

# Hierarchical matrices and adaptive cross approximation applied to the boundary element method with multi-domain governed by iterative coupling

T. Grytsenko & A. Peratta

*Wessex Institute of Technology, Southampton, UK*

## Abstract

The approach presented in this paper is based on the Adaptive Cross Approximation (ACA) applied to the matrices coming from the Boundary Element Method (BEM) with multi-domain. The algorithm uses a hierarchical matrix (H-matrix) storage approach splitting the coefficient matrices representing the interactions inside the sub-domains into many blocks where rank of the off-diagonal blocks is reduced with the help of ACA approximation. The sub-domains are then coupled through the iterative process. These optimisations of the coefficient matrices in conjunction with highly effective algorithms for manipulation with H-matrices allow one to perform the operation of matrix-vector multiplication with almost linear complexity  $O(N \log N)$ . The approach allows one to solve the linear systems of equations for BEM with multi-domain having nearly 100.000 DOFs using the usual PC. This paper formulates the approach and demonstrates its numerical properties by means of a theoretical example involving a cube with 27 sub-domains.

## 1 Introduction

The Boundary Element Method (BEM) with multi-domain [1–3] produces as many coefficient matrices as the number of sub-domains in the system. Those coefficient matrices can be coupled within either one matrix or unified into one system through the iterative process [4].

The former approach works well if the neighboring sub-domains are coupled exploiting the continuity of potentials and normal fluxes. However, if this is not



the case and the relationship between the sub-domains have to be expressed using the constraint equations, the latter approach has to be used [4].

In the case with iterative coupling of the sub-domains, each sub-domain produces a dense coefficient matrix having  $O(N^2)$  storage requirement. The solution of such matrices forces the usage of direct LU-based methods requiring  $O(N^3)$  of floating point operations.

In this way the calculation becomes too expensive from a computational point of view thus representing a big obstacle for solving large scale BEM problems involving more than 20.000 unknowns.

The method presented in this paper allows to compresses the dense matrices using the Adaptive Cross Approximation (ACA) [5] algorithm. The method uses a hierarchical matrix (H-matrix) storage approach [6] where each sub-domain of the model is split into many blocks classified into two categories, weakly and strongly coupled. The formers are off-diagonal blocks which represent remote interactions between the source points and field elements, and therefore can be approximated by low-rank matrices using the ACA approach [5]. These blocks are stored in a special Rk-format [6]. The latter blocks describing close interactions between the source points and field elements are stored without any changes in a full-matrix format [6].

This reorganisation of the LSE, implemented in conjunction with algorithms for manipulation with H-matrices reduces the calculation complexity of matrix-vector multiplication to approximately  $O(N \log N)$ . The simplification of MVM reduces considerably the computational burden of the solving stage and shows much better solution time in comparison with standard iterative solvers such as preconditioned GMRES, CG etc.

This paper formulates the approach and comments on the performance of BEM with multi-domain using the hierarchical matrices and ACA. The theoretical example involving nearly 100.000 DOFs is solved on a usual desktop PC.

The paper is organised as follows: in Section 2, the BEM with multi-domain method formulation for Laplace equation is considered; Section 3 formulates the method for iterative coupling of the sub-domains; Section 4 demonstrates the numerical results of application to the theoretical example involving a model with 27 sub-domains and finally Section 5 makes the conclusions.

## 2 Boundary Element Method formulation

Consider solving the potential equation for the unknown scalar field  $u(\mathbf{x})$  given by:

$$\nabla \cdot [-k \nabla u(\mathbf{x})] = 0, \quad \mathbf{x} \in \Omega \subseteq R^3 \quad (1)$$

where  $k$  is conductivity,  $\Omega$  is the integration domain with boundary  $\Gamma = \partial(\Omega)$  of outward unit normal  $\hat{\mathbf{n}}$ , and proper boundary conditions are applied to  $\Gamma$ , i.e. Dirichlet or Neumann type. Then, the boundary integral formulation for eq. (1)



can be expressed in the following way [7, 8]:

$$c_i u(\mathbf{x}_i) + \int_{\Gamma} q^*(\mathbf{x}, \mathbf{x}_i) u(\mathbf{x}) d\Gamma - \int_{\Gamma} u^*(\mathbf{x}, \mathbf{x}_i) q(\mathbf{x}) d\Gamma = 0, \quad (2)$$

where  $u$  is the unknown field,  $q$  its normal derivative in  $\hat{\mathbf{n}}$  direction,  $u^*$  the Green's function of Laplace equation such that  $\nabla^2 u^* + \delta(\mathbf{x}_i, \mathbf{x}) = 0$ ,  $q^*$  its normal derivative in  $\hat{\mathbf{n}}$  direction, and  $c_s$  is the self-interaction coefficient. In 3D problems  $u^*$  and  $q^*$  become:  $u^* = 1/(4\pi r)$  and  $q^* = -\mathbf{r} \cdot \hat{\mathbf{n}}/(4\pi r^3)$ , respectively, where  $\mathbf{r} = \mathbf{x} - \mathbf{x}_i$  and  $r = |\mathbf{r}|$  is the distance between the field ( $\mathbf{x}$ ) and source ( $\mathbf{x}_i$ ) points.

In order to solve eq.(2),  $\Gamma$  is discretised into  $N_e$  constant triangular boundary elements  $\Gamma$ . Thus, the discretised boundary integral equation becomes:

$$c_i u_i + \sum_{j=1}^{N_e} h_j u_j - \sum_{j=1}^{N_e} g_j q_j = 0, \quad (3)$$

where  $u_e$  is the potential at the CFN in the  $j$ -th element,  $q_j$  is the mean normal flux at central DFN of  $j$ -th element, and  $q_j$  and  $h_j$  are the following integrals:

$$g_j = \frac{1}{4\pi} \int_{\Gamma_j} \frac{1}{r} d\Gamma_j \quad (4)$$

$$h_j = -\frac{1}{4\pi} \int_{\Gamma_j} \frac{\mathbf{r} \cdot \hat{\mathbf{n}}}{r^3} d\Gamma_j \quad (5)$$

The assembly scheme consists in appending one equation (2) per each selected source point  $\mathbf{x}_i$  per sub-domain to the appropriate system of equations ( $\mathbf{A} \mathbf{x} = \mathbf{b}$ ), where  $\mathbf{A} \in R^{n \times m}$  contains the coefficients  $h_j$  and  $g_j$ ,  $\mathbf{x}$  is a 1-column array with the unknown  $u$  and  $q$ ,  $\mathbf{b}$  is the right hand side 1-column array formed by the boundary conditions. The matrix  $A$  will be presented in hierarchical format [6].

### 3 The method formulation

An initial BEM 3D model is decoupled into a number of independent sub-domains  $\{\Omega_1, \Omega_2, \dots, \Omega_N\}$  each of which usually has different material properties. Those sub-domains form the appropriate linear systems of equations  $A_i x_i = b_i$  according to Section 2, where  $i$  is an index of sub-domain. The matrices  $\{A_1, A_2, \dots, A_N\}$  that correspond to the sub-domains  $\Omega_1, \Omega_2, \dots, \Omega_N$  are represented in hierarchical format [6]. Each sub-domain is solved independently taking into account the results from all its neighbors, i.e. each iteration includes the following stages:

1. The method forms an initial guess for  $q$  that is located on the interfaces between the current sub-domain  $A$  and all its adjacent neighbors. In general, as an initial guess either  $u$  or  $q$  can be considered, but for this particular explanation it is assumed that the initial guess is always  $q$ . In order to refresh the right-hand side vector **RHS**, the initial guess has to be applied as the artificial Neumann boundary conditions. For the sake of simplicity, the part



of the coefficient matrix corresponding to the imposed artificial Neumann boundary conditions is allocated as a separate structure called refreshing matrix **R**. Thus, each sub-domain forms the local LSE:  $\mathbf{A} \mathbf{x} + \mathbf{R} \mathbf{Guess} = \mathbf{RHS}$ .

2. Update the **RHS** vector of a local LSE with the current value of **Guess** vector.
3. Solve the system of equation  $\mathbf{A}_i \mathbf{x}_i + \mathbf{R}_i \mathbf{Guess}_i = \mathbf{RHS}_i$  for the current sub-domain and obtain the fluxes and the potentials at each freedom node.
4. The potentials that are the result of previous step are imposed as the artificial Dirichlet boundary conditions to all adjacent neighbors. Consequently, all adjacent sub-domains have the new values for the potentials  $u$ .
5. For each sub-domain the following steps has to be performed: update the right hand side term with either artificial boundary conditions or imposed initial guess or both; solve the system of equations to get the normal fluxes at the freedom nodes.
6. Update the normal fluxes  $q_{\Omega_1}^* = q_{\Omega_1} + (q_{\Omega_1} - q_{\Omega_2})/r$ , where  $r$  is a relaxation factor  $r > 1$ .
7. Test the convergence

$$MAX(u_{n+1} - u_n)/u_n < error \quad (6)$$

where  $u_{n+1}$  is a value of the potential at  $n + 1$  iteration whereas  $u_n$  is at iteration  $n$ ;

$$MAX(q_{\Omega_1} - q_{\Omega_2})/q_{\Omega_1} < error \quad (7)$$

where  $q_{\Omega_1}$  is a value of a normal flux for the sub-domain  $\Omega_1$  and  $q_{\Omega_2}$  is a value of the corresponding normal flux in sub-domain  $\Omega_2$ . if (6-7) are not achieved return to step 2.

The system of equations for each sub-domain is computed and approximated only once at the very beginning of iterative scheme whereas the RHS is refreshed with every iteration.

An example (see Figure 1) shows how the iterative technique is applied.

The sub-domain  $\Omega_1$  has 6 surfaces and only 4 which the boundary conditions are known for, i.e. there are two surfaces with unknown boundary conditions. At the very beginning of iterative scheme, an initial guess has to be formed for those two surfaces, i.e. the boundary conditions for the normal flux over those surfaces have to be imposed. Hence, after this operation, the sub-domain  $\Omega_1$  has the boundary conditions for all surfaces. The coefficient matrix  $A_H^1$  can now be built and the RHS can be computed taking into account imposed boundary conditions  $q_1^1$  and  $q_2^1$ . Once the system  $A_H^1 x_1 = RHS_1$  is solved and appropriate values for  $u_1^1$  and  $u_2^1$  are known, they can be applied as temporary Dirichlet-type boundary conditions to the sub-domains  $\Omega_2$  and  $\Omega_3$  appropriately. Then, the sub-domain  $\Omega_2$  has the boundary conditions for all surfaces excluding the only one, where an initial guess  $q_2^2$  has to be applied again. Once the system  $A_H^2 x_2 = RHS_2$  is solved it will produce the BC for the sub-domain  $\Omega_4$ . In this way, an initial guess  $q_2^3$  has to be applied in order to produce the integrity of the whole system. The iterative scheme

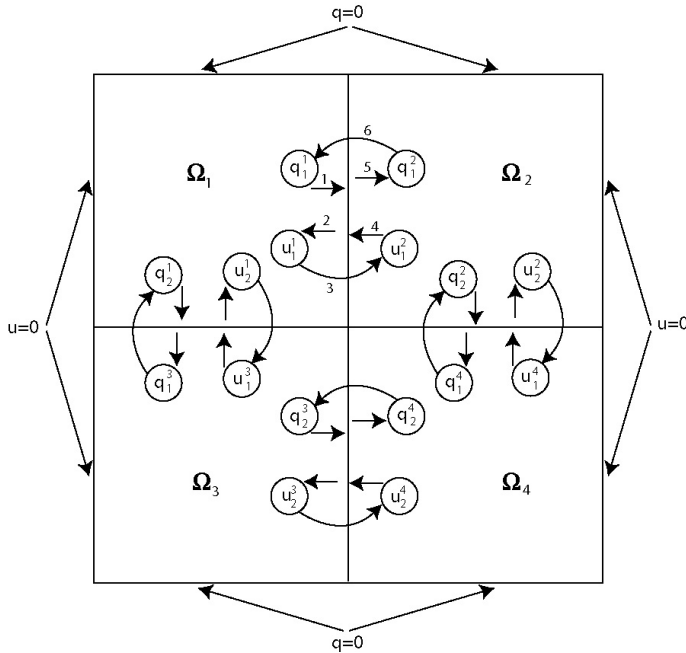


Figure 1: Iterative technique for BEM with multi-domain. The strategy for coupling two sub-domains consists of six steps: 1 - to solve the LSE for a local sub-domain taking into account the initial guess for the normal fluxes distributed over the interface, 2 - to get the solution for the potentials located on the same interface, 3 - use the solution for the potentials as the artificial BC which is at the same time the initial guess for the connected sub-domain (4), 5 - solve the LSE for the next sub-domain and get the solution for the normal fluxes, 6 - compare the solution for the normal fluxes with the initial guess for the first sub-domain.

for the sub-domain  $\Omega_1$  of the model demonstrated in Figure 1 is shown in Figure 2. At the next stage of iterative scheme, the new values for  $q_1^1$ ,  $q_2^1$ ,  $q_2^2$  and  $q_3^2$  has to be computed and appropriate RHS are refreshed. In order to do so, let's consider the sub-domains  $\Omega_1$  and  $\Omega_2$ . Once the system  $A_H^2 x_2 = RHS_2$  is solved taking into account an enforced value for  $u_1^2$ , an appropriate value for  $q_1^2$  will be computed. In fact,  $q_1^1$  and  $q_1^2$  correspond to the normal flux of the same element and therefore they have to be compared in order to be adjusted for the next iteration. Having an initial value for  $q_1^1$  from  $\Omega_1$  as well as a new one  $q_1^2$  from  $\Omega_2$ , those two values can be prepared for the next iteration according to the following expression:

$$q_1^{1*} = q_1^1 + (q_1^1 - q_1^2)/r \quad (8)$$

where  $r$  is a relaxation factor  $r > 1$ ,  $q_1^{1*}$  is a new value for  $q_1^1$  from the sub-domain  $\Omega_1$ .

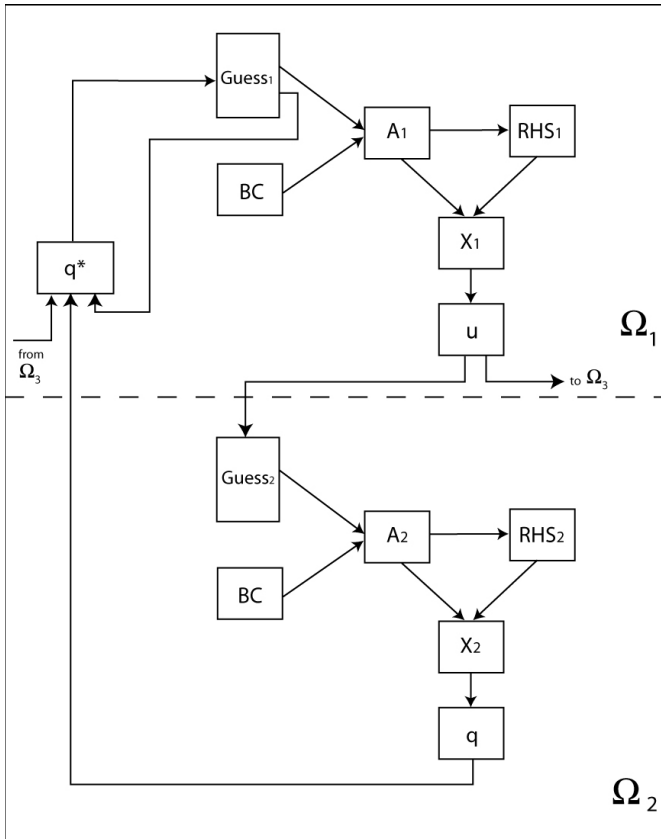


Figure 2: Coupling the sub-domains  $\Omega_1$  and  $\Omega_2$  of the model shown in Figure 1: **A** is the entire coefficient matrix, **RHS** is the corresponding right hand side, **BC** is the vector of known boundary conditions, **Guess** is the vector of the initial guess, **X** is the result vector and **u** and **q** are the vectors with the current results for the potentials and normal flux respectively.

In this way, all imposed BC has to be updated and the RHS refreshed. Once this is done, the next iteration starts. At the end of each iteration the convergence has to be tested in order to make a decision if more iterations are needed see eq.(6), i.e. if the maximal relative difference between the potential at current iteration and at the previous one is less than an appropriate *error* which is set up in advance, then the scheme must stop. Instead of the potentials, the normal fluxes can be used.

Another expression (7) is for the normal fluxes only, i.e. if the maximal relative difference between all corresponding normal fluxes over the all interfaces shared between two volumes (in this case  $\Omega_1$  and  $\Omega_2$ ) is less than an appropriate *error* which is set up in advance, then the scheme must stop. Both expressions (6-7) are used at the same time in order to secure the precision of iterative scheme.

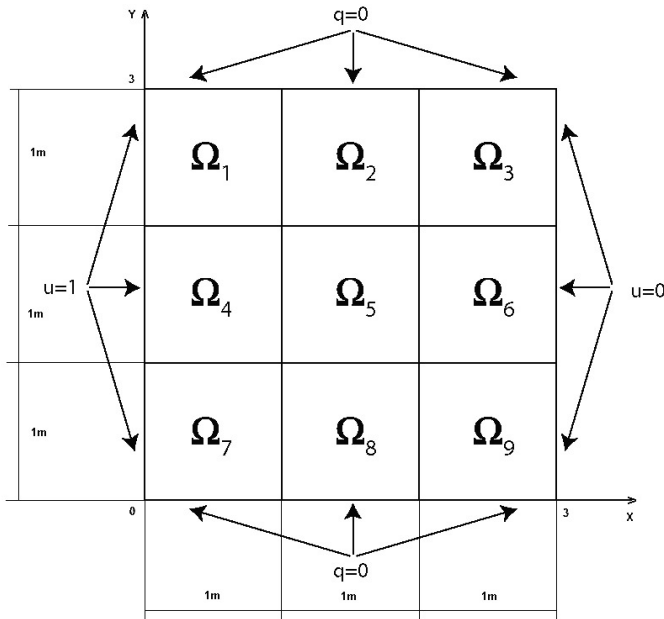


Figure 3: The front 2D representation of 3D BEM with a multi-domain model consisting of 27 sub-domains with uniformly distributed conductivity  $k = 1$ .

## 4 Case study

This section examines the theoretical example in order to verify the accuracy, storage requirements and the complexity of the solution of LSE for BEM with multi-domain using the ACA-based approach and iterative coupling.

### 4.1 The model description

The problem represents a Laplacian equation in a cube consisting of 27 sub-cubes coupled together as shown in Figure 3. The model is meshed with triangular constant elements involving different level of mesh refinement yielding up to 95504 BEs.

Despite the ability of the developed method to deal with the models where the sub-domains have different material properties, the conductivity of the sub-domains is selected to be the same everywhere  $k = 1$ . This allows to check the accuracy of the numerical solution against the analytical one which is known for the 'Cube-like' and in this particular case is defined as  $u^x = 1 - x/3$  for the potentials, where  $x$  is the  $x$ -component in 3D-coordinates  $(x, y, z)$  of the DOF (see Figure 3).



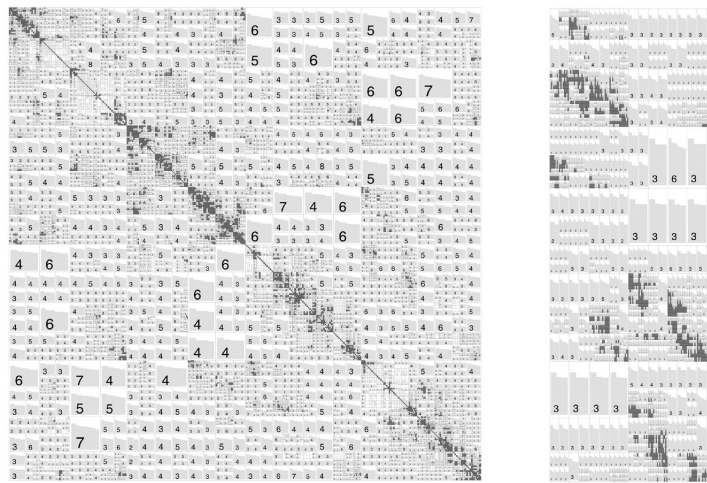


Figure 4: The coefficient H-matrix (to the right) and the refreshing H-matrix (to the left) for the sub-domain  $\Omega_1$  of the model shown in Figure 3 meshed with 22566 triangular BEs: dark blocks are Full-matrices and the grey ones are Rk-blocks.

Figure 4 demonstrates how the coefficient and refreshing matrix looks for a sub-domain  $\Omega_1$  of the model introduced in Figure 3 if it is meshed with 22566 BEs.

4.2 The numerical properties

Tables 1-4 demonstrate the numerical properties of the developed approach.

Table 1: The memory consumption (in Mb) required to store the coefficient and refreshing matrices depending on the ACA-error  $\epsilon_{ACA}$  and  $N$ .

$N/\epsilon_{ACA}$	1e-01	1e-02	1e-03	1e-04	1e-05
1944	3.1	3.1	3.1	3.1	3.1
5616	28.3	30.1	32.2	34.6	36.9
10604	63.2	75.9	89.4	105.9	113.4
22566	157.4	203.9	255.5	319.4	390.6
47568	275.5	462.2	649.5	882	1128
95504	833	1156	1509.2	-	-



Table 2: The CPU time (in sec) required to solve the LSE depending on the ACA-error  $\varepsilon_{ACA}$  and  $N$ .

$N/\varepsilon_{ACA}$	1e-01	1e-02	1e-03	1e-04	1e-05
1944	14	14	14	14	14
5616	19	20	23	25	26
10604	23	25	28	40	42
22566	63	68	77	98	119
47568	153	178	224	282	366
95504	390	425	-	-	-

Table 3: The second norm of the difference between the analytical  $S_{an}$  and numerical  $S_{num}$  solution  $\varepsilon_{AN} = \|S_{an} - S_{num}\|_2$  in function of both the number of DOFs  $N$  and  $\varepsilon_{ACA}$ .

$N/\varepsilon_{ACA}$	1e-01	1e-02	1e-03	1e-04	1e-05
1944	0.0071	0.0071	0.0071	0.0071	0.0071
5616	0.014	0.0037	0.0032	0.0032	0.0032
10604	0.024	0.0029	0.0026	0.0026	0.0026
22566	0.034	0.0029	0.0024	0.0024	0.0024
47568	0.051	0.011	0.01	0.01	0.01
95504	0.049	0.008	-	-	-

## 5 Conclusions

The memory consumption grows as the  $\varepsilon_{ACA}$  decreases as well as grows proportionally to the increase of  $N$  as shown in Table 1.

As it is observed, the compression ratio increases as the number of unknowns increases. Thus, the same model meshed with nearly 100.000 BEs has higher compression ratio (8.7%) than the model meshed with 10.000 of BEs (46.6%).

The method requires more CPU time if  $\varepsilon_{ACA}$  decreases as well as when the number of unknowns increases (see Table 2).

As is shown in Table 3,  $\varepsilon_{AN}$  decreases as the number of BEs increases. However, if the ACA-error  $\varepsilon_{ACA}$  is high, the error  $\varepsilon_{AN}$  may grow due to the fact that the off-diagonal blocks of the coefficient matrix are heavily approximated

Table 4: The compression ratio of the coefficient and refreshing matrices depending on the ACA-error  $\varepsilon_{ACA}$  and  $N$ .

$N/\varepsilon_{ACA}$	1e-01	1e-02	1e-03	1e-04	1e-05
1944	101.6	101.6	101.6	101.6	101.6
5616	82.6	89	96.4	104.8	113
10604	46.6	59.8	73.2	89.9	107.2
22566	27.2	36.5	46.8	59.4	73.4
47568	11.7	19.6	27.6	37.3	47.7
95504	8.7	12.1	15.8	-	-

causing the serious damage to the coefficient matrix. Nevertheless, in general case the higher level of mesh refinement improves the precision of the solution against the analytical one.

## References

- [1] Erhart K., Divo E., and Kassab A.J. A parallel domain decomposition boundary element method technique for large-scale transient heat conduction problems. *Numerical Heat Transfer*, 2006.
- [2] P. Skerget M. Ribersek and I. Zagar. Boundary-domain integral method with subdomain technique for time dependent viscous flow. *Z. Angew Math. Mech*, 73:935–939, 1993.
- [3] W. Florez, H. Power, and F. Chejne. Numerical solution of thermal convection problems using the multidomain boundary element method. *J. Num. Meth. for Partial Differential Equations*, 18(4):469–489, 2002.
- [4] A. Peratta and V. Popov. A new scheme for numerical modelling of flow and transport processes in 3D fractured porous media. *Advances in Water Resources*, 29:42–61, 2006.
- [5] M. Bebendorf and S. Rjasanov. Adaptive low-rank approximation of collocation matrices. *Computing*, 70:1–24, 2003.
- [6] Lars Grasedyck Stefen Borm and Wolfgang Hackbusch. Hierarchical matrices. April 2005.
- [7] C. Brebbia, J. Telles, and L. Wrobel. *Boundary Elements Techniques*. Springer-Verlag, Berlin, Heidelberg, New York and Tokio, 1984.
- [8] C. Brebbia and J.Dominguez. *Boundary Elements, an Introductory Course*. Computational Mechanics Publications. McGraw-Hill, New York, Colorado, San Francisco, Mexico, Toronto, 2nd edition, 1992.

