Institute of
Applied Mechanics
Institut für Baumechanik

Graz University of Technology

Preprint Series

Institute of Applied Mechanics

Graz University of Technology

Preprint No 1/2010

An accelerated symmetric time-domain boundary element formulation for elasticity

Matthias Messner, Martin Schanz
Institute of Applied Mechanics, Graz University of Technology

Published in: *Engineering Analysis with Boundary Elements*, **34**(11), 944–955, 2010
Doi: 10.1016/j.enganabound.2010.06.007
Latest revision: April 22, 2010

Abstract

Wave propagation phenomena occur often in semi-infinite regions. It is well known that such problems can be handled well with the Boundary Element Method (BEM). However, it is also known that the BEM, with its dense matrices, becomes prohibitive with respect to storage and computing time. Focusing on wave propagation problems, where a formulation in time domain is preferable, the mentioned limit of the method becomes evident. Several approaches, amongst them the Adaptive Cross Approximation (ACA), have been developed in order to overcome these drawbacks mainly for elliptic problems.

The present work focuses on time dependent elastic problems, which are indeed not elliptic. The application of the presented fast boundary element formulation on such problems is enabled by introducing the well known Convolution Quadrature Method (CQM) as time stepping scheme. Thus, the solution of the time dependent problem ends up in the solution of a system of decoupled Laplace domain problems. This detour is worth since the resulting problems are again elliptic and, therefore, the ACA can be used in its standard fashion.

The main advantage of this approach of accelerating a time dependent BEM is that it can be easily applied to other fundamental solutions as, e.g., visco- or poroelasticity.

1 Introduction

The boundary element method (BEM) is suited to treat wave propagation problems. In the present work, this method is applied on the numerical solution of initial boundary value problems in 3-d elastodynamics. For problems of rather restricted size this is done in numerous works. The first boundary integral formulation for elastodynamics in Laplace, respectively Fourier domain with a subsequent inverse transformation has been published by Cruse and Rizzo [20] and Domínguez [21]. Alternatively, the first formulation in time domain has been developed by Mansur [37]. A detailed review on elastodynamic boundary element formulations can be found in the articles of Beskos [12, 13], Chudinovich [17], and Costabel [18]. However, in order to obtain reasonable results both classes of approaches, Laplace domain with subsequent inverse transformation as well as time domain, depend strongly on a proper choice of parameters. Beside these approaches, the Convolution Quadrature Method (CQM), a more stable time stepping procedure, was proposed by Lubich [34, 35]. The method uses the Laplace domain fundamental solution. This is essential in the case of visco- and poroelasticity (see [46, 44]) since their fundamental solutions are available in closed form only in Laplace or Fourier domain. In the context of fast BE formulations this method is used in the work of Hackbusch et al. [32].

A reformulated CQM was published by Banjai and Sauter [4] and has been extended to mixed problems by Schanz [45]. The proposed reformulation transfers the time stepping procedure to the solution of decoupled Laplace domain problems. This approach is adopted for the time discretization in the present work. An improved version can be found in [2].

For the space discretization, here, the symmetric Galerkin BEM, as presented in the work of Kielhorn and Schanz [33], is applied. In the work of Blázquez et al. [14] some comparative studies of this method with respect to other formulations are presented. Nowadays, due to improved capabilities of computer systems, larger and larger problems can be solved. However, by just increasing the size of problems, the effort of solving dense matrices scales quadratically. Thus, even though better computer hardware exists the BEM reaches its limits. Hence, in the last two decades fast methods have become popular in the field of applied mathematics and engineering. The history of such methods, i.e., asymptotically optimal approximations of dense matrices, starts with the paper by Rokhlin [42]. For the first time an algorithm was presented which scales like $\mathcal{O}(n \log n)$. Subsequently, the so called Fast Multipole Method (FMM) has been developed in [26, 16] for some large-scale n-body problems. The method was significantly improved in [27]. In the work of Of et al. [39] the FMM is applied to elastostatic problems based on a Galerkin BEM discretization. The extension to elastodynamics in Fourier domain has been published in [15] based on a collocation approach. In time domain, the FMM with a plane wave expansion is presented in [49]. A black-box FMM approach for scalar-valued-problems has been proposed by Fong and Darve [22].

Other approaches are Panel Clustering (see [31]) and the wavelet based BEM [1]. The latter method produces sparse matrices based on orthogonal systems of wavelet like functions.

All these methodologies allow to perform the matrix-vector-multiplication with almost linear complexity. However, the only approach that allows to do all matrix operations (matrix-vector-, matrix-matrix-product, matrix-matrix-addition, matrix-inversion, LU-decomposition, etc.) with almost linear complexity are the so called $\mathcal{H}$-matrices introduced by Hackbusch [29]. They can be understood as algebraic structure reflecting a geometrically motivated partitioning into sub-

blocks. Each sub-block is classified to be either admissible or not. This block structure points out the fact that $\mathcal{H}$-matrix arithmetics is easy parallelizable [9].

After having concluded the setup of an $\mathcal{H}$-matrix, admissible blocks have to be approximated. All previously mentioned methods, such as FMM, Panel Clustering, and wavelet based methods approximate discrete integral operators in a very specific way. They deal with the analytical decomposition of integral kernels and, hence, the procedure becomes problem dependent. This fact holds also for the coding of this class of methodologies. A second class are the so called algebraic approximation methods. The Singular Value Decomposition (SVD) leads to the optimal approximation, however, with $\mathcal{O}(n^3)$ complexity. Less expensive algorithms are the Mosaic Skeleton Method developed in [24] and the successively developed Adaptive Cross Approximation (ACA). The latter one is chosen in the present work. It has been applied by Bebendorf [5], Bebendorf and Rjasanow [10] to the approximation of BEM matrices for the first time. The outstanding feature of ACA compared to SVD is that it requires only the evaluation of some original matrix entries and the approximation is still almost optimal. Due to this fact, it can be used in a black-box-like manner. Its coding and adaptation to existing codes is straight forward. The algorithm is robust and it is based on a stopping criterion depending on a prescribed approximation accuracy ε. Here only some, focusing on elasticity problems will be pointed out. In elasticity, Bebendorf and Grzibovski [8] used the ACA for the solution of mixed elastostatic boundary value problems. In that work an error estimate for approximated Galerkin matrices is presented. Furthermore, an improved pivoting strategy is given, such that the ACA algorithm will not fail in some special cases. An engineering approach for the acceleration of elastostatic problems is presented in the recent work by Maerten [36]. In [11] and [28], the ACA in combination with $\mathcal{H}$-matrices is efficiently applied to crack problems in elastic media solved by using a collocation boundary element formulation. The first publication studies the behaviour of a single penny shaped crack, whereas the second one solves large crack systems.

To resume, in the paper at hand, the symmetric Galerkin BEM will be used together with the reformulated CQM in order to accelerate the solution of elastodynamic boundary value problems in time domain. The latter mentioned time discretization is essential as will be illustrated in the following. Finally, the present formulation will be validated by numerical examples.

Throughout this paper, vectors and tensors are denoted by bold symbols and matrices and vectors of the discretised system by upper case and lower case sans serif symbols, respectively. No summation convention is used in the entire work. The indices of a matrix $(\mathsf{A})_{ij}$ indicate the ij-th entry, which is a scalar in the case of scalar-valued-problems. However, in the case of vector-valued-problems an entry is meant to be matrix-valued.

2 Symmetric Boundary Element Formulation

2.1 Problem formulation

In an elastic body $\Omega \subset \mathbb{R}^3$ with a Lipschitz boundary $\Gamma = \Gamma_D \cup \Gamma_N$ and a fixed final time $T \in \mathbb{R}^+$ the following mixed initial boundary value problem has to be solved

$$\begin{aligned} -(\lambda+\mu)\nabla\nabla\cdot\mathbf{u}(\tilde{\mathbf{x}},t) - \mu\Delta\mathbf{u}(\tilde{\mathbf{x}},t) + \rho\frac{\partial^2\mathbf{u}}{\partial t^2}(\tilde{\mathbf{x}},t) = \mathbf{0} \quad & (\tilde{\mathbf{x}},t) \in \Omega\times(0,T) \\ \mathbf{u}(\mathbf{x},t) = \mathbf{g}_D(\mathbf{x},t) \quad & (\mathbf{x},t) \in \Gamma_D\times(0,T) \\ \mathbf{t}(\mathbf{x},t) := \mathcal{T}_{\mathbf{x}}\mathbf{u}(\mathbf{x},t) = \mathbf{g}_N(\mathbf{x},t) \quad & (\mathbf{x},t) \in \Gamma_N\times(0,T) \\ \mathbf{u}(\tilde{\mathbf{x}},0) = \frac{\partial\mathbf{u}}{\partial t}(\tilde{\mathbf{x}},0) = \mathbf{0} \quad & (\tilde{\mathbf{x}},t) \in \Omega\times(0). \end{aligned} \tag{1}$$

The surface displacements $\mathbf{u}(\mathbf{x},t)$ and tractions $\mathbf{t}(\mathbf{x},t)$ are prescribed by some given data $\mathbf{g}_D(\mathbf{x},t)$ on Γ_D and $\mathbf{g}_N(\mathbf{x},t)$ on Γ_N, respectively. The traction operator $\mathcal{T}_{\mathbf{x}}$ reads as

$$(\mathcal{T}_{\mathbf{x}}\mathbf{u})(\mathbf{x},t) = (\sigma\cdot\mathbf{n})(\mathbf{x},t) \tag{2}$$

with the stress tensor $\sigma(\mathbf{x},t)$ incorporating Hooke's law and the outward normal vector $\mathbf{n}(\mathbf{x})$ on the boundary Γ. The Lamé constants μ and λ are connected to the modulus of elasticity E and Poisson's ration ν

$$\mu = \frac{E}{2(1+\nu)}, \quad \lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}, \tag{3}$$

whose physical significance is more immediate.

2.2 Boundary integrals

For a given time $t \in (0,T)$ the displacement field $\mathbf{u}(\tilde{\mathbf{x}},t)$ at any interior point $\tilde{\mathbf{x}} \in \Omega$ is given by the Somigliana identity (e.g. [20, 21])

$$\begin{aligned} \mathbf{u}(\tilde{\mathbf{x}},t) = &\int_0^t\int_\Gamma \mathbf{U}(\mathbf{y}-\tilde{\mathbf{x}},t-\tau)\cdot\mathbf{t}(\mathbf{y},\tau)\ \mathrm{d}s_{\mathbf{y}}\,\mathrm{d}\tau \\ &- \int_0^t\int_\Gamma (\mathcal{T}_{\mathbf{y}}\mathbf{U})(\mathbf{y}-\tilde{\mathbf{x}},t-\tau)\cdot\mathbf{u}(\mathbf{y},\tau)\ \mathrm{d}s_{\mathbf{y}}\,\mathrm{d}\tau \qquad \tilde{\mathbf{x}}\in\Omega,\ \mathbf{y}\in\Gamma \end{aligned} \tag{4}$$

with the fundamental solution $\mathbf{U}(\mathbf{y}-\tilde{\mathbf{x}},t-\tau)$. In order to obtain a symmetric formulation the traction integral equation is needed in addition to (4). This equation is obtained by applying the traction operator $\mathcal{T}_{\tilde{\mathbf{x}}}$ on (4). Next, a limiting process $\Omega \ni \tilde{\mathbf{x}} \to \mathbf{x} \in \Gamma$ is performed on both equations

$$\mathbf{u}(\mathbf{x},t) = \lim_{\Omega\ni\tilde{\mathbf{x}}\to\mathbf{x}\in\Gamma}\mathbf{u}(\tilde{\mathbf{x}},t) \qquad \mathbf{t}(\mathbf{x},t) = \lim_{\Omega\ni\tilde{\mathbf{x}}\to\mathbf{x}\in\Gamma}\mathcal{T}_{\tilde{\mathbf{x}}}\mathbf{u}(\tilde{\mathbf{x}},t). \tag{5}$$

The first boundary integral equation for $\mathbf{u}(\mathbf{x},t)$ is used on Γ_D and the second one for $\mathbf{t}(\mathbf{x},t)$ on Γ_N. Then $\tilde{\mathbf{g}}_D(\mathbf{x},t)$, $\tilde{\mathbf{g}}_N(\mathbf{x},t) \in \Gamma$ are chosen to be arbitrary but fixed extensions to the whole boundary of $\mathbf{g}_D(\mathbf{x},t)$ and $\mathbf{g}_N(\mathbf{x},t)$. Inserting the extended decompositions of the Cauchy data

$$\mathbf{u}(\mathbf{x},t) = \tilde{\mathbf{u}}(\mathbf{x},t) + \tilde{\mathbf{g}}_D(\mathbf{x},t) \qquad \mathbf{t}(\mathbf{x},t) = \tilde{\mathbf{t}}(\mathbf{x},t) + \tilde{\mathbf{g}}_N(\mathbf{x},t) \tag{6}$$

into (5) leads to the symmetric boundary integral formulation [47, 19]

$$\begin{aligned}(\mathcal{V}*\tilde{\mathbf{t}})(\mathbf{x},t)-(\mathcal{K}*\tilde{\mathbf{u}})(\mathbf{x},t) &= -(\mathcal{V}*\tilde{\mathbf{g}}_N)(\mathbf{x},t)+\left((\tfrac{1}{2}\mathcal{I}+\mathcal{K})*\tilde{\mathbf{g}}_D\right)(\mathbf{x},t)\\ &\qquad (\mathbf{x},t)\in\Gamma_D\times(0,T),\\ (\mathcal{K}'*\tilde{\mathbf{t}})(\mathbf{x},t)+(\mathcal{D}*\tilde{\mathbf{u}})(\mathbf{x},t) &= \left((\tfrac{1}{2}\mathcal{I}-\mathcal{K}')*\tilde{\mathbf{g}}_N\right)(\mathbf{x},t)-(\mathcal{D}*\tilde{\mathbf{g}}_D)(\mathbf{x},t)\\ &\qquad (\mathbf{x},t)\in\Gamma_N\times(0,T).\end{aligned} \tag{7}$$

The introduced integral operators are the single layer $\mathcal{V}$, the double layer $\mathcal{K}$, the adjoint double layer $\mathcal{K}'$, and the hypersingular operator $\mathcal{D}$, respectively. They are defined with the convolution operator $*$

$$\begin{aligned}(\mathcal{V}*\tilde{\mathbf{t}})_\Gamma(\mathbf{x},t) &= \int_0^t\int_\Gamma \tilde{\mathbf{t}}(\mathbf{y},\tau)\mathbf{U}(\mathbf{y}-\mathbf{x},t-\tau)\,\mathrm{d}s_\mathbf{y}\,\mathrm{d}\tau\\ (\mathcal{K}*\tilde{\mathbf{u}})_\Gamma(\mathbf{x},t) &= \int_0^t \lim_{\varepsilon\to0}\int_{\mathbf{y}\in\Gamma:|\mathbf{y}-\mathbf{x}|\geq\varepsilon} \tilde{\mathbf{u}}(\mathbf{y},\tau)(\mathcal{T}_\mathbf{y}\mathbf{U})^\mathsf{T}(\mathbf{y}-\mathbf{x},t-\tau)\,\mathrm{d}s_\mathbf{y}\,\mathrm{d}\tau\\ (\mathcal{K}'*\tilde{\mathbf{t}})_\Gamma(\mathbf{x},t) &= \int_0^t \lim_{\varepsilon\to0}\int_{\mathbf{y}\in\Gamma:|\mathbf{y}-\mathbf{x}|\geq\varepsilon} \tilde{\mathbf{t}}(\mathbf{y},\tau)(\mathcal{T}_\mathbf{x}\mathbf{U})(\mathbf{y}-\mathbf{x},t-\tau)\,\mathrm{d}s_\mathbf{y}\,\mathrm{d}\tau\\ (\mathcal{D}*\tilde{\mathbf{u}})_\Gamma(\mathbf{x},t) &= -\int_0^t \lim_{\Omega\ni\tilde{\mathbf{x}}\to\mathbf{x}\in\Gamma}\mathcal{T}_{\tilde{\mathbf{x}}}\int_\Gamma \tilde{\mathbf{u}}(\mathbf{y},\tau)(\mathcal{T}_\mathbf{y}\mathbf{U})(\mathbf{y}-\mathbf{x},t-\tau)\,\mathrm{d}s_\mathbf{y}\,\mathrm{d}\tau.\end{aligned} \tag{8}$$

The last, so called hypersingular integral operator has to be understood as a finite part integral. A detailed description of these integral operators and their properties can be found in [33, 37]. For the sake of readability from now on the $\tilde{\cdot}$ symbol which denotes the extensions of the Cauchy data to the whole boundary is omitted.

2.3 Time discretization

For the time discretization the Convolution Quadrature Method (CQM) is adopted. This method has initially been developed by Lubich [34, 35] and has been applied to the boundary element method by Schanz and Antes [46]. In the following, only those parts of its theoretical framework are recalled which are necessary for the understanding of the present work. The idea is to approximate the convolution integrals in time

$$(k*f)_n \approx \sum_{j=0}^{n} \omega_{n-j}^{\Delta t}(\hat{k})\, f_j \quad \text{with} \quad f_j = f(j\Delta t). \tag{9}$$

This is done by splitting up the time interval $(0,T)$ into $N+1$ time steps of equal length Δt. Equation (9) shows the approximation of the convolution integral at a certain discrete time $t_n = n\Delta t$. The quadrature weights $\omega_j^{\Delta t}$ are defined by

$$\omega_j^{\Delta t}(\hat{k}) := \frac{R^{-j}}{N+1}\sum_{\ell=0}^{N}\hat{k}(s_\ell)\zeta^{\ell j} \quad \text{with} \quad \zeta = e^{\frac{2\pi\mathrm{i}}{N+1}} \quad \text{and} \quad s_\ell = \frac{\gamma(R\zeta^{-\ell})}{\Delta t}. \tag{10}$$

In (10), R represents the radius of a circle in the domain of analyticity of $\hat{k}(s)$. This is the Laplace transform of the original time-domain kernel function $k(t)$ with the complex argument s. In the following, s_ℓ is denoted as complex frequency, since it might be seen so. $\gamma(\zeta)$ is the quotient of the characteristic polynomials of the underlying A-stable multistep method. For the comparison of different multistep methods and a detailed derivation of the CQM see [44].

Banjai and Sauter proposed in [4] to extend the quadrature weights $\omega_j^{\Delta t}$ to be valid for negative indices $j < 0$, utilizing the fact that they have to vanish in this case due to causality. Thus, the sum in (9) can be extended to $j = N$, and after inserting (10) into (9) both sums are exchanged. Finally, a reformulated approximation of the convolution integral

$$(k * f)_n \approx \frac{R^{-n}}{N+1} \sum_{\ell=0}^{N} \hat{k}(s_\ell) \hat{f}_\ell \, \zeta^{\ell n} \quad \text{with} \quad \hat{f}_\ell = \sum_{j=0}^{N} R^j f_j \, \zeta^{-\ell j}. \tag{11}$$

is obtained. After adopting (11) for the time discretization of (7), a system of $N+1$ semi-discrete equations with $\ell \in (N+1)$

$$\begin{aligned} (\hat{\mathcal{V}}\,\hat{\mathbf{t}})(\mathbf{x}, s_\ell) - (\hat{\mathcal{K}}\,\hat{\mathbf{u}})(\mathbf{x}, s_\ell) &= \hat{\mathbf{f}}_D(\mathbf{x}, s_\ell) \quad \mathbf{x} \in \Gamma_D \\ (\hat{\mathcal{K}}'\,\hat{\mathbf{t}})(\mathbf{x}, s_\ell) + (\hat{\mathcal{D}}\,\hat{\mathbf{u}})(\mathbf{x}, s_\ell) &= \hat{\mathbf{f}}_N(\mathbf{x}, s_\ell) \quad \mathbf{x} \in \Gamma_N \end{aligned} \tag{12}$$

and the load vectors

$$\begin{aligned} \hat{\mathbf{f}}_D(\mathbf{x}, s_\ell) &= -(\hat{\mathcal{V}}\,\hat{\mathbf{g}}_N)(\mathbf{x}, s_\ell) + ((\tfrac{1}{2}\mathcal{I} + \hat{\mathcal{K}})\,\hat{\mathbf{g}}_D)(\mathbf{x}, s_\ell) \\ \hat{\mathbf{f}}_N(\mathbf{x}, s_\ell) &= ((\tfrac{1}{2}\mathcal{I} - \hat{\mathcal{K}}')\,\hat{\mathbf{g}}_N)(\mathbf{x}, s_\ell) - (\hat{\mathcal{D}}\,\hat{\mathbf{g}}_D)(\mathbf{x}, s_\ell) \end{aligned} \tag{13}$$

is obtained. The unknown Cauchy data $\mathbf{u}(\mathbf{x}, t_n)$ and $\mathbf{t}(\mathbf{x}, t_n)$ for $\mathbf{x} \in \Gamma$ are obtained by

$$\mathbf{u}(\mathbf{x}, t_n) = \frac{R^{-n}}{N+1} \sum_{\ell=0}^{N} \hat{\mathbf{u}}(\mathbf{x}, s_\ell) \, \zeta^{n\ell} \quad \text{and} \quad \mathbf{t}(\mathbf{x}, t_n) = \frac{R^{-n}}{N+1} \sum_{\ell=0}^{N} \hat{\mathbf{t}}(\mathbf{x}, s_\ell) \, \zeta^{n\ell}. \tag{14}$$

The now semi discrete operators $\hat{\mathcal{V}}$, $\hat{\mathcal{K}}$, $\hat{\mathcal{K}}'$, and $\hat{\mathcal{D}}$ are the Laplace transformed counterparts of their time-domain version containing the Laplace transformed fundamental solution $\hat{\mathbf{U}}(\mathbf{y} - \mathbf{x}, s)$ (see [33]). A more detailed description of this reformulated CQM and its application onto mixed elastodynamic problems can be found in [45].

2.4 Spatial discretization

For the spatial discretization a standard Galerkin approach is employed. The unknown $\hat{\mathbf{u}}(\mathbf{x}, s)$ and $\hat{\mathbf{t}}(\mathbf{x}, s)$ of the semi-discrete system of equations (12) are unique solutions of the variational formulation related to (7)

$$\begin{aligned} \langle \hat{\mathcal{V}}\hat{\mathbf{t}}, \mathbf{w} \rangle_{\Gamma_D} - \langle \hat{\mathcal{K}}\hat{\mathbf{u}}, \mathbf{w} \rangle_{\Gamma_D} &= \langle \hat{\mathbf{f}}_D, \mathbf{w} \rangle_{\Gamma_D} \\ \langle \hat{\mathcal{K}}'\hat{\mathbf{t}}, \mathbf{v} \rangle_{\Gamma_N} + \langle \hat{\mathcal{D}}\hat{\mathbf{u}}, \mathbf{v} \rangle_{\Gamma_N} &= \langle \hat{\mathbf{f}}_N, \mathbf{v} \rangle_{\Gamma_N} \end{aligned} \tag{15}$$

for all test functions $\mathbf{w}(\mathbf{x})$ and $\mathbf{v}(\mathbf{x})$. In order to discretize (15), a triangulation of the whole boundary Γ into P linear triangles τ_P is made

$$\Gamma \approx \Gamma_h = \bigcup_{p=1}^{P} \tau_p. \tag{16}$$

Appropriate finite dimensional subspaces for discontinuous and continuous polynomials, respectively, are defined

$$\begin{aligned} S_h^{\gamma}(\Gamma_{D,h}) &:= \operatorname{span}\{\psi_m^{\gamma}\}_{m=1}^{M_D} \qquad \gamma \in \mathbb{N} \\ S_h^{\beta}(\Gamma_{N,h}) &:= \operatorname{span}\{\varphi_m^{\beta}\}_{m=1}^{M_N} \qquad \beta \in \mathbb{N} \setminus 0. \end{aligned} \tag{17}$$

The unknown tractions and displacements $\hat{\mathsf{t}}_h, \hat{\mathsf{u}}_h$ and suitable test functions $\mathsf{w}_h, \mathsf{v}_h$ are in these subspaces

$$\hat{\mathsf{t}}_h, \mathsf{w}_h \in S_h^{\gamma}(\Gamma_{D,h}) \quad \text{and} \quad \hat{\mathsf{u}}_h, \mathsf{v}_h \in S_h^{\beta}(\Gamma_{N,h}). \tag{18}$$

The fact that $\hat{\mathsf{t}}_h$ and $\hat{\mathsf{u}}_h$ are unique solutions of (15) leads to $N+1$ linear systems of equations of the form

$$\begin{pmatrix} \hat{\mathsf{V}}_h & -\hat{\mathsf{K}}_h \\ \hat{\mathsf{K}}_h^{\mathsf{T}} & \hat{\mathsf{D}}_h \end{pmatrix} \cdot \begin{pmatrix} \hat{\mathsf{t}}_h \\ \hat{\mathsf{u}}_h \end{pmatrix} = \begin{pmatrix} \hat{\mathsf{f}}_D \\ \hat{\mathsf{f}}_N \end{pmatrix}, \tag{19}$$

each depending on s_ℓ and the matrices and vectors are

$$\begin{aligned} \hat{\mathsf{V}}_h \in \mathbb{C}^{M_D \times M_D},\ \hat{\mathsf{K}}_h \in \mathbb{C}^{M_D \times M_N},\ \hat{\mathsf{D}}_h \in \mathbb{C}^{M_N \times M_N} \\ \text{and } \hat{\mathsf{t}}_h, \hat{\mathsf{f}}_D \in \mathbb{C}^{M_D},\ \hat{\mathsf{u}}_h, \hat{\mathsf{f}}_N \in \mathbb{C}^{M_N}, \end{aligned} \tag{20}$$

respectively. M_D denotes the number of unknowns on Γ_D and M_N on Γ_N, respectively. The entries of the matrices in (19) are given by

$$\begin{aligned} (\hat{\mathsf{V}}_h)_{ij} &= \int_{\tau_i} \psi_i^{\gamma}(\mathbf{x}) \int_{\tau_j} \hat{\mathbf{U}}(\mathbf{y}-\mathbf{x}, s) \psi_j^{\gamma}(\mathbf{y}) \, \mathrm{d}s_{\mathbf{y}} \, \mathrm{d}s_{\mathbf{x}} \\ (\hat{\mathsf{K}}_h)_{ij} &= \int_{\tau_i} \psi_i^{\gamma}(\mathbf{x}) \int_{\operatorname{supp}(\varphi_j^{\beta})} (\mathcal{T}_{\mathbf{y}} \hat{\mathbf{U}})^{\mathsf{T}}(\mathbf{y}-\mathbf{x}, s)\, \varphi_j^{\beta}(\mathbf{y}) \, \mathrm{d}s_{\mathbf{y}} \, \mathrm{d}s_{\mathbf{x}} \\ (\hat{\mathsf{D}}_h)_{ij} &= -\int_{\operatorname{supp}(\varphi_i^{\beta})} \varphi_i^{\beta}(\mathbf{x}) \mathcal{T}_{\mathbf{x}} \int_{\operatorname{supp}(\varphi_j^{\beta})} (\mathcal{T}_{\mathbf{y}} \hat{\mathbf{U}})^{\mathsf{T}}(\mathbf{y}-\mathbf{x}, s)\, \varphi_j^{\beta}(\mathbf{y}) \, \mathrm{d}s_{\mathbf{y}} \, \mathrm{d}s_{\mathbf{x}}. \end{aligned} \tag{21}$$

All integrals are evaluated within their computational domains, i.e., their support. The support of the discontinuous polynomials ψ_m^{γ} is defined by a boundary element τ_p. The support of the continuous polynomials φ_m^{β} stretches to all adjacent elements. The integrals of $\hat{\mathsf{K}}_h$ and $\hat{\mathsf{D}}_h$ are of strong singular and hypersingular type, respectively. By applying the regularization proposed by Kielhorn and Schanz [33] they are transformed into weak singular integrals. For their numerical integration special coordinate transformations, also known as Duffy transformations (see [43]), are used.

3 Matrix Approximation

An approximation for a matrix $\mathsf{A} \in \mathbb{C}^{t \times s}$

$$\mathsf{A} \approx \mathsf{S}_k = \mathsf{U}\mathsf{V}^{\mathsf{T}} \quad \text{with} \quad \mathsf{U} \in \mathbb{C}^{t \times k} \text{ and } \mathsf{V} \in \mathbb{C}^{s \times k}, \tag{22}$$

with a small rank k compared to t and s can be found whenever the generating kernel function $k(\mathbf{x},\mathbf{y})$ in the computational domain of A is asymptotically smooth. As shown in [7] and [30] all kernel functions $k(\mathbf{x},\mathbf{y})$ of elliptic operators with constant coefficients and $\mathbf{x} \neq \mathbf{y}$ have this property. Only in the case $\mathbf{x} = \mathbf{y}$ they become singular and are not smooth. Beside this spatial singularity the kernel functions of the hyperbolic operators in (7) have an additional temporal singularity for $t = \tau$. By means of the CQM (see section 2.3) the hyperbolic operators are transformed into elliptic ones. This fact is essential for the application of the approximation.

3.1 Hierarchical matrices

Due to the previously described kernel properties of elliptic operators the necessity to separate the near- from the far-field becomes evident. Low rank approximations of the type (22) can be obtained only for well separated computational domains $\mathbf{x} \neq \mathbf{y}$. Thus, $\mathcal{H}$-matrices [30, 7] are used. Their setup is based on the following idea: The index sets I and J of row and column degrees of freedom are permuted in such a way that those who are far away from each other do also obtain indices with a large offset.

First, by means of a distance based hierarchical subdivision of I and J cluster trees T_I and T_J are created. In each step of this procedure a new level of son clusters is inserted into the cluster trees. A son cluster is not further subdivided and is called to be a leaf if his size reaches a prescribed minimal size b_{min}. Basically, two approaches can be distinguished. First, the subdivision based on bounding boxes splits the domain into axis-parallel boxes which contain the son clusters. Second, the subdivision based on principal component analysis splits the domain into well balanced son clusters leading to a minimal cluster tree depth. Details for both approaches can be found in [30]. In the present work, the latter approach is adopted. The former one would lead to identical results for the examples presented in section 5.

Now, the $\mathcal{H}$-matrix structure is defined by the block cluster tree $T_{I\times J} := T_I \times T_J$. Its setup is performed by means of the following admissibility criterion

$$\min(\operatorname{diam}(t), \operatorname{diam}(s)) \leq \eta \operatorname{dist}(t, s), \tag{23}$$

with the clusters $t \subset T_I$, $s \subset T_J$ and the admissibility parameter $0 < \eta < 1$. The diameter of the clusters t and s and their distance is computed as usual

$$\begin{aligned}
\operatorname{diam}(t) &= \max_{i_1, i_2 \in t} |\mathbf{x}_{i_1} - \mathbf{x}_{i_2}|, \\
\operatorname{diam}(s) &= \max_{j_1, j_2 \in s} |\mathbf{y}_{j_1} - \mathbf{y}_{j_2}|, \\
\operatorname{dist}(t, s) &= \min_{i \in t,\, j \in s} |\mathbf{x}_i - \mathbf{y}_j|.
\end{aligned}$$

Each cluster t and s is associated with its computational domain $\mathbf{x}_t$ and $\mathbf{y}_s$ on Γ. The support of the corresponding degrees of freedom of row i and column j are denoted by $\mathbf{x}_i$ and $\mathbf{y}_j$, i.e.,

$$\mathbf{x}_t := \bigcup_{i \in t} \mathbf{x}_i \quad \text{and} \quad \mathbf{y}_s := \bigcup_{j \in s} \mathbf{y}_j \quad \text{with} \quad \mathbf{x}_i, \mathbf{y}_j \in \mathbb{R}^3. \tag{24}$$

If (23) is fulfilled, a block $b = t \times s$ is admissible. If condition (23) is not fulfilled the admissibility is recursively checked for their son clusters, until either (23) holds or both clusters t

and s become leafs. In the latter case block b is not admissible. As can be seen in (21) all matrices are generated by kernel functions of the type $\hat{k}(\mathbf{y}-\mathbf{x},s)$. Admissible blocks have well separated computational domains $\mathbf{x}_t$ and $\mathbf{y}_s$ and the algorithm presented in section 3.2 is used to approximate them. Not admissible blocks must be evaluated without approximation.

3.2 Adaptive Cross Approximation

A remark to the notation in this section: $(\mathsf{A})_{ij}$ denotes the ij-th entry, whereas $(\mathsf{A})_i$ and $(\mathsf{A})_j$ are the i-th row vector and j-th column vector, respectively. The idea of the Adaptive Cross Approximation is to split up a matrix $\mathsf{A}\in\mathbb{C}^{t\times s}$ into $\mathsf{A}=\mathsf{S}_k+\mathsf{R}_k$ where S_k denotes the rank k approximation of A and R_k the residuum to be minimized. Starting from

$$\mathsf{S}_0 \coloneqq 0 \quad \text{and} \quad \mathsf{R}_0 \coloneqq \mathsf{A}, \tag{25}$$

a first pivot $\gamma_1=(\mathsf{R}_0)_{ij}^{-1}$ has to be found, where i and j are the row and column indices of the actual (0-th in this case) approximation step. Hints for the right choice of the initial pivot γ_1 can be found in [7]. In each ongoing step ν the scaled outer product of the pivot row and column is subtracted from R_ν and added to S_ν

$$\begin{aligned}\mathsf{R}_{\nu+1} &\coloneqq \mathsf{R}_\nu - \mathsf{u}_{\nu+1}\mathsf{v}_{\nu+1}^\top\\ \mathsf{S}_{\nu+1} &\coloneqq \mathsf{S}_\nu + \mathsf{u}_{\nu+1}\mathsf{v}_{\nu+1}^\top\end{aligned} \tag{26}$$

with the i-th row vector and j-th column vector defined as

$$\mathsf{v}_{\nu+1}=\gamma_{\nu+1}\,(\mathsf{R}_\nu)_i \quad \text{and} \quad \mathsf{u}_{\nu+1}=(\mathsf{R}_\nu)_j. \tag{27}$$

The residuum R_ν is minimized and the rank of the approximant S_ν is increased step by step. The pivot $\gamma_{\nu+1}$ is chosen to be the largest entry in modulus of either the row $(\mathsf{R}_\nu)_i$ or column $(\mathsf{R}_\nu)_j$. Finally, the approximation stops if the following criterion holds

$$\|\mathsf{u}_{\nu+1}\|_F\,\|\mathsf{v}_{\nu+1}\|_F < \varepsilon\|\mathsf{S}_{\nu+1}\|_F. \tag{28}$$

Note, the entire matrix A will never be generated. Therefore, special care has to be taken in order to find the pivot such that the algorithm converges to the prescribed accuracy ε [41, 7].

By using the definition of the absolute value $|z|=\sqrt{z\bar{z}}$, with $z\in\mathbb{C}$ and the complex conjugate $\bar{z}$, the Frobenius norm of the approximant $\|\mathsf{S}_{\nu+1}\|_F$ can be reformulated as

$$\begin{aligned}\|\mathsf{S}_{\nu+1}\|_F^2 &= \sum_{i=1}^{m}\sum_{j=1}^{n}\left|(\mathsf{S}_{\nu+1})_{ij}\right|^2\\ &= \sum_{i=1}^{m}\sum_{j=1}^{n}\left|(\mathsf{S}_\nu)_{ij}+(\mathsf{u}_{\nu+1})_i\,(\mathsf{v}_{\nu+1})_j\right|^2\\ &= \|\mathsf{S}_\nu\|_F^2+\sum_{\ell=1}^{\nu}\left(\mathsf{u}_\ell^\top\overline{\mathsf{u}}_{\nu+1}\mathsf{v}_\ell^\top\overline{\mathsf{v}}_{\nu+1}+\overline{\mathsf{u}}_\ell^\top\mathsf{u}_{\nu+1}\overline{\mathsf{v}}_\ell^\top\mathsf{v}_{\nu+1}\right)+\|\mathsf{u}_{\nu+1}\|_F^2\|\mathsf{v}_{\nu+1}\|_F^2.\end{aligned} \tag{29}$$

Due to this recursive representation all algebraic evaluations in (26) and (28) can be performed with $\mathcal{O}(k^2(t+s))$ complexity. Usually the computational costs for generating matrix entries dominates by far the costs needed for the ACA algorithm. Hence, the complexity and memory requirement scale like $\mathcal{O}(k(t+s))$ (see [41]).

3.3 Vector-valued problems

Before explaining how to apply the ACA to problems of a vector-valued type, the influence of the pivot value γ_v is recalled. It is responsible for the correct scaling of the outer product which is subtracted from the reminder R_v and added to the approximant S_v. A necessary condition is that its value is non-zero. It turns out that the optimal choice is to look for the largest entry in modulus of either the previously computed row or column of R_v. Based on this entry the next outer product is computed.

In the case of a scalar-valued problem the largest non zero entry can be found in a straight forward manner. However, the problem arises when dealing with vector-valued problems, e.g. 3-d elasticity problems. In this case each entry is of matrix type $A_{ij} \in \mathbb{C}^{3\times 3}$ and the pivot might be defined as

$$\gamma_v = (R_v)_{ij}^{-1} \quad \text{with} \quad \|(R_v)_{ij}\|_p = \max. \tag{30}$$

For no norm $\|\cdot\|_p$ it is guaranteed that a proper pivot entry can be found. E.g. , if $(R_v)_{ij}$ contains one very small entry compared to the remaining entries of $(R_v)_i$ the pivot row u_v is scaled up by γ_v and (28) does not hold. If $(R_v)_{ij}$ contains a zero entry the pivot γ_v is not even defined. This becomes evident, if e.g. the single layer operator in (21) is evaluated on some plane $\Gamma_k := \{\mathbf{x},\mathbf{y} \in \mathbb{R}^3 : y_k - x_k = 0\}$ that lies perpendicular to the coordinate axis k. In this case certain off diagonal entries vanish

$$\hat{U}_{pq}(\mathbf{y}-\mathbf{x},s) = 0 \quad \mathbf{x},\mathbf{y} \in \Gamma_k,\ p \vee q = k \tag{31}$$

$\hat{U}_{pq} \in \mathbb{C}^{3\times 3}$ is the matrix-valued fundamental solution for elastodynamics in Laplace domain. Due to these facts a straight forward generalization of the scalar-valued ACA to the matrix-valued version is not possible. Hence, in this work the 3-d elasticity problem is split up into 3×3 scalar-valued problems. Any matrix A_h and vector b_h in (19) becomes partitioned into sub-blocks

$$A_h = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \qquad b_h = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}. \tag{32}$$

In each of these sub-blocks the ACA is applied as presented in section 3.2. Thus, remember, sub-matrices containing only zero entries due to (31) do not even have to be stored.

4 Fast solution procedure

By using the reformulated CQM for the temporal discretization, the hyperbolic problem (1) is transformed into a system of decoupled Laplace domain problems. Based on the fact that they become elliptic the known admissibility criterion (23) for the setup of $\mathcal{H}$-matrices and the ACA can be used. These problems depend on specific complex frequencies s_ℓ. Based on their construction they always appear as complex conjugates. Due to this fact only half of the problems has to be solved. The other half results in the conjugate complex solution.

Recalling, the solution procedure starts with the transformation of all prescribed boundary conditions using (11). Then, the system of decoupled Laplace domain problems is solved and, finally, the time domain solution is obtained by using (14).

4.1 Solution of mixed problems

As described in section 2, $N+1$ positive definite and, thus, invertible equation systems of the form

$$\begin{pmatrix} \hat{\mathsf{V}}_h & -\hat{\mathsf{K}}_h \\ \hat{\mathsf{K}}_h^{\mathsf{T}} & \hat{\mathsf{D}}_h \end{pmatrix} \cdot \begin{pmatrix} \hat{\mathsf{t}}_h \\ \hat{\mathsf{u}}_h \end{pmatrix} = \begin{pmatrix} \hat{\mathsf{f}}_D \\ \hat{\mathsf{f}}_N \end{pmatrix}$$

have to be solved for each complex frequency s_ℓ. For the solution of this block skew symmetric matrix the following solution procedure (see [48]) is applied. By inserting the first equation

$$\hat{\mathsf{t}}_h = \hat{\mathsf{V}}_h^{-1} \left(\hat{\mathsf{f}}_D + \hat{\mathsf{K}}_h \hat{\mathsf{u}}_h \right) \tag{33}$$

into the second, the Schur complement system

$$\underbrace{\left(\hat{\mathsf{D}}_h + \hat{\mathsf{K}}_h^{\mathsf{T}} \hat{\mathsf{V}}_h^{-1} \hat{\mathsf{K}}_h \right)}_{\hat{\mathsf{S}}_h} \hat{\mathsf{u}}_h = \underbrace{\hat{\mathsf{f}}_N - \hat{\mathsf{K}}_h^{\mathsf{T}} \hat{\mathsf{V}}_h^{-1} \hat{\mathsf{f}}_D}_{\hat{\mathsf{y}}_h} \tag{34}$$

is obtained. The resulting Schur complement matrix is symmetric and positive definite. Instead of performing a direct inversion of $\hat{\mathsf{V}}_h$ a nested iterative solver is applied. First, the right hand side

$$\hat{\mathsf{y}}_h = \hat{\mathsf{f}}_N - \hat{\mathsf{K}}_h^{\mathsf{T}} \hat{\mathsf{c}}_h \tag{35}$$

with the solution $\hat{\mathsf{c}}_h$ of $\hat{\mathsf{V}}_h \hat{\mathsf{c}}_h = \hat{\mathsf{f}}_D$ is computed. Then, the matrix-vector-multiplication for the Schur complement system is defined as

$$\hat{\mathsf{S}}_h \hat{\mathsf{u}}_h = \hat{\mathsf{D}}_h \hat{\mathsf{u}}_h + \hat{\mathsf{K}}^{\mathsf{T}} \hat{\mathsf{b}}_h \tag{36}$$

with $\hat{\mathsf{b}}_h$ out of $\hat{\mathsf{V}}_h \hat{\mathsf{b}}_h = \hat{\mathsf{K}}_h \hat{\mathsf{u}}_h$. Note, both $\hat{\mathsf{V}}_h$ and $\hat{\mathsf{S}}_h$ are complex symmetric but not hermitian and, therefore, no conjugate gradient scheme (CG) can be used. An iterative solver capable of solving the present system has to be taken. In this work a restarted GMRES solver is used. No preconditioner is applied. It will be the topic of further investigations.

4.2 Solution of Neumann problems

The problem statement (1) reduces to prescribed boundary conditions for the Neumann part of the boundary only, i.e., the complete set of boundary conditions are

$$\mathbf{t}(\mathbf{x},t) = \mathbf{g}_N(\mathbf{x},t) \quad (\mathbf{x},t) \in \Gamma \times (0,T). \tag{37}$$

Hence, the displacements $\mathbf{u}$ are sought on the complete $\Gamma \times (0,T)$. Due to the zero initial conditions $\mathbf{u}(\cdot, t=0) = \dot{\mathbf{u}}(\cdot, t=0) = \mathbf{0}$ rigid body motions cause no problems regarding the solvability of the reduced variational formulation

$$\langle \mathcal{D} * \mathbf{u}, \mathbf{v} \rangle_\Gamma = \langle (\mathcal{I} - \mathcal{K}') * \mathbf{t}, \mathbf{v} \rangle_\Gamma. \tag{38}$$

The presented temporal and spatial discretization leads to $N+1$ decoupled equation systems of the type

$$\hat{\mathsf{D}}_h \hat{\mathsf{u}}_h = \hat{\mathsf{f}}_N, \tag{39}$$

which can be solved by taking any iterative solver capable of solving complex symmetric but not hermitian systems. Again, a not preconditioned restarted GMRES solver is used.

5 Numerical examples

In this section, the previously presented solution procedures are tested. In order to show the validity of the results only benchmark examples, whose analytical solution is known, are treated. Aspects regarding the efficiency of the presented methodology are pointed out. The most important criterion is the compression rate of matrices depending on either the approximation accuracy ε or the complex frequency (10). The compression rate is defined as the ratio between the size of the compressed matrix and the size of the dense matrix. In other words a compression rate of 50% means that only one half of the original dense matrix is computed and stored. Due to the fact that the cost of evaluating matrix entries outweighs the overhead of constructing $\mathcal{H}$-matrices, applying ACA and solving equation systems, the compression rate provides information about memory consumption and speed up.

All computations were performed by using the HyENA C++ library for the numerical solution of partial differential equations using the boundary element method [38]. The part of the library that deals with $\mathcal{H}$-matrix arithmetics and the ACA as well as the restarted GMRES solver stem form the AHMED C++ library [6]. For the Fourier like transformations in (11) and (14) the FFTW routines [23] are taken.

5.1 Longitudinal waves in a rod

A 3-d rod of size $\ell_1 = 3.0\,\mathrm{m}$ and $\ell_2 = \ell_3 = 1.0\,\mathrm{m}$, as depicted in Figure 1, is considered. It is fixed on one end and the other end is excited by a pressure jump $t_1 = -1.0H(t)\,\mathrm{N/m^2}$. $H(t)$ denotes the unit step function. The material parameters of steel ($\rho = 7850.0\,\mathrm{kg/m^3}$, $\mu = 1.055 \times 10^{11}\,\mathrm{N/m^2}$, $\lambda = 0.0\,\mathrm{N/m^2}$) are taken. Poisson ratio is chosen to be zero, such that the results can be compared with the analytical solution of longitudinal waves in a 1-d elastodynamic rod (see [25]). The

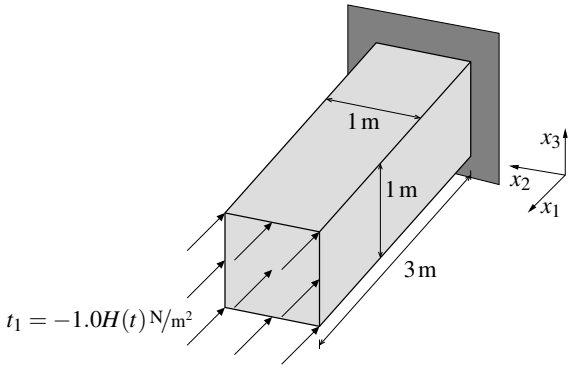


Figure 1: System and boundary conditions

rod shown in Figure 1 is discretised into 3660 triangular boundary elements of uniform mesh size $h = 0.08\,\mathrm{m}$. The displacements and tractions are approximated by piecewise constant and continuous linear polynomials, respectively

$$\mathsf{t}_h \in S_h^0(\Gamma_{D,h}) \quad \text{and} \quad \mathsf{u}_h \in S_h^1(\Gamma_{N,h}).$$

In order to compare different time discretizations the dimensionless value

$$\beta = \frac{c_1 \Delta t}{h} \tag{40}$$

is introduced. It can be referred to also as the *Courant-Friedrichs-Lewy* (CFL) number. This value depends on the velocity of the compression wave $c_1 = \sqrt{(\lambda + 2\mu)/\rho}$, the time step size Δt and the average mesh size h. The overall analyzed time is $T = 0.005\,\mathrm{s}$. Therefore, for $\beta = 0.3$ about 1000, for $\beta = 0.5$ about 600 and for $\beta = 0.7$ about 430 time steps are necessary.

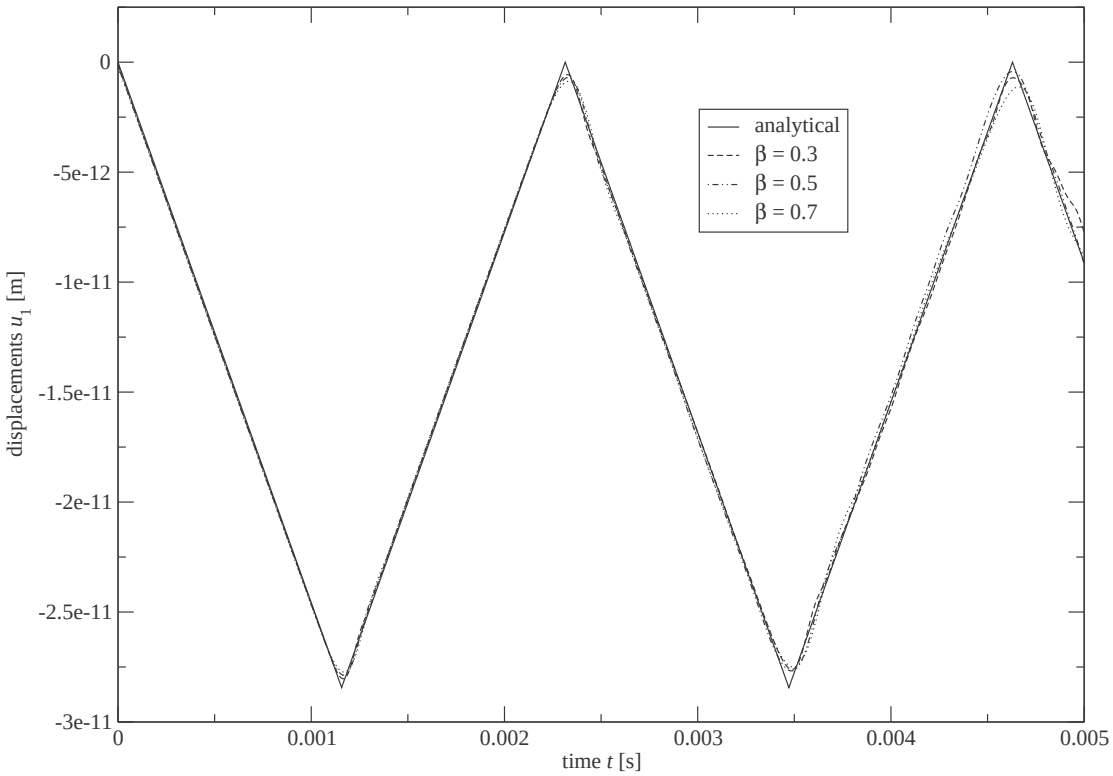


Figure 2: Longitudinal displacements u_1 at the center of the free end

Figure 2 shows the longitudinal displacements at the center of the free end and Figure 3 the normal tractions at the center of the fixed end. As reference the analytical 1-d solution is given as well. Obviously, for displacements better results can be achieved rather than for tractions. As expected, larger β values lead to more numerical damping. This becomes apparent towards the end of the plots. For both, displacements and tractions $\beta = 0.3$ leads to the best results. All numerical results shown in Figure 2 and 3 were computed with an approximation accuracy $\varepsilon = 1.0 \times 10^{-05}$ of the ACA (see (28)).

Before showing compression rates a representative part of the partitioned matrix (32) has to be chosen. From now on this will be $\hat{\mathsf{D}}_{11}$. In the present case it is of size 1680×1680 entries. Differences, regarding the compression rates, when comparing $\hat{\mathsf{D}}_{11}$ to other matrices

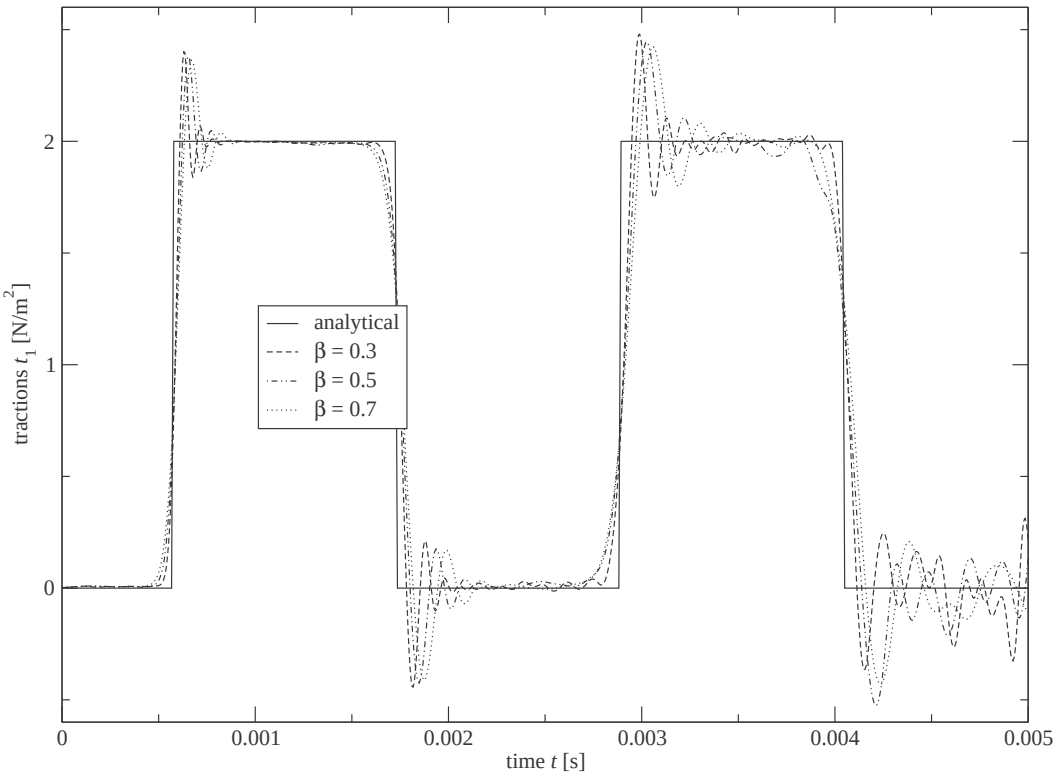
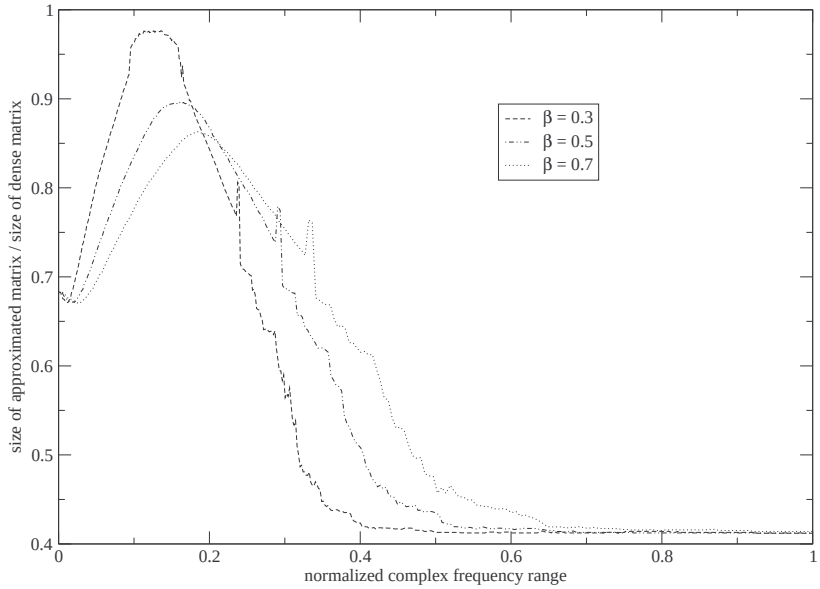


Figure 3: Normal tractions t_1 at the center of the fixed end

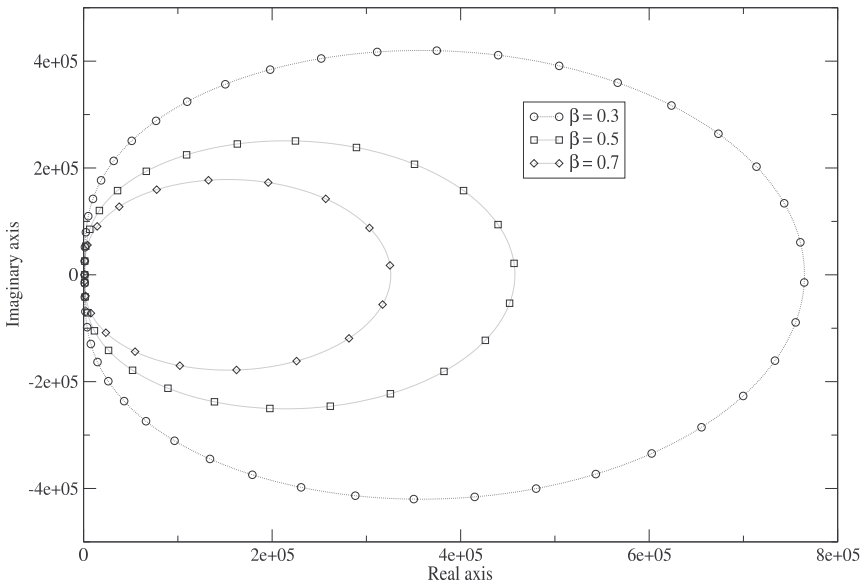
$\hat{\mathsf{D}}_{12}$, $\hat{\mathsf{D}}_{13}$, $\hat{\mathsf{D}}_{22}$, and $\hat{\mathsf{D}}_{33}$ are negligible. In Figure 4a, its compression ratios are plotted against the normalized complex frequencies s_ℓ. In Figure 4b, the respective complex frequencies are plotted in the complex plane. The bad compression ratio for frequencies with small index ℓ is the most apparent behavior. It is caused by the fact that the imaginary part of s_ℓ, which is responsible for the oscillating behavior of the kernel function $\hat{k}$, increases very fast and the real part, i.e., the damping factor, is still very small. Thus, the more the real part increases the better the compression becomes and, finally, a constant ratio is obtained. Moreover, Figure 4 shows the larger β becomes the smaller the imaginary part of the frequencies is and, hence, better compression ratios can be achieved.

Next, results for two different approximation accuracies of the ACA $\varepsilon = 1.0 \times 10^{-03}$ and $\varepsilon = 1.0 \times 10^{-05}$ are compared in Figure 5. Only tractions results are presented since they are much more sensitive, i.e., the displacement results are equal for both precisions. Even when looking at the traction results plotted in Figure 5a no evident difference is apparent. However, Figure 5b shows that for $\varepsilon = 1.0 \times 10^{-03}$ and high frequencies the compression increases as expected, whereas it converges to the same ratio in both cases for large real parts of s_ℓ.

In Figure 6, the CPU-time with and without ACA for the entire frequency range is compared. As expected, the results for the matrix assembly (see Figure 6a) reflect the compression ratios plotted Figure 5b, i.e., in the second half the compression is roughly 50 % and, hence, the computing time is roughly halved. Figure 6b shows the results for the iterative solution of the

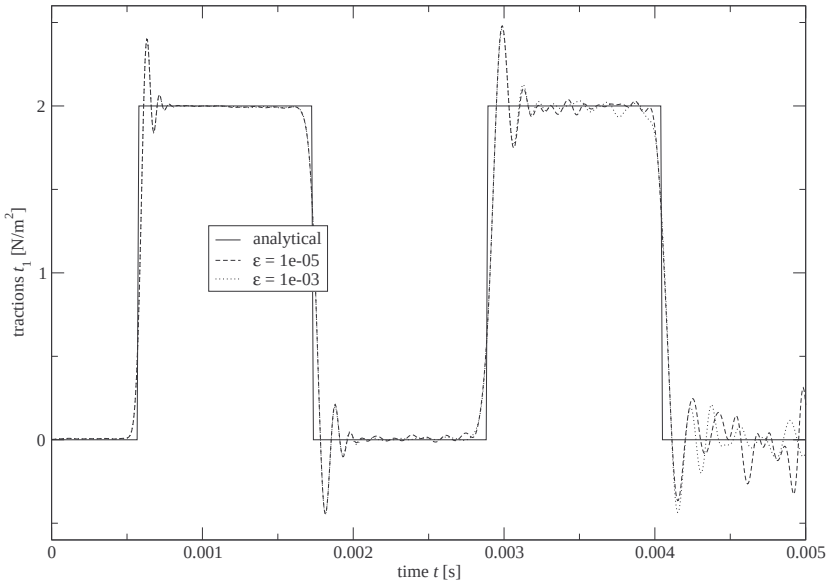


(a) Compression ratio of $\hat{\mathsf{D}}_{11}$

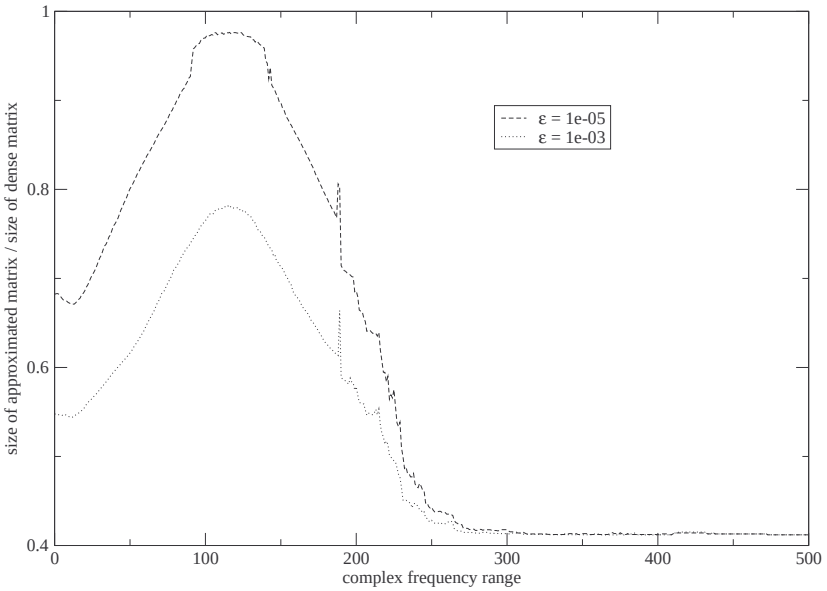


(b) Complex frequencies

Figure 4: Compression ratios and used complex frequencies s_ℓ for different β values

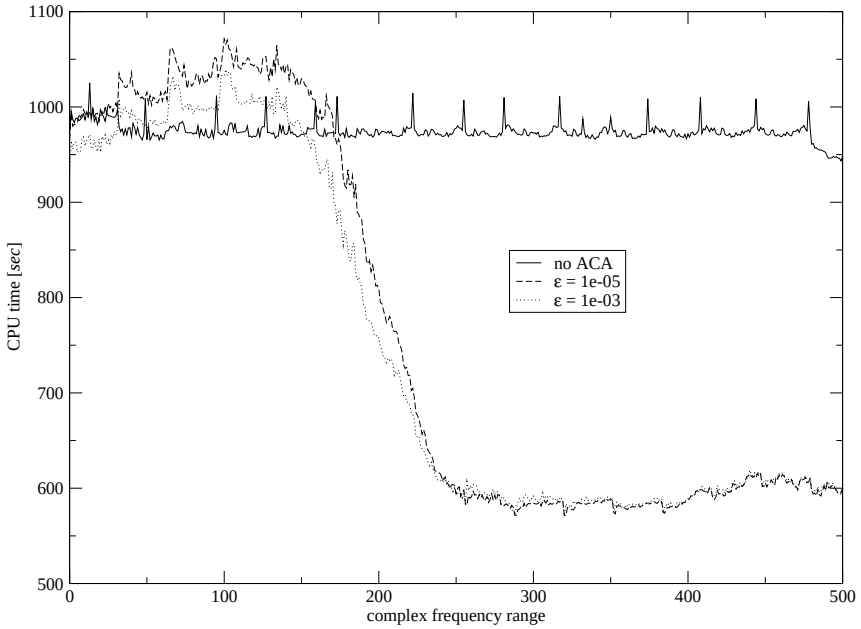


(a) Normal tractions at fixed end

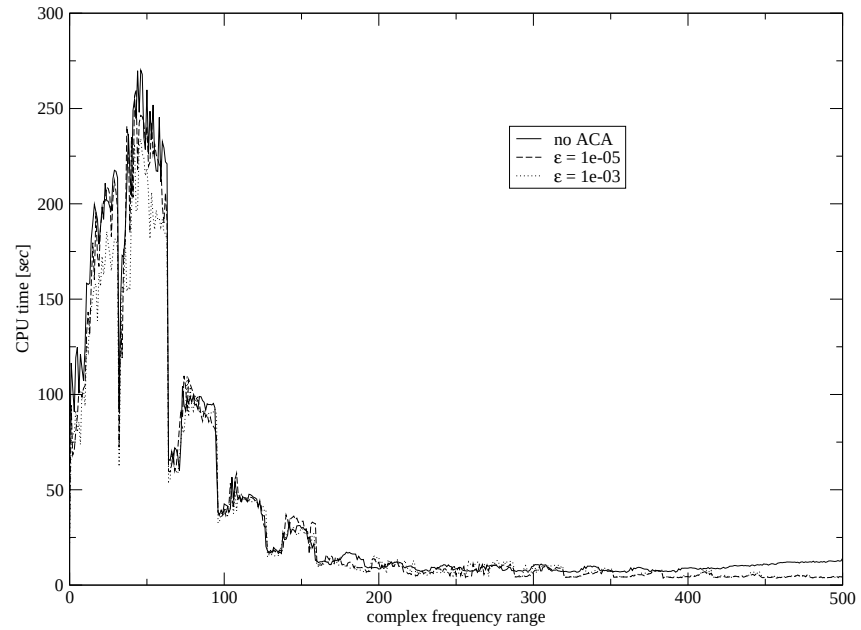


(b) Compression ratios of $\hat{D}_{11}$

Figure 5: Comparison of tractions and compression ratios for different ε values



(a) Matrix Assembly



(b) Iterative Solution

Figure 6: Comparison of CPU-time for different ε values

linear systems. Here, the correlation to the compression ratio is not such pronounced. However, it becomes apparent that condition numbers depend closely on the frequencies. The step-like structure in Figure 6b reflects the parallelized frequency intervals. In this case 32 CPU's were used.

5.2 Waves in the half-space

In this section, numerical results for half-space problems are compared with their analytical counterparts derived by Pekeris [40]. A plane half-space of the dimensions $80\,\mathrm{m} \times 80\,\mathrm{m}$ is excited

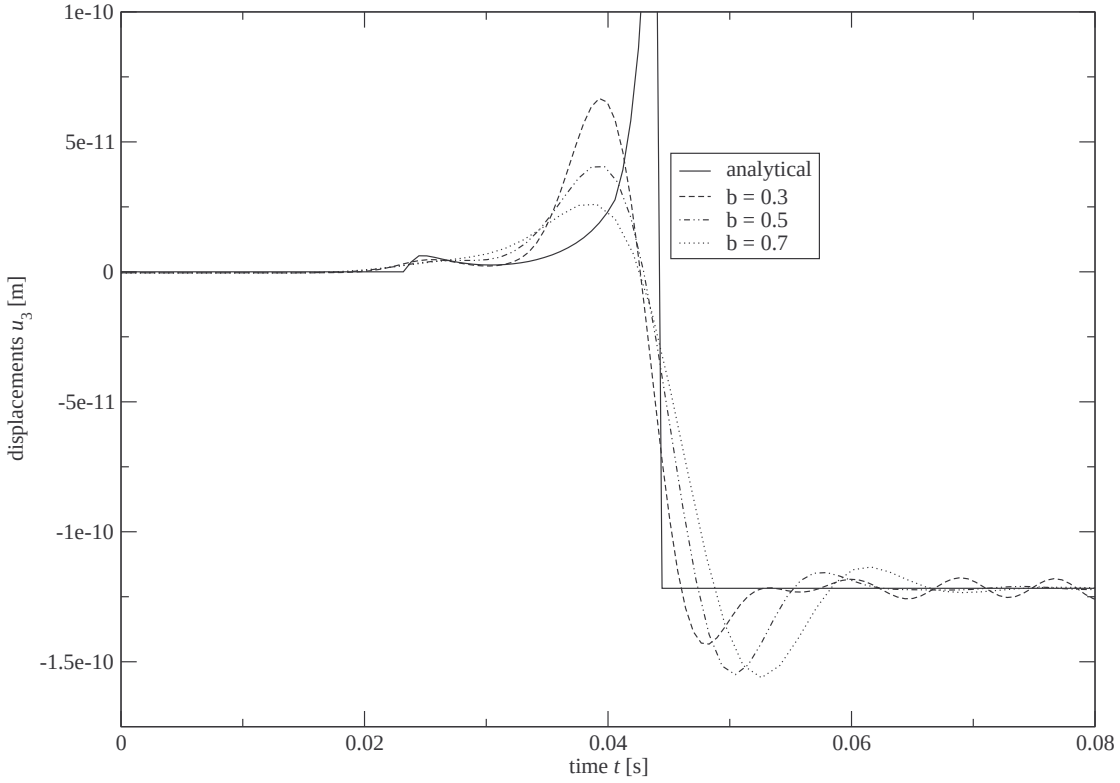


Figure 7: Vertical displacements u_3 at the observation point H

at the center by a vertical pressure jump $t_3 = -1.0H(t)\,\mathrm{N/m^2}$, i.e., the two elements at the center are excited. The material parameters of soil ($\rho = 1884.0\,\mathrm{kg/m^3}$, $\mu = \lambda = 1.3627 \times 10^8\,\mathrm{N/m^2}$) are taken. The present half-space is discretised into 14646 triangular boundary elements of uniform mesh size $h = 1.0\,\mathrm{m}$. Again, displacements and tractions are approximated by continuous linear and piecewise constant polynomials, respectively.

In Figure 7, numerically obtained results are compared to the analytical solution. The plot shows the vertical displacement at the observations point H, which is located at a distance of 10.97 m from the excitation. The results for three different time discretizations are presented. For the entire observation period of $T = 0.08\,\mathrm{s}$ and for $\beta = 0.3$ about 140, for $\beta = 0.5$ about 90 and for $\beta = 0.7$ about 60 time steps are necessary. All three β values lead to results which

reflect the Rayleigh wave quite well. However, only $\beta = 0.3$ allows to identify the arrival time ($t \approx 0.0235\,\mathrm{s}$) of the compressional wave.

Results in Figure 7 are computed with an approximation accuracy of $\varepsilon = 1.0 \times 10^{-05}$. The representative matrix $\hat{\mathsf{D}}_{11}$ for this example is of size 7484×7484 entries. Figure 8a shows the compression ratio for all three β values depending on the complex frequencies s_ℓ. It has the same behavior as in Figure 4a. Furthermore, the present half-space problem is solved also for $\varepsilon = 1.0 \times 10^{-03}$. The obtained displacements match perfectly with those presented in Figure 7. The respective compression ratios are compared in Figure 8b.

6 Conclusion

An accelerated time-domain boundary element formulation for elastodynamics is presented. The Galerkin discretization is adopted in space and a reformulated Convolution Quadrature Method in time. Due to this latter mentioned methodology it is possible to rewrite the hyperbolic problem into decoupled elliptic problems. This allows the usage of known fast techniques. By means of the $\mathcal{H}$-matrix format the far-field is separated from the near-field. Next, the Adaptive Cross Approximation is used to approximate far-field regions. The near-field is treated in the standard manner. Different to other applications of the ACA, here, vector-valued problems are dealt with. It is pointed out that a straight forward generalization of the algorithm to vector-valued problems is not possible. Hence, a repartitioning of the arising system matrices has been introduced. With this approach a kernel independent technique has been presented. E.g. the extension to visco- or poroelasticity is straight forward due to its black-box-like property.

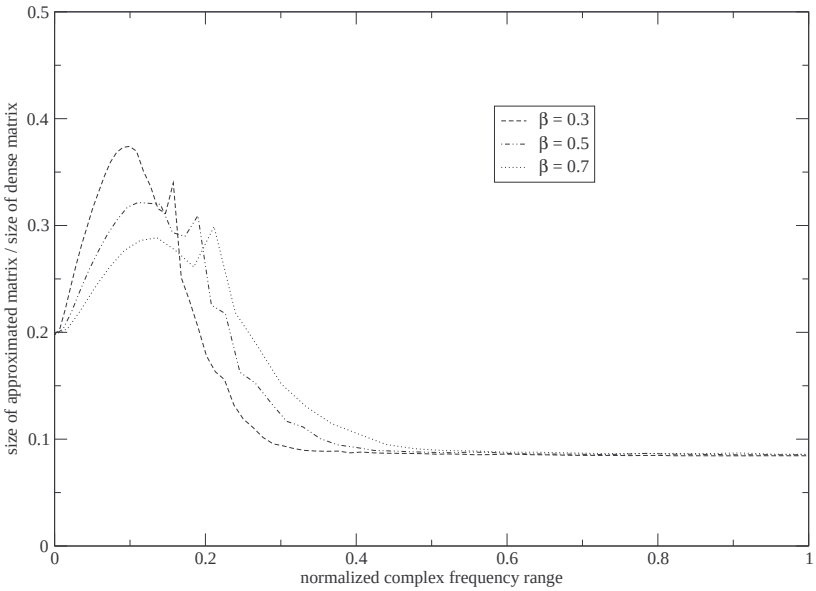
The presented numerical examples approve that this approach leads to results which match well with the analytical solutions. For both examples, the elastic rod and the half-space, $\beta = 0.3$ leads to the best results even tough the compression rates gets worse due to high frequencies. However, when talking about the quality of the results with respect to computational effort, $\beta = 0.5$ is recommended. Moreover, for both presented examples even the worse approximation accuracy $\varepsilon = 1.0 \times 10^{-3}$ leads to acceptable results.

An aspect which needs further investigations is the decreasing compression of matrices depending on frequencies having a large imaginary and a rather small real part. Investigations thereon are done in the work of Banjai and Hackbusch [3]. They improve the compression by splitting up the matrix into a sum of an $\mathcal{H}$ and $\mathcal{H}^2$-matrix. However, this fact unplugs from the kernel independent approach followed in the present work.

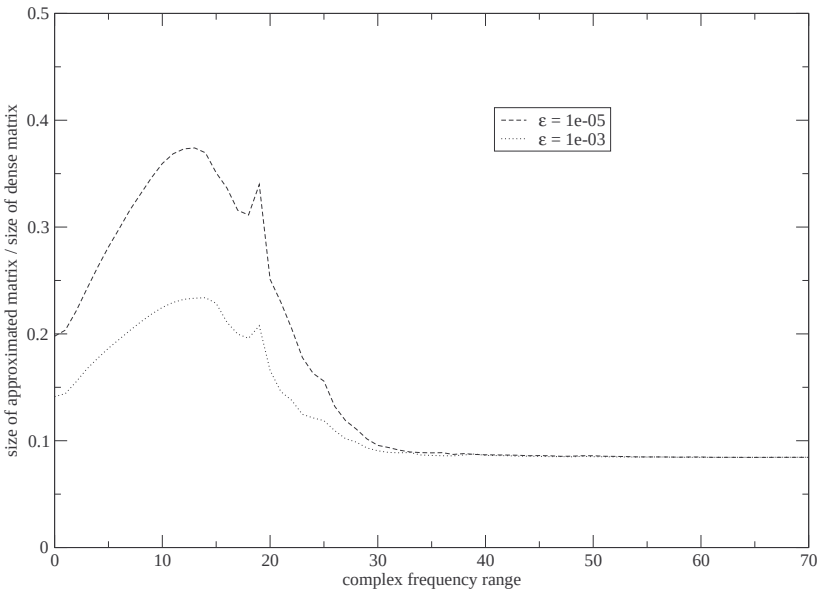
Acknowledgement The authors gratefully acknowledge the financial support by the Austrian Science Fund (FWF) under grant P-18481.

References

[1] B. Alpert, G. Beylkin, R. Coifman, and V. Rokhlin. Wavelet-like bases for the fast solutions of second-kind integral equations. *SIAM Journal for Scientific Computing*, 14(1):159–184, 1993.



(a) Compression ration for different β values



(b) Compression ratios for different ε values

Figure 8: Comparison of different compression rations for $\hat{D}_{11}$

[2] L. Banjai. Multistep and multistage convolution quadrature for the wave equation: Algorithms and experiments. *SIAM Journal for Scientific Computing [Submitted]*, 2009.

[3] L. Banjai and W. Hackbusch. Hierarchical matrix techniques for low- and high-frequency helmholtz problems. *IMA Journal of Numerical Analysis*, 28(1):46–79, 2008.

[4] L. Banjai and S. Sauter. Rapid solution of the wave equation in unbounded domains. *SIAM Journal on Numerical Analysis*, 47(1):227–249, 2008.

[5] M. Bebendorf. Approximation of boundary element matrices. *Numerische Mathematik*, 86(4):565–589, 2000.

[6] M. Bebendorf. Another software library on hierarchical matrices for elliptic differential equations (AHMED). `http://bebendorf.ins.uni-bonn.de/AHMED.html`, 2008. [Online; accessed 17-January-2010].

[7] M. Bebendorf. *Hierarchical Matrices: A Means to Efficiently Solve Elliptic Boundary Value Problems*, volume 63 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, 2008.

[8] M. Bebendorf and R. Grzibovski. Accelerating Galerkin BEM for linear elasticity using adaptive cross approximation. *Mathematical Methods in the Applied Sciences*, 29:1721–1747, 2006.

[9] M. Bebendorf and R. Kriemann. Fast parallel solution of boundary integral equations and related problem. *Computing and Visualization in Science*, 8(3):121–135, 2005.

[10] M. Bebendorf and S. Rjasanow. Adaptive low-rank approximation of collocation matrices. *Computing*, 70:1–24, 2003.

[11] I. Benedetti, M. H. Aliabadi, and G. Davi. A fast 3D dual boundary element method based on hierarchical matrices. *International Journal for Solids and Structures*, 45(7-8):2355 – 2376, 2008.

[12] D. E. Beskos. Boundary element methods in dynamic analysis. *Applied Mechanics Review*, 40(1):1–23, 1987.

[13] D. E. Beskos. Boundary element methods in dynamic analysis: Part II (1986-1996). *Applied Mechanics Review*, 50(3):149–197, 1997.

[14] A. Blázquez, R. Vodička, F. París, and V. Mantič. Comparing the conventional displacement BIE and the BIE formulations of the first and second kind in frictionless contact problems. *Engineering Analysis with Boundary Elements*, 26(10):815 – 826, 2002.

[15] S. Chaillat, M. Bonnet, and J.-F. Semblat. A multi-level fast multipole BEM for 3-D elastodynamics in the frequency domain. *Computer Methods in Applied Mechanics and Engineering*, 197(49–50):4233–4249, 2008.

[16] H. Cheng, L. Greengard, and V. Rokhlin. A fast adaptive multipole algorithm in three dimensions. *Journal of Computational Physics*, 155:468–498, 1999.

[17] I. Chudinovich. Boundary Equations in Dynamic Problems of the Theory of Elasticity. *Acta Applicandae Mathematicae*, 65:169–183, 2001.

[18] M. Costabel. Time-dependent problems with the boundary integral equation method. In E. Stein, R. de Borst, and T.J.R. Hughes, editors, *Encyclopedia of Computational Mechanics*, volume 1, chapter 25. John Wiley & Sons, New York, Chister, Weinheim, 2005.

[19] M. Costabel and E. P. Stephan. Integral equations for transmission problems in linear elasticity. *Journal of Integral Equations and Applications*, 2:211–223, 1990.

[20] T. A. Cruse and F. J. Rizzo. A Direct Formulation and Numerical Solution of the General Transient Elastodynamic Problem. I. *Journal of Mathematical Analysis and Applications*, 22:244–259, 1968.

[21] J. Domínguez. *Boundary Elements in Dynamics*. Computational Mechanics Publications, Southampton, Boston, 1993.

[22] W. Fong and E. Darve. The black-box fast multipole method. *Journal of Computational Physics*, 228(23):8712–8725, 2009.

[23] M. Frigo and S. G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. special issue on "Program Generation, Optimization, and Platform Adaptation".

[24] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin. A theory of pseudoskeleton approximations. *Linear Algebra and its Applications*, 261(1-3):1 – 21, 1997.

[25] K. F. Graff. *Wave Motions in Elastic Solids*. Dover: New York, 1991.

[26] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73:325–348, 1987.

[27] L. Greengard and V. Rokhlin. A new version of the fast multipole method for the Laplace equation in three dimensions. *Acta Numerica*, 6:229–269, 1997.

[28] T. Grytsenko and A. N. Galybin. Numerical analysis of multi-crack large-scale plane problems with adaptive cross approximation and hierarchical matrices. *Engineering Analysis with Boundary Elements*, 34(5):501 – 510, 2010.

[29] W. Hackbusch. A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: Introduction to $\mathcal{H}$-matrices. *Computing*, 62:89–108, 1999.

[30] W. Hackbusch. *Hierarchische Matrizen: Algorithmen und Analysis*. Springer-Verlag, 2009.

[31] W. Hackbusch and Z. P. Nowak. On the fast matrix multiplication in the boundary element method by panel clustering. *Numerische Mathematik*, 54(4):463–491, 1989.

[32] W. Hackbusch, W. Kress, and S. Sauter. Sparse Convolution Quadrature for Time Domain Boundary Integral Formulations of the Wave Equation by Cutoff and Panel-Clustering. In M. Schanz and O. Steinbach, editors, *Boundary Element Analysis*, volume 29 of *Lecture Notes in Applied and Computational Mechanics*, pages 113–134. Springer-Verlag Berlin Heidelberg, 2007.

[33] L. Kielhorn and M. Schanz. Convolution quadrature method based symmetric Galerkin boundary element method for 3-d elastodynamics. *International Journal for Numerical Methods in Engineering*, 76(11):1724–1746, 2008.

[34] C. Lubich. Convolution quadrature and discretized operational calculus I. *Numerische Mathematik*, 52:129–145, 1988.

[35] C. Lubich. Convolution quadrature and discretized operational calculus II. *Numerische Mathematik*, 52:413–425, 1988.

[36] F. Maerten. Adaptive cross-approximation applied to the solution of system of equations and post-processing for 3d elastostatic problems using the boundary element method. *Engineering Analysis with Boundary Elements*, 34(5):483 – 491, 2010.

[37] W. J. Mansur. *A Time-Stepping Technique to Solve Wave Propagation Problems Using the Boundary Element Method*. PhD thesis, University of Southampton, 1983.

[38] Ma. Messner, Mi. Messner, F. Rammerstorfer, and P. Urthaler. Hyperbolic and elliptic numerical analysis BEM library (HyENA). `http://www.mech.tugraz.at/HyENA`, 2010. [Online; accessed 22-January-2010].

[39] G. Of, O. Steinbach, and W. L. Wendland. A fast multipole boundary element method for the symmetric boundary integral formulation. In *IABEM 2002*, UT Austin, TX, USA, May 28-30 2002. International Association for Boundary Element Methods.

[40] C. L. Pekeris. The seismic surface pulse. In *Proceedings of the National Academy of Siences of the United States of America*, volume 41, pages 469–480, 1955.

[41] S. Rjasanow and O. Steinbach. *The Fast Solution of Boundary Integral Equations*. Springer-Verlag, 2007.

[42] V. Rokhlin. Rapid solution of integral equations of classical potential theory. *Journal of Computational Physics*, 60(2):187 – 207, 1985. ISSN 0021-9991.

[43] S. Sauter and C. Schwab. *Randelementmethoden*. B.G. Teubner Verlag, 2004.

[44] M. Schanz. *Wave Propagation in Viscoelastic and Poroelastic Continua: A Boundary Element Approach*, volume 2 of *Lecture Notes in Applied Mechanics*. Springer-Verlag, Berlin, Heidelberg, New York, 2001.

[45] M. Schanz. On a reformulated convolution quadrature based boundary element method. *Computer Modeling in Engineering and Sciences*, page (accepted), 2010.

[46] M. Schanz and H. Antes. A new visco- and elastodynamic time domain Boundary Element formulation. *Computational Mechanics*, 20:452–459, 1997.

[47] S. Sirtori, G. Maier, G. Novati, and S. Miccoli. A Galerkin symmetric boundary-element method in elasticity: Formulation and implementation. *International Journal for Numerical Methods in Engineering*, 35:255–282, 1992.

[48] O. Steinbach. *Numerical Approximation Methods for Elliptic Boundary Value Problems*. Springer, 2008.

[49] T. Takahashi, N. Nishimura, and S. Kobayashi. A fast BIEM for three-dimensional elastodynamics in time domain. *Engineering Analysis with Boundary Elements*, 28:165–180, 2002.