

3D-ACA FOR THE TIME DOMAIN BOUNDARY ELEMENT METHOD: COMPARISON OF FMM AND \mathcal{H} -MATRIX BASED APPROACHES

MARTIN SCHANZ¹

¹ Graz University of Technology
Technikerstraße 4, 8010 Graz, Austria,
m.schanz@tugraz.at, www.mech.tugraz.at

Key words: Boundary Element Method, generalised Convolution Quadrature, Fast Multipole Method, generalised ACA

Abstract. The time domain Boundary Element Method (BEM) for the homogeneous wave equation with vanishing initial conditions is considered. The generalized convolution quadrature method (gCQ) developed by Lopez-Fernandez and Sauter is used for the temporal discretisation. The spatial discretisation is done classically using low order shape functions. A collocation approach is applied.

Essentially, the gCQ requires to establish boundary element matrices of the corresponding elliptic problem in Laplace domain at several complex frequencies. Consequently, an array of system matrices is obtained. This array of system matrices can be interpreted as a three-dimensional array of data which should be approximated by a data-sparse representation. The generalised Adaptive Cross Approximation (3D-ACA) can be applied to get a data sparse representation of these three-dimensional data arrays. Adaptively, the rank of the three-dimensional data array is increased until a prescribed accuracy is obtained. On a pure algebraic level it is decided whether a low-rank approximation of the three-dimensional data array is close enough to the original matrix. Within the data slices corresponding to the BEM calculations at each frequency either the standard \mathcal{H} -matrices approach with ACA or a fast multipole (FMM) approach can be used. The third dimension of the data array represents the complex frequencies. Hence, the algorithm makes not only a data sparse approximation in the two spatial dimensions but detects adaptively how much frequencies are necessary for which matrix block. Numerical studies show the performance of these methods.

1 INTRODUCTION

Wave propagation problems appear often in engineering, e.g., for non-destructive testing or exploring the underground. Most of such problems are formulated with hyperbolic partial differential equations, e.g., in acoustics or elastodynamics. Despite that mostly a linear theory is sufficient, the handling of space and time requires expensive discretisation methods, where for scattering problems even an unbounded domain has to be considered. The latter are preferably solved with the boundary element method (BEM). The basis are boundary integral equations

by the use of retarded potentials as counterpart to the governing hyperbolic partial differential equation. The first numerical realisation of a time domain boundary element formulation is originated by Mansur [21] in the 80th of the last century. Despite often used, this approach suffers from instabilities (see, e.g., [23]). A stable space-time formulation has been published by Bamberger and Ha-Duong [3], which has been further explored by the group of Aimi [1, 2]. These approaches work directly in time domain, whereas a transformation to Laplace- or Fourier-domain results in suitable formulations as well, e.g., [9]. Somehow in between transformation and time-domain methods are BE formulations based on the convolution quadrature (CQ) method proposed by Lubich [19, 20]. Such a formulation is a true time stepping method utilising the Laplace domain fundamental solutions and properties. Applications of the CQ to BEM can be found, e.g., in [26, 27]. The generalisation of this seminal technique to variable time step sizes has been proposed by López-Fernández and Sauter [16, 18] and is called generalised convolution quadrature method (gCQ). Applications can be found in acoustics with absorbing boundary conditions [24] and in thermoelasticity [15].

The drawback of all BE formulations either for elliptic and much stronger for hyperbolic problems is the high storage and computing time demand as a standard formulation scales with $\mathcal{O}(M^2)$ for M unknowns. In time domain, additionally, the time complexity has to be considered, where in the case of a CQ based formulation the complexity is of order $\mathcal{O}(M^2N)$ for N time steps. For elliptic problems fast methods has been proposed as the fast multipole method (FMM) [13] or \mathcal{H} -matrix based methods with the adaptive cross approximation (ACA) used in the matrix blocks [8, 5]. The extension of FMM to the time variable has been published in [11] for acoustics. In combination with CQ, fast methods are published in combination with a reformulation of CQ [4, 22].

Here, a different approach is used. Independently whether the CQ in the original form or gCQ is used, essentially, a three-dimensional data array has to be efficiently computed and stored. This data array is constructed by the spatial discretisation, resulting in two-dimensional data, and the used complex frequencies of the algorithm, which gives the third dimension. To find a low rank representation of this three-dimensional tensor the generalised adaptive cross approximation (3D-ACA) can be used. This technique is a generalisation of ACA [8] and is proposed by Bebendorf et al. [6, 7].

Here, the 3D-ACA is applied on a gCQ based time domain formulation utilising the original idea of the multivariate ACA [7]. The original version utilising ACA in the \mathcal{H} -matrices for the frequencies will be compared to a version where the FMM is used. Recently a very similar approach has been published by Seibel [29], where the conventional convolution quadrature method is used and, contrary to the approach proposed here, the \mathcal{H}^2 technique is used for the approximation of the boundary element matrices at the different frequencies.

2 PROBLEM SETTING

To present the proposed algorithm a Dirichlet problem is used as model problem. Let $\Omega \subset \mathbb{R}^3$ be a bounded Lipschitz domain and $\Gamma = \partial\Omega$ its boundary with the outward normal \mathbf{n} . The

acoustic wave propagation is governed by

$$\begin{aligned} \frac{\partial^2}{\partial t^2} u(\mathbf{x}, t) - c^2 \Delta u(\mathbf{x}, t) &= 0 & (\mathbf{x}, t) \in \Omega \times (0, T) \\ u(\mathbf{x}, 0) = \frac{\partial}{\partial t} u(\mathbf{x}, 0) &= 0 & \mathbf{x} \in \Omega \\ u(\mathbf{x}, t) &= g_D(\mathbf{x}, t) & (\mathbf{x}, t) \in \Gamma \times (0, T) \end{aligned} \quad (1)$$

with the wave speed c and the end time $T > 0$. This Dirichlet problem will be solved with an integral equation, where the so-called indirect approach using layer potentials is selected. The single layer potential

$$(\mathcal{V} * \vartheta)(\mathbf{x}, t) = \int_0^t \int_{\Gamma} U(\mathbf{x} - \mathbf{y}, t - \tau) \vartheta(\mathbf{y}, \tau) \, d s_{\mathbf{y}} \, d \tau, \quad (2)$$

solves the problem (1), with the fundamental solution $U(\mathbf{x} - \mathbf{y}, t - \tau) = \frac{1}{4\pi \|\mathbf{x} - \mathbf{y}\|} \delta\left(t - \tau - \frac{\|\mathbf{x} - \mathbf{y}\|}{c}\right)$. This boundary integral equation is weakly singular. Alternatively, the problem can be solved utilising the double layer potential defined by

$$(\mathcal{K} * \Theta)(\mathbf{x}, t) = \int_0^t \int_{\Gamma} \nabla U(\mathbf{x} - \mathbf{y}, t - \tau) \cdot \mathbf{n} \Theta(\mathbf{y}, \tau) \, d s_{\mathbf{y}} \, d \tau. \quad (3)$$

Above the operator $\nabla U \cdot \mathbf{n}$ is the usual co-normal derivative applied on the fundamental solution resulting in the flux fundamental solution. The density functions are denoted by $\vartheta(\mathbf{x}, t)$ or $\Theta(\mathbf{x}, t)$, respectively. Using these operators, which are also called retarded potentials, the solution for these density functions can be obtained by

$$(\mathcal{V} * \vartheta)(\mathbf{x}, t) = g_D(\mathbf{x}, t) \quad (\mathbf{x}, t) \in \Gamma_D \times (0, T) \quad (4)$$

or considering the double layer potential

$$(\mathcal{C}\Theta + \mathcal{K} * \Theta)(\mathbf{x}, t) = g_D(\mathbf{x}, t) \quad (\mathbf{x}, t) \in \Gamma \times (0, T). \quad (5)$$

The so-called integral free term in (5) is determined with the limiting process

$$\mathcal{C}\Theta(\mathbf{x}, t) = \lim_{\varepsilon \rightarrow 0} \int_{\partial B_{\varepsilon}(\mathbf{x}) \cap \Omega} \nabla \frac{1}{\|\mathbf{x} - \mathbf{y}\|} \cdot \mathbf{n} \Theta(\mathbf{y}, t) \, d s_{\mathbf{y}} \quad (6)$$

with $B_{\varepsilon}(\mathbf{x})$ denoting a ball of radius ε centered at \mathbf{x} and $\partial B_{\varepsilon}(\mathbf{x})$ is its surface. In case of a Galerkin formulation this expression reduces to $1/2\Theta(\mathbf{x}, t)$ but in a collocation schema it is dependent on the geometry at the collocation node. The physical solution $u(\mathbf{x}, t)$ is obtained with the determined density function inserted in the layer potentials.

3 BOUNDARY ELEMENT FORMULATION

Spatial discretisation The boundary Γ is discretised with elements resulting in an approximation $\Gamma_h = \bigcup_{e=1}^E \tau_e$, which is the union of geometrical boundary elements τ_e , here linear surface triangles. Finite element bases on the boundary Γ are used to construct the approximation spaces

$$X_{SLP} = \text{Span}\{\varphi_1, \varphi_2, \dots, \varphi_M\}, \quad X_{DLP} = \text{Span}\{\psi_1, \psi_2, \dots, \psi_M\}. \quad (7)$$

The unknown density functions are approximated by a linear combination of functions in X_{SLP} or X_{DLP}

$$\vartheta_h = \sum_{k=1}^M \vartheta_k(t) \psi_k(\mathbf{x}) \quad \Theta_h = \sum_{k=1}^M \Theta_k(t) \varphi_k(\mathbf{x}). \quad (8)$$

Note, the coefficients $\vartheta_k(t)$, and $\Theta_k(t)$ are still continuous functions of time t . In the following, the shape functions φ_k will be chosen linear continuous and ψ_k constant discontinuous. This is in accordance with the necessary function spaces for the integral equations, which are not discussed here (see [25]).

Hence, inserting the spatial shape functions from above in the boundary integral equation (4) and applying the collocation method results in the semi-discrete equation system

$$\mathbb{V} * \vartheta_h = \mathbf{g}_D. \quad (9)$$

The double layer approach takes (5) with the above shape functions resulting in

$$(\mathbb{C}\Theta_h + \mathbb{K} * \Theta_h) = \mathbf{g}_D. \quad (10)$$

The given boundary data are approximated by the same shape functions. The notation with sans serif fonts in (9) and (10) indicates that in these vectors the nodal values are collected and in the matrices the respective values of the integrated fundamental solutions.

The weak singular integrals are treated with a Duffy transformation [10]. The regular integrals are handled by a standard Gaussian quadrature using a heuristic distance-based formula to determine the number of Gauss points.

Temporal discretisation The above given semi-discrete integral equations are discretised in time using the generalised Convolution Quadrature Method (gCQ) [16, 18]. Here, the variant using Runge-Kutta methods as the underlying time stepping technique is used. Details and the analysis can be found in [18].

Discretising the time in N not necessarily constant time steps Δt_i , i.e., $[0, T] = [0, t_1, \dots, t_N]$, $\Delta t_i = t_i - t_{i-1}$, $i = 1, 2, \dots, N$, the gCQ can be applied. Let us assume an A- and L-stable Runge-Kutta method given by its Butcher tableau $\begin{array}{c|c} c & A \\ \hline & \mathbf{b}^\top \end{array}$ with $A \in \mathbb{R}^{m \times m}$, $\mathbf{b}, \mathbf{c} \in \mathbb{R}^m$ and m is the number of stages. The stability assumptions require that $\mathbf{b}^\top A^{-1} = (0, 0, \dots, 1)$ holds.

With the vector $\mathbb{1} = (1, 1, \dots, 1)^T$ of size m the semi-discretised integral formulation (9) can be approximated by

$$\hat{V}((\Delta t_n \mathbf{A})^{-1})(\vartheta_h)_n = (\mathbf{g}_D)_n - \sum_{\ell=1}^{N_Q} \hat{V}(s_\ell) W^{\Delta t_n}((\vartheta_h)_{n-1}, s_\ell) \quad (11)$$

and the double layer approach (10) is

$$\left(\mathbf{C} + \hat{K}((\Delta t_n \mathbf{A})^{-1}) \right) (\Theta_h)_n = (\mathbf{g}_D)_n - \sum_{\ell=1}^{N_Q} \hat{K}(s_\ell) W^{\Delta t_n}((\Theta_h)_{n-1}, s_\ell) . \quad (12)$$

The notation $(\cdot)_n$ indicates the discrete value of the respective function/vector at t_n and $\hat{(\cdot)}$ indicates that the Laplace transform of the respective integral kernel is used. The abbreviation $W^{\Delta t_n}((\vartheta_h)_{n-1}, s_\ell)$ is an integration weight collecting the influence of the past time steps. The whole algorithm and also all used parameters can be found in [17] or [14] using the same notation as here.

Essentially, this algorithm requires the evaluation of the integral kernel at N_Q points s_ℓ , which are complex frequencies. Consequently, we get an array of system matrices. This array of system matrices can be interpreted as a three-dimensional array of data which will be approximated by a data-sparse representation based on 3D-ACA.

4 GENERALISED ACA: 3D-ACA

An approximation of a three-dimensional array of data or a tensor of third order $\mathcal{C} \in \mathbb{C}^{M \times M \times N_Q}$ in terms of a low-rank approximation has been proposed in [7] and is referred to as a generalisation of adaptive cross approximation or also called 3D-ACA. In this approach, the 3D array of data to be approximated is generated by defining the outer product by

$$\mathcal{C} = \mathbf{H} \otimes \mathbf{f} \quad (13)$$

with $\mathbf{H} \in \mathbb{C}^{M \times M}$, $\mathbf{f} \in \mathbb{C}^{N_Q}$. The matrix \mathbf{H} corresponds to the spatial discretization of the different potentials used in the boundary element formulations from above at a specific frequency s_ℓ , e.g., the single layer potential in (11) on the right hand side. This matrix will be called face or slice and may be computed as dense or approximated matrix. Here, two choices will be utilized:

- The face can be an \mathcal{H} -matrix after introduction of a suitable hierarchical partition. In the examples, a balanced cluster tree [5] is introduced and the adaptive cross approximation (ACA) [8] with recompression is used.
- The face can be approximated in the matrix free representation of the fast multipole method (FMM). In the examples, a uniform cluster tree based on the usual geometric subdivision is applied and the kernel expansion is performed with a Chebychev-interpolation

denoted with S_p [12]

$$\hat{U}(\mathbf{x}, \mathbf{y}, s_\ell) \approx \sum_n S_p(\mathbf{x}, \mathbf{x}_n) \sum_m \hat{U}(\mathbf{x}_n, \mathbf{y}_m, s_\ell) S_p(\mathbf{y}, \mathbf{y}_m). \quad (14)$$

The polynomial order is p and $\mathbf{x}_n, \mathbf{y}_m$ are the respective Chebychev nodes.

The vector \mathbf{f} , called fiber in the following, collects selected elements of \mathbf{H} at the set of frequencies determined by the gCQ. The latter would amount in N_Q entries and the same amount of faces. The 3D-ACA on the one hand approximates, as above mentioned, the faces with low rank matrices or FMM and, more importantly, the amount of necessary frequencies is adaptively determined. Hence, a sum of outer products \mathcal{C}^ℓ as given in (13) is established. The summation is stopped if \mathcal{C}^ℓ is comparable to \mathcal{C} up to a prescribed precision ε . This is measured with a Frobenius-norm. The basic concept of this approach is sketched in Algorithm 1. Essentially,

Algorithm 1 Pseudo code of 3D-ACA (taken from [29])

```

function 3D ACA(ENTRY,  $\varepsilon$ )            $\triangleright$  ENTRY provides the integrated kernel values at
collocation point  $x_i$  and element  $j$ 
 $\mathcal{C}^{(0)} = 0, k_1 = 0$  and  $\ell = 0$ 
while  $\|\mathbf{H}_\ell\|_F \|\mathbf{f}_\ell^k\|_2 > \varepsilon \|\mathcal{C}^{(\ell)}\|_F$  do
     $\ell = \ell + 1$ 
     $H_\ell[i, j] = \text{ENTRY}(i, j, k_\ell) - \mathcal{C}^{(\ell-1)}[i, j, k_\ell], \quad i, j = 1, \dots, M$ 
     $H_\ell[i_\ell, j_\ell] = \max_{i,j} |H_\ell[i, j]|$ 
     $f_\ell[k] = H_\ell[i_\ell, j_\ell]^{-1} (\text{ENTRY}(i_\ell, j_\ell, k) - \mathcal{C}^{(\ell-1)}[i_\ell, j_\ell, k]), \quad k = 1, \dots, N_Q$ 
     $\mathcal{C}^{(\ell)} = \mathcal{C}^{(\ell-1)} + \mathbf{H}_\ell \otimes \mathbf{f}_\ell^k$ 
     $k_{\ell+1} = \arg \max_k |f_\ell[k]|$ 
end while
 $r = \ell - 1$                                 $\triangleright$  Final rank, i.e., necessary frequencies
    return  $\mathcal{C}^r = \sum_{\ell=1}^r \mathbf{H}_\ell \otimes \mathbf{f}_\ell^k$ 
end function
    
```

three-dimensional crosses are established. These crosses consist of a face \mathbf{H}_ℓ , which is the respective matrix at a distinct frequency s_ℓ times a fiber \mathbf{f}_ℓ , which contains one matrix entry, the pivot element, at all N_Q frequencies. These crosses are added up until a predefined error is obtained. The amount of used frequencies is the so-called rank r of this approximation of the initial data cube. The mentioned pivot element is denoted in Algorithm 1 with $H_\ell[i_\ell, j_\ell]$. Like in the 'normal' ACA this pivot element can be selected arbitrarily. However, selecting the maximum element ensures convergence. For details see [7] or the application using \mathcal{H}^2 -matrices in the faces [29].

The stopping criterion requires a norm evaluation of $\mathcal{C}^{(\ell)}$. Assuming some monotonicity the

norm can be computed recursively

$$\|\mathcal{C}^{(\ell)}\|_F^2 = \sum_{d,d'} \left(\sum_{i,j} H_d[i,j] \overline{H_{d'}[i,j]} \right) \left(\sum_k f_d[k] \overline{f_{d'}[k]} \right) \quad (15)$$

following [29].

The multiplication of the three-dimensional data array with a vector is changed by the proposed algorithm. Essentially, the algorithm separates the frequency dependency such that $\mathbf{H}_\ell(\hat{\mathbf{V}})$ is independent of the frequency and the fibers $\mathbf{f}_\ell^k(\hat{\mathbf{V}})$ present this dependency. Let us use the multiplication on the right hand side of (11) as example, which is changed to

$$\sum_{k=1}^{N_Q} \hat{\mathbf{V}}(s_k) \mathbf{W}_k^{\Delta t_n} = \sum_{k=1}^{N_Q} \sum_{\ell=1}^r \mathbf{H}_\ell(\hat{\mathbf{V}}) \otimes \mathbf{f}_\ell^k(\hat{\mathbf{V}}) \mathbf{W}_k^{\Delta t_n} = \sum_{\ell=1}^r \mathbf{H}_\ell(\hat{\mathbf{V}}) \otimes \sum_{k=1}^{N_Q} \mathbf{f}_\ell^k(\hat{\mathbf{V}}) \mathbf{W}_k^{\Delta t_n}. \quad (16)$$

The complexity of the original operation is $\mathcal{O}(N_Q M^2)$ for M spatial unknowns. The approximated version has the complexity $\mathcal{O}(r(M^2 + N_Q))$. It consists of the inner sum, which is a vector times vector multiplication of length N_Q , and the outer sum which is a matrix times vector multiplication of size M . Hence, the leading term with M^2 has only a factor of r instead of N_Q compared to the dense computation. It can be expected that for larger problem sizes with a significant reduction from N_Q to r complex frequencies this discrete convolution is faster.

The above sketched algorithm requires some comments regarding its use and implementation. In principle, this algorithm can be applied to the whole matrix or in case of hierarchical matrix structures to each matrix block. The latter is made here as it is more efficient as long as the same hierarchical structure is used for all frequencies [28]. Further, for the FMM version in the far field blocks the interpolation (14) is used, where only the fundamental solution is frequency dependent. Hence, in a far field block it is sufficient to use them in the algorithm and only in the matrix vector product the polynomials are applied. This allows as well to have a possibility to determine the above discussed pivot element and a straight forward norm computation (15). For the technicalities of the ACA based version see [14].

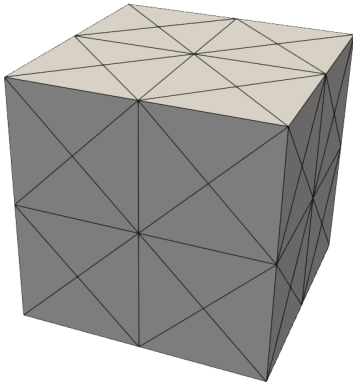
5 NUMERICAL EXAMPLE

The above proposed method to accelerate the gCQ based time domain boundary element method is tested to show that the approximation of the 3D-ACA does not spoil the results, i.e., the newly introduced approximation error is smaller than the error of the dense BE formulation. Main focus is on the overall obtained reduction in storage, i.e., the compression which is measured as the relation between the dense storage and the storage necessary for the proposed method. Further, the two variants FMM-based and ACA-based are compared.

As mentioned above, linear triangles are used for the discretisation of the geometry and either linear continuous or constant discontinuous shape functions for the densities are applied.

The underlying time stepping method has been the 2-stage Radau IIA. The approximation in the ACA version within the faces has been chosen to be $\varepsilon_{ACA} = 10^{-4} \dots 10^{-8}$ for the different refinement levels of the spatial discretisation. The FMM based results are computed with the same depth of the cluster tree as the level number is. The polynomial order is set as level plus 2, i.e., it starts with three up to seven. In each refinement level the mesh size is halved as well as the time step size. The ratio of time steps to mesh size was kept constant with 0.7. The precision of the method with respect to the frequencies, i.e., the ε in Algorithm 1 was selected as $\varepsilon = 100 * \varepsilon_{ACA}$. The final equation is solved with BiCGstab without any preconditioner, where the precision ε_{ACA} is set. All results are computed for a wave speed $c = 1 \text{ m/s}$.

The test geometry is a unit cube $[-0.5, 0.5]^3$ with the coordinate system located in the middle of the cube. In Fig. 1, this cube is displayed with the mesh of refinement level 1 and the table of all used meshes is given aside. These meshes are created by bisecting the cathetus of the coarser



(a) Unit cube (level 1, $h = 0.5 \text{ m}$)

level	nodes	elements	h	Δt
1	50	96	0.5 m	0.3 s
2	194	384	0.25 m	0.15 s
3	770	1536	0.125 m	0.075 s
4	3074	6144	0.0625 m	0.0375 s
5	12290	24576	0.03125 m	0.01875 s

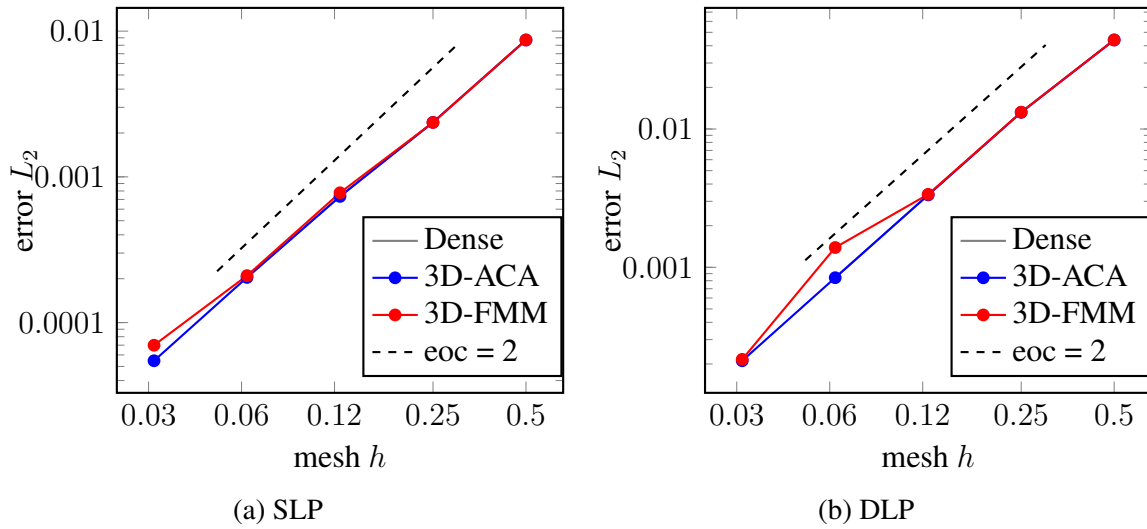
(b) Used meshes

Figure 1: Unit cube: Geometry and discretisation parameters

mesh. As load a smooth pulse $u(\mathbf{y}, t) = \frac{(t - \frac{r}{c})^2}{r} e^{-c(t - \frac{r}{c})} H(t - \frac{r}{c})$ with $r = \|\mathbf{x} - \mathbf{y}\|$ at the excitation point $\mathbf{x} = (0.8, 0.2, 0.3)^T$ is used. The total observation time $T = 3 \text{ s}$ is selected such that the smooth pulse travels over the whole unit cube. The acronym 3D-ACA will be used in the following figures for the ACA based version and 3D-FMM for the FMM based version. To show that both does not spoil the results the error in space and time it is measured at internal points as a pointwise L_2 -error. The convergence rate is denoted by $\text{eoc} = \log_2(L_2^h / L_2^{h+1})$, where the superscript h and $h + 1$ denotes two subsequent refinement level.

In Fig. 2, this error is plotted for the different refinement levels using both approaches from the above section. It can be observed that the error of a dense computation, i.e., without the proposed approximation, is mostly preserved by the 3D-ACA and the 3D-FMM. The outlier in the 3D-FMM results can be improved by finding better parameters.

Fig. 3 shows the obtained compression, i.e., the relation of the used storage compared to the

Figure 2: L_2 -error versus refinement in space and time

necessary storage of a dense computation. It is obvious that a tremendous reduction in storage can be achieved, where the FMM based version is superior for larger meshes. Unfortunately, the computing time is actually not convincing. In Fig. 4 the CPU time is displayed for establishing the 3-D data array and for the convolution. The time to establish the data array is good, however, the time for the convolution is dramatically increasing, especially for the FMM based method. The reason can be found in (16), where the computing of the tensor product to establish the data sparse structure is visible. The inner summation over the frequencies seems to spoil the performance.

Acknowledgments: Financial support by the joint DFG/FWF Collaborative Research Centre CREATOR (CRC – TRR361/F90) is gratefully acknowledged.

REFERENCES

- [1] Aimi, A.; Diligenti, M.: A new Space-Time Energetic Formulation for Wave Propagation Analysis in Layered Media by BEMs. *Int. J. Numer. Methods. Engrg.*, **75**(9), 1102–1132, 2008.
- [2] Aimi, A.; Diligenti, M.; Frangi, A.; Guardasoni, C.: A Stable 3D Energetic Galerkin BEM Approach for Wave Propagation Interior Problems. *Eng. Anal. Bound. Elem.*, **36**(12), 1756–1765, 2012.
- [3] Bamberger, A.; Ha-Duong, T.: Formulation Variationnelle Espace-temps pour le Calcul par Potentiel Retardé d’une onde Acoustique. *Math. Meth. Appl. Sci.*, **8**, 405–435 and 598–608, 1986.

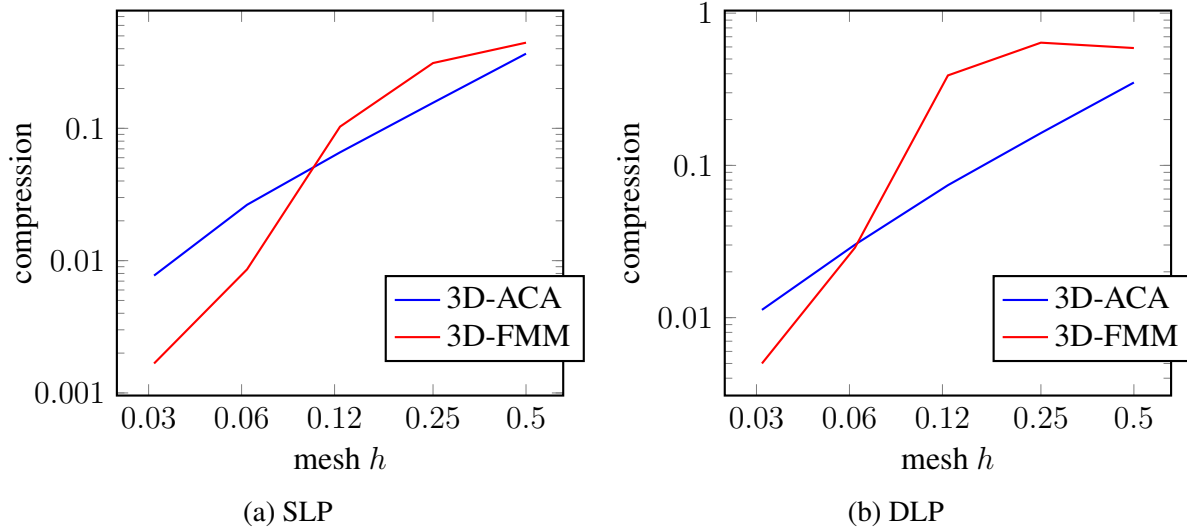


Figure 3: Compression (ratio storage 3D-ACA/dense) versus refinement in space and time

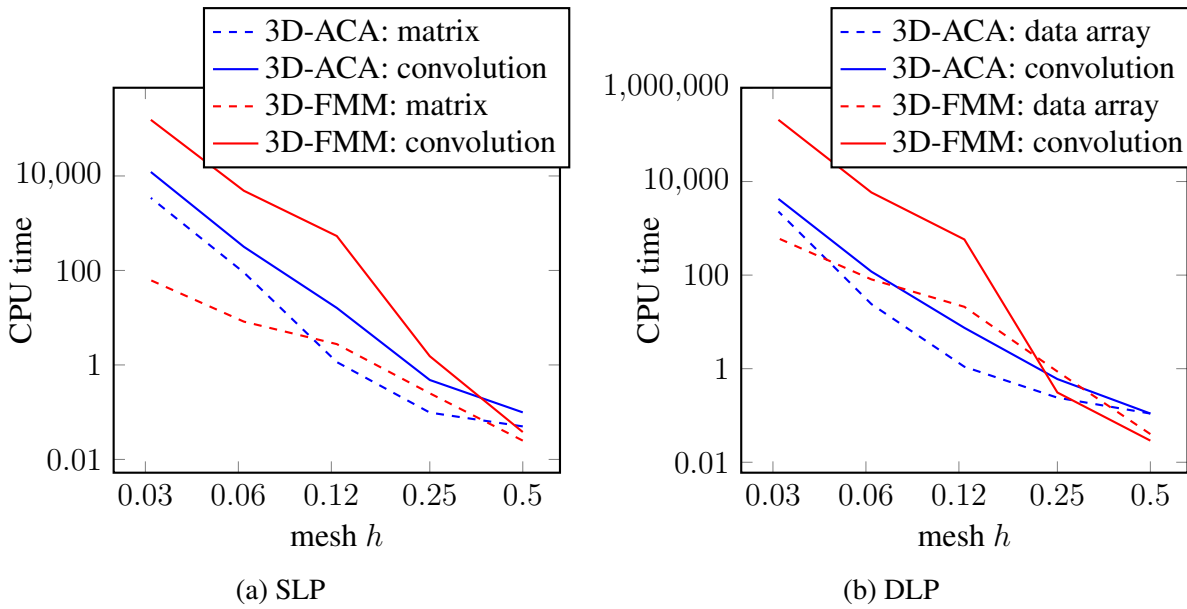


Figure 4: Computing time versus refinement in space and time

- [4] Banjai, L.; Kachanovska, M.: Fast Convolution Quadrature for the Wave Equation in Three Dimensions. *J. Comput. Phys.*, **279**, 103–126, 2014.
- [5] Bebendorf, M.: *Hierarchical Matrices: A Means to Efficiently Solve Elliptic Boundary Value Problems*, Vol. 63, *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, 2008.
- [6] Bebendorf, M.: Adaptive Cross Approximation of Multivariate Functions. *Constr. Approx.*, **34**(2), 149–179, 2011.
- [7] Bebendorf, M.; Kühnemund, A.; Rjasanow, S.: An Equi-Directional Generalization of Adaptive Cross Approximation for Higher-Order Tensors. *Appl. Num. Math.*, **74**, 1–16, 2013.
- [8] Bebendorf, M.; Rjasanow, S.: Adaptive Low-Rank Approximation of Collocation Matrices. *Computing*, **70**, 1–24, 2003.
- [9] Cruse, T. A.; Rizzo, F. J.: A Direct Formulation and Numerical Solution of the General Transient Elastodynamic Problem, I. *Aust. J. Math. Anal. Appl.*, **22**(1), 244–259, 1968.
- [10] Duffy, M. G.: Quadrature over a Pyramid or Cube of Integrands with a Singularity at a Vertex. *SIAM J. Numer. Anal.*, **19**(6), 1260–1262, 1982.
- [11] Ergin, A. A.; Shanker, B.; Michielssen, E.: Fast Evaluation of Three-Dimensional Transient Wave Fields Using Diagonal Translation Operators. *J. Comput. Phys.*, **146**(1), 157–180, 1998.
- [12] Fong, W.; Darve, E.: The Black-Box fast Multipole Method. *J. Comput. Phys.*, **228**(23), 8712–8725, 2009.
- [13] Greengard, L.; Rokhlin, V.: A new Version of the Fast Multipole Method for the Laplace equation in Three Dimensions. *Acta Num.*, **6**, 229–269, 1997.
- [14] Haider, Anita M.; Rjasanow, Sergej; Schanz, Martin: Generalised Adaptive Cross Approximation for Convolution Quadrature based Boundary Element Formulation. *arXiv*, 2024.
- [15] Leitner, Michael; Schanz, Martin: Generalized Convolution Quadrature based Boundary Element Method for Uncoupled Thermoelasticity. *Mech Syst Signal Pr*, **150**, 107234, 2021.
- [16] Lopez-Fernandez, M.; Sauter, S.: Generalized Convolution Quadrature with Variable Time Stepping. *IMA J. of Numer. Anal.*, **33**(4), 1156–1175, 2013.

- [17] Lopez-Fernandez, M.; Sauter, S.: Generalized Convolution Quadrature with Variable Time Stepping. Part II: Algorithm and Numerical Results. *Appl. Num. Math.*, **94**, 88–105, 2015.
- [18] López-Fernández, María; Sauter, Stefan: Generalized Convolution Quadrature based on Runge-Kutta Methods. *Numer. Math.*, **133**(4), 743–779, 2016.
- [19] Lubich, C.: Convolution Quadrature and Discretized Operational Calculus. I. *Numer. Math.*, **52**(2), 129–145, 1988.
- [20] Lubich, C.: Convolution Quadrature and Discretized Operational Calculus. II. *Numer. Math.*, **52**(4), 413–425, 1988.
- [21] Mansur, W. J.: *A Time-Stepping Technique to Solve Wave Propagation Problems Using the Boundary Element Method*. Phd thesis, University of Southampton, 1983.
- [22] Messner, M.; Schanz, M.: An Accelerated Symmetric Time-Domain Boundary Element Formulation for Elasticity. *Eng. Anal. Bound. Elem.*, **34**(11), 944–955, 2010.
- [23] Peirce, A.; Siebrits, E.: Stability Analysis and Design of Time-Stepping Schemes for General Elastodynamic Boundary Element Models. *Int. J. Numer. Methods. Engrg.*, **40**(2), 319–342, 1997.
- [24] Sauter, S.A.; Schanz, M.: Convolution Quadrature for the Wave Equation with Impedance Boundary Conditions. *J. Comput. Phys.*, **334**, 442–459, 2017.
- [25] Sayas, F.-J.: *Retarded Potentials and Time Domain Boundary Integral Equations: A Road Map*, Vol. 50, *Springer Series in Computational Mathematics*. Springer, Cham, 2016.
- [26] Schanz, M.: *Wave Propagation in Viscoelastic and Poroelastic Continua: A Boundary Element Approach*, Vol. 2, *Lecture Notes in Applied Mechanics*. Springer-Verlag, Berlin, Heidelberg, New York, 2001.
- [27] Schanz, M.; Antes, H.: A New Visco- and Elastodynamic Time Domain Boundary Element Formulation. *Comput. Mech.*, **20**(5), 452–459, 1997.
- [28] Schanz, Martin: Realizations of the Generalized Adaptive Cross Approximation in an Acoustic Time Domain Boundary Element Method. *Proc. Appl. Math. Mech.*, **23**(2), e202300024, 2023.
- [29] Seibel, Daniel: Boundary Element Methods for the Wave Equation based on Hierarchical Matrices and Adaptive Cross Approximation. *Numer. Math.*, **150**(2), 629–670, 2022.