3D-ACA for boundary element methods in time domain

Anita M. Haider[1], Martin Schanz[2]

[1] Institute of Applied Mechanics, 8010 Graz, Austria, Email: anita.haider@tugraz.at

[2] Institute of Applied Mechanics, 8010 Graz, Austria, Email: m.schanz@tugraz.at

Abstract

The acoustic wave equation is solved in time domain with a boundary element formulation. The time discretisation is performed with the convolution quadrature method and for the spatial approximation standard elements and a collocation schema is applied. In the interest of increasing the efficiency of the boundary element method a low-rank approximation such as the adaptive cross approximation is carried out. We discuss about a generalization of the adaptive cross approximation to approximate a three-dimensional array of data, i.e. usual boundary element matrices at several complex frequencies. This approximation is used within the generalized convolution quadrature method to obtain a real time domain method. First results are presented for some simple problems.

Introduction

The boundary element method (BEM) is in acoustics one of the preferred method, especially when scattering problems are under study. In this case also time domain formulations are important. The work of Mansur [1] may be claimed as the first time domain formulation with numerical results. Based on this approach a lot of formulations and applications can be found in the literature, which will not be recalled here. An important step forward in the application of BE formulations has been the introduction of so-called fast methods, e.g., fast multipole methods (FMM), hierarchical matrices with adaptive cross approximation (ACA), etc. In acoustics an FMM based fast time domain method has been introduced by Michielsen and co-workers [2], which might be one of the fastest implementations. A different approach to formulate a time domain BEM is based on the convolution quadrature method (CQ) proposed by Lubich [3]. This methodology results, essentially, in a true time stepping method utilizing BE matrices in Laplace domain at distinct complex frequencies. Its extension to a variable time step size is called generalized convolution quadrature method (gCQ) [4], which will be used here as basic formulation. An application of this methodology in acoustics with absorbing boundary conditions can be found in [5]. The acceleration of a usual CQ based time domain BE formulation with ACA has been published in [6]. In this application of ACA to the time domain BEM only each matrix in each time(frequency) step has been compressed, resulting in a fast method. However, still all complex frequencies have to be considered. Here, the extension of ACA to a 3D data array, the so-called 3D-ACA [7], will be applied to a gCQ based BE formulation in acoustics, which allows beside the compression within each frequency as well a reduction of the used complex frequencies of the gCQ.

Problem setting

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

$$\frac{\partial^2}{\partial t^2} u(\mathbf{x},t) - c^2 \Delta u(\mathbf{x},t) = 0 \qquad (\mathbf{x},t) \in \Omega \times (0,T) \quad (1)$$

$$u(\mathbf{x},0) = \frac{\partial}{\partial t} u(\mathbf{x},0) = 0 \qquad \mathbf{x} \in \Omega \quad (2)$$

$$u(\mathbf{x},t) = g(\mathbf{x},t) \quad (\mathbf{x},t) \in \Gamma \times (0,T) \quad (3)$$

with the wave speed c and the end time $T > 0$. Exemplarily, a Dirichlet problem is used, where $g(\mathbf{x},t)$ denotes the given data. An extension of the subsequent proposed method to Neumann problems or, more important, mixed problems is straight forward. As well exemplarily, a single layer approach is used for the presentation of the idea. This results in the well known boundary integral equation

$$u(\mathbf{x},t) = \int_0^t \int_\Gamma U(\mathbf{x},\mathbf{y},t-\tau)\phi(\mathbf{y},\tau)\,\mathrm{d}s_{\mathbf{y}}\mathrm{d}\tau \qquad (\mathbf{x},t) \in \Omega \times (0,T) \quad (4)$$

where $U(\mathbf{x},\mathbf{y},t-\tau)$ denotes the fundamental solution of the governing equation (1). The unknown density function $\phi(\mathbf{y},\tau)$ is as usual determined by

$$\int_0^t \int_\Gamma U(\mathbf{x},\mathbf{y},t-\tau)\,\phi(\mathbf{y},\tau)\,\mathrm{d}s_{\mathbf{y}}\mathrm{d}\tau = (\mathcal{V} * \phi)(\mathbf{x},t) = g(\mathbf{x},t) \quad (\mathbf{x},t) \in \Gamma \times (0,T). \quad (5)$$

The spatial discretisation is performed in a standard setting. The boundary is panellised by introducing elements $\Gamma = \bigcup_i^N \Gamma_i$. The unknown density function ϕ is approximated with constant shape functions. The given boundary data are approximated in the same way and collocation is used to set up the equation system.

Discretising the time in N not necessarily constant time steps Δt_i the gCQ [4, 8] can be applied

$$\hat{\mathcal{V}}\left(\frac{1}{\Delta t_n}\right)\phi_n + \frac{1}{2\pi \mathrm{i}} \int_C \hat{\mathcal{V}}(s)\,\frac{x_{n-1}(s)}{1-\Delta t_n s}\,\mathrm{d}\,s = g_n\,. \quad (6)$$

$()_n$ denotes the discrete value of the respective function at t_n and $\hat{\mathcal{V}}$ indicates that the Laplace transform of the integral kernel is used in the single layer operator. Further, x_n includes the results from the last time steps. The whole algorithm can be given as [9]

- Choose an A-stable Runge-Kutta method given by $\mathbf{A}$ and $\mathbf{e}_s = (0,0,\ldots,1)^T, \mathbf{1} = (1,1,\ldots,1)$

- First step

$$\hat{\mathcal{V}}\left((\Delta t_1 \mathbf{A})^{-1}\right)\boldsymbol{\phi}_1 = \mathbf{g}_1$$

  with implicit assumption of zero initial condition

- For all steps $n = 2, \ldots, N$ the algorithm has two steps

  1. Update the solution vector $\mathbf{x}_{n-1}$ at all integration points s_k

$$\mathbf{x}_{n-1}(s_k) = (\mathbf{I} - \Delta t_{n-1} s_k \mathbf{A})^{-1}\left[(\mathbf{e}_s \cdot \mathbf{x}_{n-2}(s_k))\,\mathbf{1} + \Delta t_{n-1}\mathbf{A}\boldsymbol{\phi}_{n-1}\right]$$

     for $k = 1, \ldots, N_Q$ with the number of integration points N_Q.

  2. Compute the solution of the integral at the actual time step t_n

$$\hat{\mathcal{V}}\left((\Delta t_n \mathbf{A})^{-1}\right)\boldsymbol{\phi}_n = \mathbf{g}_n - \sum_{k=1}^{N_Q} \omega_k \hat{\mathcal{V}}(s_k)\,(\mathbf{e}_s \cdot \mathbf{x}_{n-1}(s_k))\,(\mathbf{I} - \Delta t_n s_k \mathbf{A})^{-1}\,\mathbf{1}$$

where the essential parameter is $N_Q = N \log^2(N)$ and the integration points s_k are dependent on the eigenvalues of $\mathbf{A}$ and $\Delta t_{max}, \Delta t_{min}$. Several other parameters are hidden and can be found in [9, 10]. Essentially, this algorithm requires the evaluation of the integral kernel at N_Q points s_k, 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 we want to approximate by a data-sparse representation.

Generalization of the adaptive cross approximation

The idea of a generalization of adaptive cross approximation (3D-ACA) has been proposed in [7] and is sketched in Algorithm 1. As discussed above, we generate a three-dimensional array of data $\mathcal{C}_{ij,k}$. The first two indices corresponds to the spatial discretization. One is related to the collocation point and the other to the basis function. The third index of the 3D array corresponds to the complex frequencies from the gCQ. The algorithm starts by assembling the system matrix at an arbitrary chosen but fixed frequency. Let the corresponding index be k_ℓ. This system matrix, which we call face $\mathcal{H}$, is compressed by the standard adaptive cross approximation. Therefore, we have to decompose the system matrix into a hierarchical scheme first. This hierarchical structure is based on the usual geometrical considerations and used for all further assembled faces. After the low-rank approximated face is determined, we have to define the position of the pivot element. In principle, the maximum entry of the matrix determines the pivot position, where h^ℓ_{ij} denotes the entries of the current face. Regarding to this position we compute the fiber $\mathcal{F}$, an array with all frequencies where the indices related to the spatial discretization are fixed, i.e. one matrix entry at all frequencies for a distinct collocation point and shape function is assembled, f^ℓ_k. In this way the first cross with an approximated face and a fiber is generated. An schematic sketch of the fiber and face can be found in Fig.1. For the next cross, the face or the fiber has to be updated. At further iterations the residual of the face and of the fiber has to be determined and based on this residual the position of the pivot element is computed. The algorithm terminates successfully if a suitable stopping criterion with a given accuracy ε is satisfied. The current approximation is defined as $\mathcal{S}^\ell = \sum_{d=1}^{\ell} \widetilde{\mathcal{H}}^d \otimes \widetilde{\mathcal{F}}^d$, where $\otimes$ denotes the outer product of a matrix with a vector.

Algorithm 1 idea generalized ACA

For $\ell = 1, 2, 3 \ldots$

1. compute face $\mathcal{H}^\ell$ via low-rank approximation $h^\ell_{ij} = \mathcal{C}_{ij,k_\ell}$
2. define pivot position $(i_\ell, j_\ell) := \arg\max_{i,j} |h^\ell_{ij}|$
3. compute fiber $\mathcal{F}^\ell$ $\quad f^\ell_k = \mathcal{C}_{i_\ell\, j_\ell, k}$

Stop if $\|\widetilde{\mathcal{H}^\ell}\|_F \|\widetilde{\mathcal{F}^\ell}\|_F \leq \varepsilon \|\mathcal{S}^\ell\|_F$

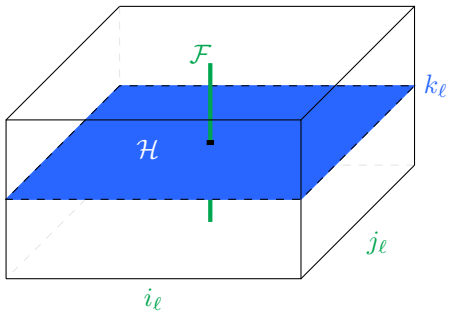

Figure 1: Sketched idea of generalized ACA (i_ℓ... collocation point, j_ℓ... basis function, k_ℓ ... frequency

The crucial task to determine the pivot position is how to find the maximum entry in a low-rank approximated face in an appropriate time for admissible blocks, i.e. ACA approximated blocks. Certainly, it is no option to compute the matrix entry of face $\mathcal{H}$ by multiplying the low-rank matrices. Instead, we define the index i or j by the maximum of the absolute row sum of the low rank matrices, respectively. To get the idea, let assume the block $\mathcal{A}$ is approximated by the truncated singular value decomposition $\mathcal{A} \approx \mathcal{A}_r = \mathbf{U}\Sigma\mathbf{V}^H$, where the low-rank matrices $\mathbf{U}$ and $\mathbf{V}$ consist of orthonormal columns. In this caes, we define the index i of the pivot position by

$$\begin{aligned} \arg\max_{i,j} |a_{ij}| &= \arg\max_{i,j} \sum_{k=1}^{r} |u_{ik}\sigma_k| \underbrace{|v_{jk}|}_{\leq 1} \\ &\leq \arg\max_{i} \sum_{k=1}^{r} |u_{ik}\sigma_k| \end{aligned} \tag{7}$$

and, analogously, for index j. This concept is based on the fact that the columns of $\mathbf{U}$ and $\mathbf{V}$ are orthonormal. To adopt this to an ACA approximated block, a QR-

decomposition of the low-rank matrix $\mathbf{U}$ and $\mathbf{V}$ is performed

$$\begin{aligned}\mathcal{A} \approx \mathcal{A}_r = \mathbf{U}\mathbf{V}^H &= Q_U(R_U R_V^H)Q_V^H \\ &= Q_U\,\check{\mathbf{U}}\check{\Sigma}\check{\mathbf{V}}^H Q_V^H = \overline{\mathbf{U}}\overline{\Sigma}\overline{\mathbf{V}}^H\end{aligned} \tag{8}$$

and, secondly, we use the SVD for the smaller inner matrix $R_U R_V^H$. Finally, we end up with low-rank matrices $\overline{\mathbf{U}}$ and $\overline{\mathbf{V}}$ where their columns are unitary. This allows to use (7) for the selection of the pivot position. Additionally, this process makes an implicit recompression of the block by the SVD of the inner matrix product.

Numerical results

How the introduced algorithm performs is shown by two numerical experiments. In the first test, solely the three-dimensional array of data $\mathcal{C}$ is computed, whereas in the second test a BE computation with the single layer potential from (5) is studied.

For the first test, the single layer operator in (6) is assembled at 11 different frequencies given by the gCQ. The underlying geometry is a unit cube with 1536 constant elements. The squared error of the approximation of $\mathcal{C}$ in the Frobenius norm is plotted in Fig. 2 against the iteration counter of the algorithm. Note, the iteration number

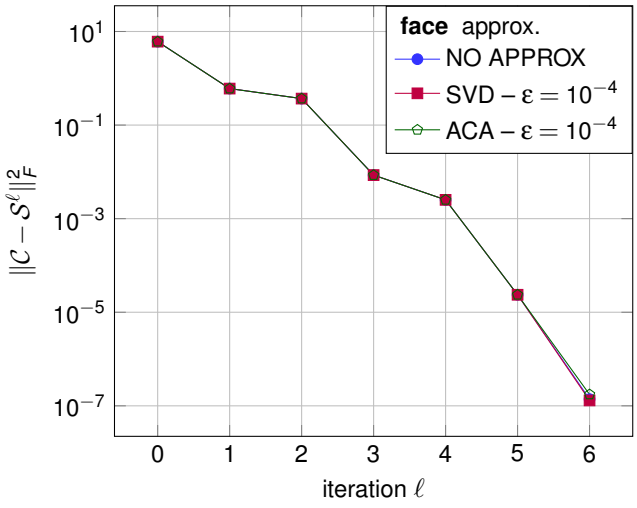

Figure 2: Approximation error in the Frobenius norm for the 3D data array $\mathcal{C}$ versus frequency

corresponds to the number of necessary complex frequencies. The accuracy of the generalized ACA is chosen as $\varepsilon = 10^{-4}$. We perform the numerical experiment first without any low-rank approximation of the face, then with an approximation by a singular value decomposition and, last, by adaptive cross approximation. The accuracy of the low-rank approximated face is set as well to $\varepsilon = 10^{-4}$ for both approximation methods. For all three alternatives, the stopping criterion is satisfied after five iterations. At the fifth iteration, the ACA approximated face exhibits a slightly different approximation error. This indicates that the low-rank approximation error of the faces has already an effect. Nevertheless, it may be concluded that it is sufficient to evaluate the single layer operator only at a few instead of all frequencies and still a sufficiently quality of the data is maintained, resulting in a reduction of the computation time and the memory consumption.

The second test shows the behavior of the proposed algorithm in a BE formulation. Oposite to the above used model problem a direct BE formulation is used. However, as well a Dirichlet problem as given in (1) is solved. To observe the behavior of the method a smooth pulse is used as load such that the analytical solution is known. Again, the geometry is a unit cube and the applied multistep method is a BDF 1. The total time is set to $T = 3.5\,\text{sec}$ and the wave speed to $c = 1\,\text{m/sec}$. Starting from an uniform discretisation with 96 triangular elements ($h = 0.5$) and 7 time steps in each refinement step the elements are halved as well as the time step size. The unknown Neumann data are approximated by constant spatial shape functions and the given Dirichlet data by linear shape functions. In Fig. 3, the L_2-error in space and time is presented. The expected linear convergence

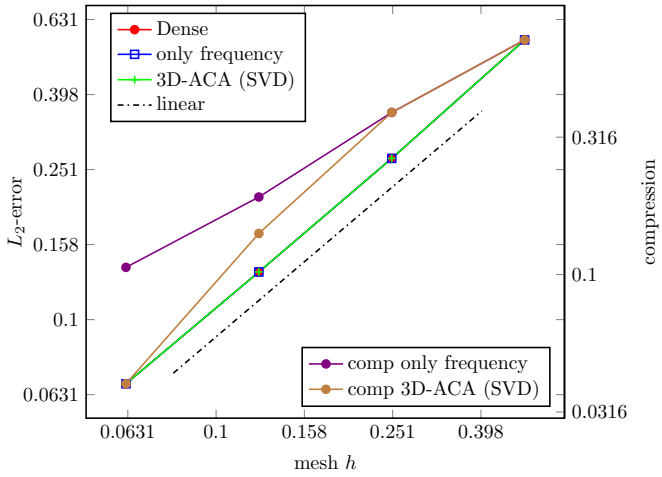

Figure 3: Convergence rate and compression for a Dirichlet problem on the unit cube

order can be observed for all three calculations, i.e. a dense calculation without any fast methods, the 3D-ACA where only the frequencies ($\mathcal{F}$ in Fig. 1) are compressed but the spatial compression is not applied, and a full 3D-ACA computation using a SVD to compress the spatial matrices $\mathcal{H}$ in Fig. 1. It may be remarked that these three results are visible only as one line in the plot. The two other lines give the compression, i.e. the relation of the required storage of a dense calculation to the compressed one. Obviously, the algorithm is quite effective as in the finest level only 4 % of the original storage is required by preserving the convergence order.

Conclusions

The proposed time domain boundary element formulation is based on the generalized convolution quadrature method. This time discretization allows the application of the generalized adaptive cross approximation (3D-ACA) to realize a data-sparse formulation. The latter is data sparse not only with respect to the spatial variable but also with respect to time. The presented results show that the method allows to identify essential

complex frequencies within a prescribed approximation quality. Overall, the storage reduction is in these examples up to 4% of the dense computation. Certainly, this value depends on the problem to be treated. Nevertheless, the formulation determines the truncation in an adaptive process without prior knowledge of the problem or experience of the user.

References

[1] Mansur, W. J.: A Time-Stepping Technique to Solve Wave Propagation Problems Using the Boundary Element Method. Dissertation. University of Southampton, 1983.

[2] Ergin, A. A., Shanker, B. and Michielssen, E.: Fast Evaluation of Three-Dimensional Transient Wave Fields Using Diagonal Translation Operators. J. Comput. Phys. 146 (1998), 157-180.

[3] Lubich, C.: On the Multistep Time Discretization of Linear Initial-Boundary Value Problems and their Boundary Integral Equations'. Numer. Math. 67 (1994), 365-–389.

[4] Lopeź-Fernandez, M. and Sauter, S.: Generalized convolution quadrature with variable time stepping. IMA J. of Numer. Anal., 33 (2013), 1156–1175.

[5] Sauter, S. A. and Schanz, M.: Convolution Quadrature for the Wave Equation with Impedance Boundary Conditions. J. Comput. Phys. 334 (2017), 442-–459.

[6] Messner, M. and Schanz, M.: An Accelerated Symmetric Time-Domain Boundary Element Formulation for Elasticity. Eng. Anal. Bound. Elem. 34 (2010), 944—955.

[7] Bebendorf, M., Kühnemund, A. and Rjasanow, S.: An equi-directional generalization of adaptive cross approximation for higher-order tensors. Appl. Numer. Math. 74 (2013), 1–16 .

[8] Lopeź-Fernandez, M. and Sauter, S. A.: Generalized convolution quadrature based on Runge-Kutta methods. Numer. Math., 133 (2016), 743–779.

[9] Lopeź-Fernandez, M. and Sauter, S.: Generalized convolution quadrature with variable time stepping. Part II: Algorithm and numerical results. Appl. Num. Math., 94 (2015), 88–105.

[10] Leitner, M. and Schanz, M.: Generalized convolution quadrature based boundary element method for uncoupled thermoelasticity. Mech. Syst. Signal. Pr., 150 (2021), 107234.