in conjunction with the transformed boundary conditions to solve any specific problem.

The form of the elastodynamic fundamental solutions in the Laplace domain allows to write each of them as the sum of two contributions: the first term does not depend on s and contains the same singularities as those present in the elastostatic 3D fundamental solutions; the second term depends on s, but contain only weak singularities. This circumstance leads, after the classical boundary elements discretization procedure, to a linear system of the form

$$\left[\mathbf{A} + \tilde{\mathbf{A}}(s)\right]\tilde{\mathbf{x}}(s) = \tilde{\mathbf{y}}(s) \tag{2}$$

where $\mathbf{A}$ is the matrix stemming from the integration of the terms containing the singularities and needs to be computed only once in advance, while $\tilde{\mathbf{A}}(s)$ stems from the integration of the terms depending on s and has to be computed for each value of the Laplace parameter.

To analyze a general elastodynamic problem by using the Laplace transform technique, one has generally to compute the solution of the system (2) for a set of Laplace parameters s_k, with $k = 1,\ldots,L$, in order to calculate the time-dependent values of any relevant variable by means of some Laplace inverse transformation technique. Wen et al. [6] obtained for example accurate results for long durations in the time domain by using

$$s_k = \sigma + \frac{2\pi k i}{T} \qquad k = 0,\ldots,25 \tag{3}$$

with $\sigma T = 5$ and $T/t_0 = 20$, where t_0 is the unit time. In this work, the solution for the previous set of Laplace points will be computed for some elastodynamic problems and the performance of the hierarchical BEM in terms of memory and time requirements will be compared to that of the standard BEM.

Hierarchical matrices for elastodynamic BEM in the Laplace domain

To improve both storage memory and time required by the elastodynamic BEM analysis in the Laplace domain, system (2) is represented in *hierarchical format* for each value of the Laplace parameter.

The hierarchical or *low rank* representation of a BEM matrix is built by generating the matrix itself as a collection of sub blocks, some of which admit a special *approximated* and *compressed* format. Such blocks, referred to as *low rank blocks*, can be stored in the form

$$\mathbf{B} \cong \mathbf{B}_k = \sum_{i=1}^{k} \mathbf{u}_i \cdot \mathbf{v}_i^T = \mathbf{U} \cdot \mathbf{V}^T \tag{4}$$

The block **B** of order $m \times n$ is approximately generated through the product of **U,** of order $m \times k$, and $\mathbf{V}^T$, of order $k \times n$. If *k*, i.e. the *rank* of the block, is *low* then the representation (4) allows to reduce both memory storage and the computational cost of the matrix-vector multiplication, which is the bottleneck of any iterative solver.

The approximation of the low rank blocks (4) is built by computing only *some* of the entries of the original blocks through adaptive algorithms known as Adaptive Cross Approximation (ACA) [22,23], that allow to reach an initially selected accuracy ε. Low rank blocks represent the numerical interaction, through *asymptotic smooth* kernels, between sets of collocation points and clusters of integration elements which are sufficiently far apart from each other. The distance between clusters of elements enters a certain admissibility condition of the form

$$min\left(diam\ \Omega_{coll}, diam\ \Omega_{int}\right) \leq \eta\ dist(\Omega_{coll}, \Omega_{int}) \tag{5}$$

where Ω_{coll} *and* Ω_{int} are clusters of elements and $\eta > 0$ is a parameter influencing the number of admissible blocks on one hand and the convergence speed of the adaptive approximation of the low rank blocks on the other hand [24]. The blocks that do not satisfy such condition are called *full rank blocks* and they need to be computed and stored entirely, without approximation. Once low and full rank blocks have been generated, some recompression techniques can be used to further reduce the storage memory and computational complexity of the single blocks and of the overall hierarchical matrix (reduced SVD [25] and *coarsening* [26]).

As an almost optimal representation is obtained, the solution of the system can be tackled either directly, through hierarchical matrix inversion [27], or indirectly, through iterative methods [28]. In both cases, the efficiency of the solution relies on the use of a special arithmetic, i.e. a set of algorithms that implement the operations on matrices represented in hierarchical format, such as addition, matrix-vector multiplication, matrix-matrix multiplication, inversion and hierarchical LU decomposition. A collection of algorithms that implement many of such operations is given in [24] while the hierarchical LU decomposition is discussed in [28].

The use of iterative methods takes full advantages of the hierarchical representation, exploiting the efficiency of the low-rank matrix-vector multiplication. The convergence of iterative solvers can be improved by using suitable preconditioners. In this work a hierarchical LU preconditioner is built starting from a coarse approximation of accuracy ε_p of the collocation matrix. An iterative GMRES algorithm is eventually used in conjunction with such preconditioner for solving the system for each value of the Laplace parameter of interest.

Selecting the accuracy ε_c for the collocation matrix, the hierarchical counterpart of system (2) is written

$$\left[\mathscr{A}\left(\varepsilon_c\right) + \tilde{\mathscr{A}}\left(s_k, \varepsilon_c\right)\right]\tilde{\mathbf{x}}\left(s_k\right) = \tilde{\mathbf{y}}\left(s_k\right) \tag{6}$$

The LU preconditioner $\mathbf{P}(s_k)$ is generated in hierarchical format as well, selecting a reduced accuracy ε_p. The preconditioned final system has the form

$$\mathscr{P}(s_k,\varepsilon_p)\left[\mathscr{A}(\varepsilon_c)+\tilde{\mathscr{A}}(s_k,\varepsilon_c)\right]\tilde{\mathbf{x}}(s_k)=\mathscr{P}(s_k,\varepsilon_p)\tilde{\mathbf{y}}(s_k) \tag{7}$$

Even using the hierarchical format, the setup of a preconditioner is an expensive procedure. In the previous scheme a new preconditioner should be built for each value s_k of the Laplace parameter. To further speed up the overall dynamic analysis, an idea is put forward here and it is validated through numerical investigation. The preconditioner set up by using the hierarchical format can be thought of as a coarse approximation of the inverse of the system matrix for a given value s_k. If two subsequent values s_k and s_{k+j} of the Laplace parameter are *close* to each other, the preconditioner set up for s_k could constitute an approximation of the inverse of the system matrix set for s_{k+j}. This idea configures the use of *local preconditioners*, in the sense introduced above, and will be investigated in the next section.

Numerical experiments

Let us consider the bar depicted in Fig. 1, subjected to impact traction load $\sigma(t)=10\cdot\delta(t)$ on the (grey) superior base and constrained in correspondence of the thick points of the inferior base. The bar has square cross section $a\times a$, with $a=1$ and height $h=4$. A representative material with $E=1000$ and $\nu=0.3$ has been considered (all the quantities are non-dimensionalized). To test the performance of the hierarchical solver at varying mesh sizes, three different meshes have been considered, as shown in Table 1.

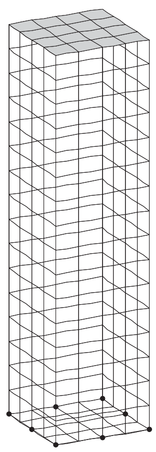

Figure 1 Analyzed configuration.

	Elements	*Nodes*
Mesh 1	288	866
Mesh 2	450	1352
Mesh 3	648	1946

Table 1 Analyzed meshes.

The parameters for the hierarchical analysis have been set to the following values: cardinality of the leaf $C_{leaf}=36$, admissibility parameter $\eta=3$, collocation matrix accuracy $\varepsilon_c=10^{-5}$, preconditioner accuracy $\varepsilon_p=10^{-1}$, GMRES tolerance $\varepsilon_{GMRES}=10^{-6}$.

The preset required accuracy has been obtained for all the preformed computations, confirming the effectiveness of ACA in the approximation of the low rank block stemming form elastodynamic kernels.

The memory storage required by the hierarchical collocation matrix and by the hierarchical preconditioner for each Laplace parameter and for various mesh sizes is then analyzed and results are reported in Fig.2. Also the memory requirement for the elastostatic counterpart of the analyzed problem is reported in the same figure, for the sake of comparison, as the first value of the plotted curves ($k=-1$). As it can be noted, the amount of required memory increases when the imaginary part of the Laplace parameter increases, due to the behavior of ACA with oscillatory kernels. It is to be noted that the strategy of the *local preconditioners* has been used and this explains the memory trends for the preconditioner. Moreover, for a given Laplace parameter, analogously to what happens in the static case, the storage memory, expressed as percentage of the full rank storage, decreases when the mesh size increases.

Figure 3 reports the assembly speed up ratio for various Laplace parameters and various mesh sizes. Also in this case the value corresponding to the static case is reported as the first value of the curves. The speed up ratio is defined as the ratio between the time necessary to perform an operation in hierarchical format and the corresponding classical time. It is to be noted that the classical assembly of the matrix contributions

$\tilde{\mathbf{A}}(s_k)$ requires less time in comparison to the classical assembly of the contribution $\mathbf{A}$, that stems from the integration of the singular integrals. On the other hand, ACA converges with an average rank which is greater or equal than the average rank in the static case. This consideration explains the jump with respect to the static case in the assembly speed up ratios depicted in Fig.3. It can be noted how the assembly speed up ratios grow with the Laplace parameter. This growth is considerable and can lead to assembly times greater than the full rank assembly time. This behavior is a direct consequence of the performance of ACA with the considered elastodynamic kernels and it is related to the growth of the average rank of the approximated blocks.

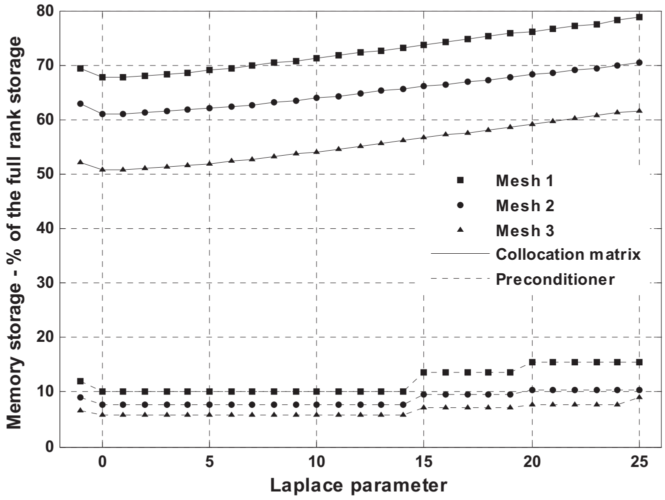

Figure 2 Memory requirements for various Laplace parameters and mesh sizes.

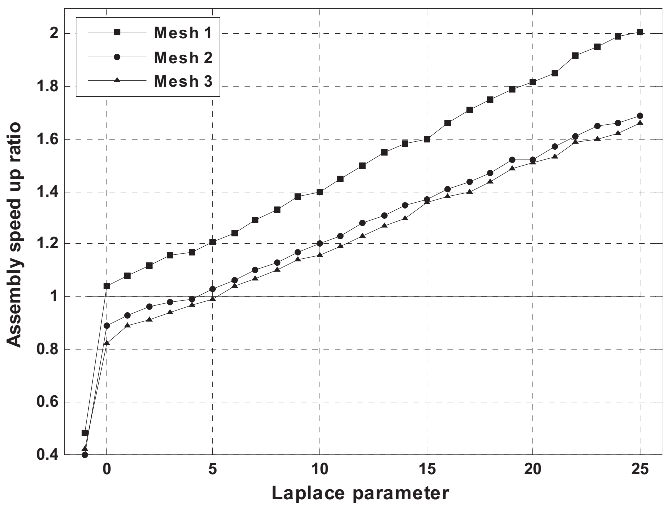

Figure 3 Assembly speed up ratio for various Laplace parameters and mesh sizes.

However, even when the hierarchical assembly times are higher than the full rank assembly times, the advantages in terms of *solution* time can be considerable and allow actual savings in terms of *total* (assembly plus solution) analysis time, as shown in Figs. 4 and 5. These figures demonstrate the effectiveness of the strategy of the *local preconditioners* introduced in the previous section, as well as the noticeable speed up ratios obtained for the three different meshes for various Laplace parameters.

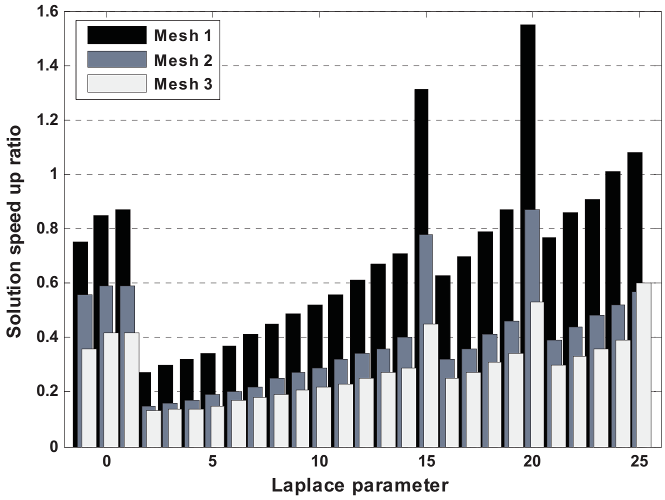

Figure 4 Solution speed up ratios for various Laplace parameters and mesh sizes.

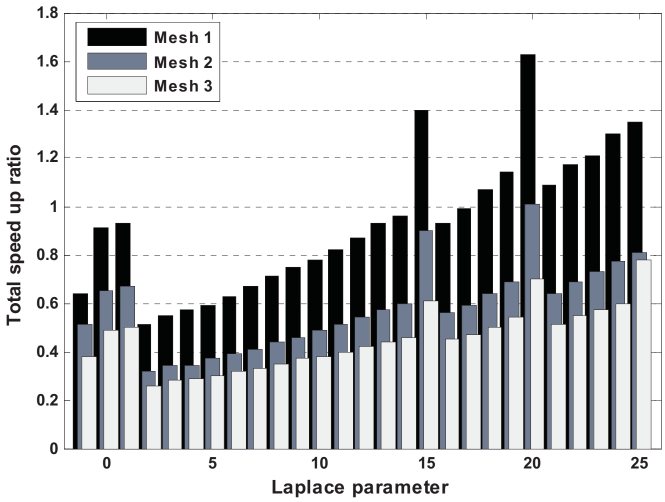

Figure 5 Total speed up ratios for various Laplace parameters and mesh sizes.

The idea of local preconditioners is further clarified by the number of GMRES iterations to convergence reported in Fig. 6 for the various Laplace parameters and different mesh sizes. When the preconditioner is computed for a certain value s_k, the number of iterations to convergence is relatively low. A given computed preconditioner is then used for the following values of the Laplace parameter and demonstrates to be able to precondition the system effectively. The number of GMRES iterations grows when the preconditioner is used *far* from the parameter for which it was computed. When the number of iterations overcomes a prefixed threshold, 140 in the figure, a new preconditioner is computed. The trends in terms of GMRES iterations confirm the effectiveness of the idea.

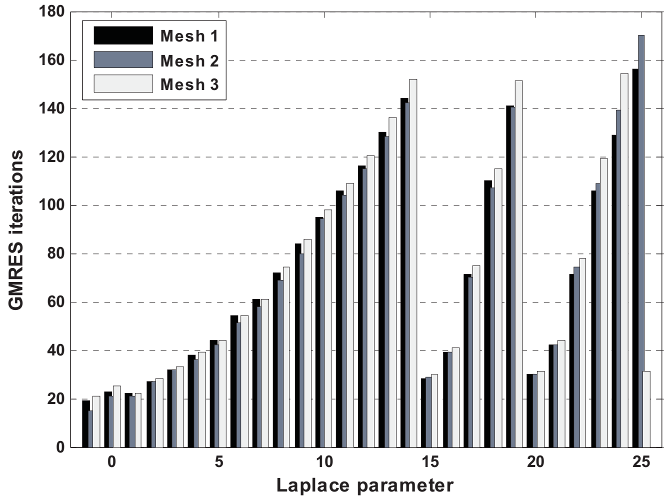

Figure 6 Number of GMRES iterations for various Laplace parameters and mesh sizes.

Summary

In this work a fast solution strategy for 3D elastodynamic BEM problems in the Laplace transform domain has been presented, implemented and tested. The strategy is based on the use of hierarchical matrices for the representation of the collocation matrix in the Laplace domain and uses a GMRES iterative solver for the solution of the final system. The hierarchical format allows to reduce the storage memory necessary for the representation of the system and speeds up the performance of the solver enhancing the speed of the matrix-vector product, which is the core and the bottleneck of any iterative solver for large systems. A coarse hierarchical LU preconditioner is used to improve the convergence of the iterative solution. Moreover, to further speed up the overall analysis in the Laplace domain, the idea of *local preconditioners* is presented and successfully assessed.

Acknowledgements

This work was partially funded within the European research project SEAT (Smart tEchnologies for stress free Air Travel) AST5-CT-2006-030958.

References

[1] G.D. Manolis and D.E. Beskos, *Boundary element methods in elastodynamics*, Unwin Hyman, London (1988).

[2] J. Dominguez, *Boundary elements in dynamics*, Computational Mechanics Publications, Elsevier Applied Science", London", (1993).

[3] M.H. Aliabadi, *The Boundary Element Method: Applications in Solids and Structures*, vol. 2. John Wiley & Sons Ltd (2002).

[4] P. Fedelinski, M.H. Aliabadi and D.P. Rooke, *Int J Sol Struct*, **32**, 3555-3571, (1995).

[5] P. Fedelinski, M.H. Aliabadi and D.P. Rooke, *Comp & Struct*, **59**(6), 1021-1031", (1996).

[6] P.H. Wen, M.H. Aliabadi and D.P. Rooke, *Comp Meth App Mech Eng*, **167**, 139-151, (1998).

[7] P.H. Wen, M.H. Aliabadi and D.P. Rooke, *Comp Meth App Mech Eng*, **173**, 365-374, (1999).

[8] P. Fedelinski, M.H. Aliabadi and D.P. Rooke, *Eng Anal Bound El*, **17**, 45-56, (1996).

[9] P.H. Wen, M.H. Aliabadi and A. Young*, J Strain Anal Eng Design*, **34**(6), 373-394", (1999).

[10] H. Rokhlin, *J Comp Phys*, **60**, 187-207, (1985).

[11] V. Popov, H. Power, *Eng. An. Bound. Elem.*, **25**, 7–18 (2001).

[12] W. Hackbusch and Z.P. Nowak, *Numerische Mathematik*, **73**, 207-243, (1989).

[13] E. E. Tyrtyshnikov, *Calcolo*, **33**, 47-57, (1996).

[14] W. Hackbusch, *Computing*, **62**, 89-108, (1999).

[15] T. Takahashi, N. Nishimura and S. Kobayashi, *Eng Anal Bound El*, **27**, 491-506, (2003).

[16] H. Fujiwara, *Geophys J Int*, **133**, 773-782, (1998).

[17] H. Fujiwara, *Geophys J Int*, **140**, 198-210, (2000).

[18] S. Chaillat, M. Bonnet and J-F. Semblat, J-F, *Comp Meth App Mech Eng*, **197**, 4233-4249, (2008).

[19] M. Messner and M. Schanz, *PAMM Proc App Math Mech*, **8**, 10309-10310, (2008).

[20] I. Benedetti, M.H. Aliabadi, G.Davì, *Int. J. Solids Structures*, **45**, 2355-2376 (2008).

[21] I. Benedetti, A. Milazzo, M.H. Aliabadi, "A fast Dual Boundary Element Method for 3D Anisotropic Crack Problems", *Int J Num Meth Eng*, Accepted for pubblication

[22] M. Bebendorf, *Numerische Mathematik*, **86**, 565-589, (2000).

[23] M. Bebendorf, S. Rjasanow, *Computing*, **70**, 1-24, (2003).

[24] S. Börm, L. Grasedyck and W. Hackbusch, *Eng. An. Bound. Elem.*, **27**, 405–422, (2003).

[25] M. Bebendorf, Effiziente numerische Lösung von Randintegralgleichungen unter Verwendung von Niedrigrang-Matrizen, Ph.D. Thesis, Universität Saarbrücken, 2000. dissertation.de, Verlag im Internet, ISBN 3-89825-183-7, (2001)

[26] L. Grasedyck, *Computing*, **74**, 205-223, (2005).

[27] L. Grasedyck, W. Hackbush, *Computing*, **70**, 295-334, (2003).

[28] M. Bebendorf*, Computing*, **74**, 225-247, (2005).