Numerical Algorithms manuscript No.
(will be inserted by the editor)

Fast boundary-domain integral method for unsteady convection-diffusion equation with variable diffusivity using the modified Helmholtz fundamental solution

Jure Ravnik · Jan Tibaut

Received: date / Accepted: date

Abstract In this paper, we develop a boundary domain integral formulation of the unsteady convection diffusion equation with variable material properties. The derivation is based on the Green's second theorem using the fundamental solution of the modified Helmholtz equation. Several discretization approaches are considered: the full matrix and domain decomposition approaches are compared with adaptive-cross-approximation and wavelet-based approximation techniques. With the use of modified Helmholtz fundamental solution, whose shape is determined by the time step size and diffusivity, we are able to achieve an improvement in the final approximated matrix size. We present several numerical tests to verify the validity of the proposed integral formulation and assess the approximation properties for different diffusivity variations and different Péclet numbers. We develop guidelines for choosing the user prescribed parameters such as the hierarchical matrix admissibility parameter, the adaptive cross approximation rank determination parameter and the wavelet thresholding parameter.

1 Introduction

The Boundary Element Method (BEM) [12] is a powerful technique used to solve partial differential equations (PDE). It relays on the use of Green's second theorem and the knowledge of the fundamental solution of the underlying problem to establish an integral formulation, which depends solely on the boundary unknowns - the unknown function and its normal derivative - flux. Thus, in order to find a

J. Ravnik
Faculty of Mechanical Engineering, University of Maribor, Smetanova ulica 17, SI-2000 Maribor, Slovenia
Tel.: +386 2 220 7745
E-mail: jure.ravnik@um.si

J. Tibaut
Faculty of Mechanical Engineering, University of Maribor, Smetanova ulica 17, SI-2000 Maribor, Slovenia
E-mail: jan.tibaut@um.si

solution of a PDE, BEM requires integration only over the boundary of the problem domain. The main drawback of the method is the fact that matrices in the final system of linear equations are fully populated. The fundamental solution of the governing PDE is also used in other numerical algorithms, such as the method of fundamental solutions [23] or meshless methods [6].

When inhomogeneous problems are considered, such as the general unsteady convection-diffusion-reaction equation, the analytical fundamental solution does not exist and the final integral equation includes domain integrals as well. When integrated natively, its computer storage and CPU time requirements scale as $\mathcal{O}(n^2)$, due to the fact that full domain and boundary integral matrices must be stored. This approach is known as the Boundary-Domain Integral Method (BDIM) (Škerget [37]). The dual reciprocity boundary element method proposed by Partridge et al. [27] can be used to avoid domain integration by using expansions with radial basis functions.

More recently, several techniques have been proposed to accelerate the solution of boundary integral equations [35]. The panel clustering method [1,20], the fast multipole method [18], wavelets [11], adaptive cross approximation (ACA) [7], and hierarchical $\mathcal{H}$ matrices [8] are all aimed at reducing the storage and computational cost from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$ or $\mathcal{O}(n)$. Most work has been done on boundary integral matrices. In this work we will apply ACA and wavelet compression onto a $\mathcal{H}$ matrix representation of boundary-boundary, boundary-domain and domain-domain integral matrices.

Transport phenomena, which include transport of momentum, mass, and heat, are the essential part of many engineering devices. They are the basis of many physical, chemical and environmental models. During last decades many researchers have been working on finding new numerical solution methods. Most of the work was done with constant fluid properties (diffusivity, viscosity, thermal conductivity). Dehghan [17] proposed a finite difference based numerical method for the solution of the three-dimensional advection-diffusion equation. Pudykiewicz [28] derived a finite volume algorithm for the solution of the reaction-advection-diffusion equation on the sphere. Sakai and Kimura [36] used a spectral method to solve a nonlinear two-dimensional unsteady advection-diffusion equation, which they transformed into a linear equation. Remešikova [34] proposed an operator splitting scheme for the numerical solution of two-dimensional convection-diffusion-adsorption problems. Kumar et al. [24] derived analytical solutions for the one-dimensional advection-diffusion equation with variable coefficients in a longitudinal finite domain. Cunha et al. [15] performed a study concerning the solution of advection-diffusion problems by BEM in 2D. Clavero and Jorge [14] proposed fractional step method for 2D parabolic convection-diffusion singularly perturbed problems.

The problem of variable diffusivity and fluid velocity presents a challenge for the use of BEM since these parameters are found in the fundamental solution. Repeated calculation of integrals due to unsteady or non-linear nature of the problem is impractical, thus researchers have proposed alternatives. Several variations of modelling of transport phenomena with variable material properties and velocity have been considered by [25,26,5,3,2,4,39,29,30].

The present paper focuses on the following topics. Firstly, we derive the integral formulation of the unsteady convection-diffusion-reaction equation with variable diffusivity and velocity using the modified Helmholtz fundamental solution.

Next, we propose four discretization approaches and finally, criticality compare the approaches using test problems that highlight challenges in the simulation of transport phenomena. We analyse the impact of adaptive cross approximation and wavelet approximation techniques and provide guidelines for optimal usage of these approximative algorithms.

2 The governing equations

In this section we derive the integral representations of the modified Helmholtz equation and the transport equation using the fundamental solution of the modified Helmholtz problem.

2.1 The modified Helmholtz equation

The inhomogeneous modified Helmholtz equation reads

$$\nabla^2 u - \mu_0^2 u = F, \quad \mathbf{r} \in \mathbb{R}^3, \tag{1}$$

where $u(\mathbf{r})$ is the unknown function and $F(\mathbf{r}, u)$ is a known inhomogeneous forcing and μ_0 is a constant. Sometimes referred to as the Yukawa equation [13] this equation is found in the Debye-Huckel theory, and in the linearization of the Poisson-Boltzmann equation. It also appears in implicit time marching schemes for the unsteady convection-diffusion transport equation, which is what we want to explore in this article. The fundamental solution $u^\star$ of the modified Helmholtz equation [37]:

$$(\nabla^2 - \mu_0^2) u^\star(\mathbf{r}, \boldsymbol{\xi}) = -\delta(\mathbf{r} - \boldsymbol{\xi}), \tag{2}$$

and its normal derivative $q^\star$ can be written as:

$$u^\star = \frac{1}{4\pi r} e^{-\mu_0 r}, \qquad q^\star = \mathbf{n} \cdot \boldsymbol{\nabla} u^\star = \frac{\mathbf{n} \cdot \mathbf{r}}{4\pi r^3}(1 + \mu_0 r) e^{-\mu_0 r}, \tag{3}$$

where $\boldsymbol{\xi}$ is the source point, $\mathbf{r}$ is the field point and the distance between the two is $r = |\boldsymbol{\xi} - \mathbf{r}|$. When considering the implicit time marching scheme for a transport equation, $\mu_0 = \sqrt{\mathcal{P}/\Delta t}$, where $\mathcal{P}$ is the Péclet number and Δt the time step. When convective transport dominates over diffusive transport the Péclet number is large, $\mathcal{P} \gg 1$ and the characteristic time scale for changes in the solution field is short, $\Delta t \to 0$. In such cases: $\mu_0 \gg 1$, and the fundamental solution diminishes quickly with distance from the source point. We want to make use of this property to achieve an improvement in approximation of integral matrices. In the opposite case, when diffusion dominates and the problem is almost steady, $\mu_0 \to 0$, the modified Helmholtz equation reverts to the Poisson type equation, for which the Laplace fundamental solution can be used. When the parameter μ varies with position, we may choose $\mu^2 = \mu_0^2 + \mu_v^2$, where μ_0 is a constant value and μ_v is the remaining variable part and, thus

$$(\nabla^2 - \mu_0^2) u = F + (\mu^2 - \mu_0^2) u. \tag{4}$$

The integral representation of such an inhomogeneous modified Helmholtz equation is

$$c(\boldsymbol{\xi})u(\boldsymbol{\xi}) + \int_{\Gamma} u\boldsymbol{\nabla}u^{\star}\cdot\mathbf{n}d\Gamma = \int_{\Gamma} u^{\star}qd\Gamma - \int_{\Omega} u^{\star}(F+(\mu^2-\mu_0^2)u)d\Omega, \tag{5}$$

where $q = \mathbf{n}\cdot\boldsymbol{\nabla}u$ is the flux.

2.2 The transport equation

The transport equation is used to model heat, mass, and momentum transport. Let u be a field function (temperature, concentration, momentum), which is transported in a fluid in a 3D domain $\Omega\in\mathbb{R}^3$ with the boundary $\Gamma=\partial\Omega$. The domain is filled with an incompressible fluid flowing with velocity $\mathbf{v}$. Let $\mathbf{r}$ represent a position in the domain. Under these assumptions, the transport equation is:

$$\frac{\partial u}{\partial t} + \mathbf{v}(\mathbf{r},t)\cdot\boldsymbol{\nabla}u = \boldsymbol{\nabla}\cdot(\alpha(\mathbf{r},t)\boldsymbol{\nabla}u) + f(\mathbf{r},t), \qquad \mathbf{r}\in\Omega, \tag{6}$$

where $\alpha(\mathbf{r},t)$ is an isotropic diffusion coefficient, which is a function of the location $\mathbf{r}$ and time t, and $f(\mathbf{r},t)$ represents sources of u in the domain.

We assume that the initial distribution of the field function $u(\mathbf{r},0)$ is known and that either Dirichlet $u(\mathbf{r},t)$ or Neumann $q(\mathbf{r},t)=\mathbf{n}\cdot\boldsymbol{\nabla}u(\mathbf{r},t)$ boundary conditions are known at the boundary $\mathbf{r}\in\Gamma$.

The transport equation may be recast into the modified Helmholtz form by employing an implicit time marching scheme using the second order backward finite difference approximation

$$\frac{\partial u}{\partial t} = \beta u + \beta' u' + \beta'' u'' = \beta u - g, \tag{7}$$

where Δt is the time step, $\beta = 3/(2\Delta t)$, $\beta' = -2/\Delta t$, and $\beta'' = 1/(2\Delta t)$, [30]. u is the function in the next time step, u' is the function in the current time step, and u'' is the function in previous time step. In order to shorten notation, we use $g=-(\beta' u' + \beta'' u'')$. The variable diffusion coefficient can be decomposed into constant and variable parts as follows: $\alpha(\mathbf{r},t) = \alpha_0 + \alpha'(\mathbf{r},t)$ where α_0 is the constant part and α' is the variable part. Finally, we define $\mu_0 = \sqrt{\beta/\alpha_0}$. Using these relationships, we may recast eq. (6) into modified Helmholtz form:

$$(\nabla^2 - \mu_0^2)u = \frac{1}{\alpha_0}\mathbf{v}\cdot\boldsymbol{\nabla}u - \frac{1}{\alpha_0}\boldsymbol{\nabla}\cdot\left(\alpha'\boldsymbol{\nabla}u\right) - \frac{1}{\alpha_0}(f+g). \tag{8}$$

2.2.1 Integral representation

Treating the terms on the right hand side of equation (8) as source terms, and making use of the modified Helmholtz fundamental solution, we derive the following integral representation for a source point $\boldsymbol{\xi}$

$$c(\boldsymbol{\xi})u(\boldsymbol{\xi},t) + \int_{\Gamma} uq^{\star}\cdot d\Gamma = \int_{\Gamma} u^{\star}qd\Gamma$$
$$+\frac{1}{\alpha_0}\underbrace{\int_{\Omega} u^{\star}\boldsymbol{\nabla}\cdot(\alpha'\boldsymbol{\nabla}u)d\Omega}_{\text{due to variable coef.}} - \frac{1}{\alpha_0}\underbrace{\int_{\Omega} u^{\star}\mathbf{v}\cdot\boldsymbol{\nabla}ud\Omega}_{\text{due to var. velocity}} + \frac{1}{\alpha_0}\underbrace{\int_{\Omega} u^{\star}\left(f+g\right)d\Omega}_{\text{sources}}, \tag{9}$$

where $c(\boldsymbol{\xi})$ is the geometrical factor. When the source point is at the boundary ($\boldsymbol{\xi} \in \Gamma$), the geometrical factor $c(\boldsymbol{\xi})$ depends on the shape of the boundary. When the source point is in the domain ($\boldsymbol{\xi} \in \Omega$) then $c = 1$. The first two domain integrals on the right hand side of equation (9) include the gradient of the unknown function. Calculation of this gradient can be avoided by algebraic manipulation to move the derivative towards the fundamental solution.

The domain integral, which is due to the variable velocity field, can be rewritten using the definition of divergence and the fact, that the velocity field is solenoidal:

$$\begin{aligned}\int_\Omega u^\star \mathbf{v} \cdot \boldsymbol{\nabla} u d\Omega = \int_\Omega \boldsymbol{\nabla} \cdot (u^\star u \mathbf{v}) d\Omega - \int_\Omega u \mathbf{v} \cdot \boldsymbol{\nabla} u^\star d\Omega = \\ = \int_\Gamma u^\star u \mathbf{v} \cdot d\boldsymbol{\Gamma} - \int_\Omega u \mathbf{v} \cdot \boldsymbol{\nabla} u^\star d\Omega, \end{aligned} \tag{10}$$

where the divergence theorem has been used to transform the domain integral into a boundary integral.

Next, let us focus on the domain integral, which is due to the variable coefficient. Using rules for chain differentiation, we may write

$$\begin{aligned}\int_\Omega u^\star \boldsymbol{\nabla} \cdot (\alpha' \boldsymbol{\nabla} u) d\Omega = \int_\Omega \boldsymbol{\nabla} \cdot (u^\star \alpha' \boldsymbol{\nabla} u) d\Omega + \int_\Omega u \boldsymbol{\nabla} \alpha' \cdot \boldsymbol{\nabla} u^\star d\Omega \\ - \int_\Omega \boldsymbol{\nabla} \cdot (\alpha' u \boldsymbol{\nabla} u^\star) d\Omega + \int_\Omega \alpha' u \nabla^2 u^\star d\Omega. \end{aligned} \tag{11}$$

The two integrals that feature a divergence of the kernel can be written as boundary integrals using the divergence theorem, yielding

$$\begin{aligned}\int_\Omega u^\star \boldsymbol{\nabla} \cdot (\alpha' \boldsymbol{\nabla} u) d\Omega = \int_\Gamma u^\star \alpha' \boldsymbol{\nabla} u \cdot d\boldsymbol{\Gamma} + \int_\Omega u \boldsymbol{\nabla} \alpha' \cdot \boldsymbol{\nabla} u^\star d\Omega \\ - \int_\Gamma \alpha' u \boldsymbol{\nabla} u^\star \cdot d\boldsymbol{\Gamma} + \int_\Omega \alpha' u \nabla^2 u^\star d\Omega. \end{aligned} \tag{12}$$

The kernel of the last domain integral in equation (12) includes a Laplacian of the fundamental solution. This can be rewritten by using the definition in equation (2) as

$$\int_\Omega \alpha' u \nabla^2 u^\star d\Omega = -\int_\Omega \alpha' u \delta(\mathbf{r}, \boldsymbol{\xi}) d\Omega + \mu_0^2 \int_\Omega \alpha' u u^\star d\Omega. \tag{13}$$

At this point we choose the constant part of the coefficient to be $\alpha_0 = \alpha(\boldsymbol{\xi})$. Thus α' is equal to zero at the source point, and since the Kronecker delta is zero everywhere else, the first integral on the right hand side of equation (13) vanishes. Using this, equation (12) simplifies to

$$\begin{aligned}\int_\Omega u^\star \boldsymbol{\nabla} \cdot (\alpha' \boldsymbol{\nabla} u) d\Omega = \int_\Gamma u^\star \alpha' \boldsymbol{\nabla} u \cdot d\boldsymbol{\Gamma} + \int_\Omega u \boldsymbol{\nabla} \alpha' \cdot \boldsymbol{\nabla} u^\star d\Omega \\ - \int_\Gamma \alpha' u \boldsymbol{\nabla} u^\star \cdot d\boldsymbol{\Gamma} + \mu_0^2 \int_\Omega \alpha' u u^\star d\Omega. \end{aligned} \tag{14}$$

Introducing equations (10) and (14) into (9), taking into account that $\boldsymbol{\nabla}\alpha = \boldsymbol{\nabla}\alpha'$, $\alpha'/\alpha(\boldsymbol{\xi}) = \alpha/\alpha(\boldsymbol{\xi}) - 1$ and multiplying the equation with $\alpha(\boldsymbol{\xi})$, we obtain the

following expression:

$$\alpha(\boldsymbol{\xi})c(\boldsymbol{\xi})u(\boldsymbol{\xi},t)+\int_{\Gamma}\alpha u q^{\star}d\Gamma=\int_{\Gamma}\alpha u^{\star}qd\Gamma-\int_{\Gamma}u^{\star}u\mathbf{v}\cdot\mathbf{n}d\Gamma+\int_{\Omega}u(\boldsymbol{\nabla}\alpha+\mathbf{v})\cdot\boldsymbol{\nabla}u^{\star}d\Omega+\int_{\Omega}u^{\star}\left(f+g+\beta\left(\frac{\alpha}{\alpha(\boldsymbol{\xi})}-1\right)u\right)d\Omega. \tag{15}$$

In this equation α is the diffusion coefficient and $\mathbf{v}$ is the fluid velocity, which both vary in space and time. The constant part of the coefficient $\alpha(\boldsymbol{\xi})$ is taken to be the coefficient at the location of the source point.

3 Discretization

We consider several discretization techniques to set up a system of linear equations for the solution the governing equation in a given domain. All are based on the collocation approach and used in such a way that we seek the solution of the problem on the boundary and in the domain as well. A computational mesh is produced in the entire domain. The collocation points are placed into boundary functions nodes and boundary flux nodes as well as into domain nodes in order to find the solution in the domain as well. The function u is interpolated over a boundary elements as $u=\sum\varphi_i u_i$, inside each domain element as $u=\sum\Phi_i u_i$, and flux is interpolated over boundary elements as $q=\sum\phi_i q_i$. The same interpolation scheme is used for all other fields as well.

Considering that the boundary is divided into b boundary elements ($\sum_b\Gamma_b=\Gamma$) and the interior domain into c cells ($\sum_c\Omega_c=\Omega$), we can utilize the interpolation scheme to write a discrete version of equation (15) as

$$\alpha(\boldsymbol{\xi})c(\boldsymbol{\xi})u(\boldsymbol{\xi},t)+\sum_b\sum_i u_{b,i}\int_{\Gamma_b}\alpha\varphi_i q^{\star}d\Gamma=\sum_b\sum_i q_{b,i}\int_{\Gamma_b}\alpha\phi_i u^{\star}d\Gamma-\sum_b\sum_i(u\mathbf{v})_{b,i}\cdot\int_{\Gamma_b}\varphi_i u^{\star}\mathbf{n}d\Gamma+\sum_c\sum_i[u(\boldsymbol{\nabla}\alpha+\mathbf{v})]_{c,i}\cdot\int_{\Omega_c}\Phi_i\boldsymbol{\nabla}u^{\star}d\Omega+\sum_c\sum_i\left(f+g+\beta\left(\frac{\alpha}{\alpha(\boldsymbol{\xi})}-1\right)u\right)_{c,i}\int_{\Omega_c}\Phi_i u^{\star}d\Omega, \tag{16}$$

where i represents a nodal index within an element. In the discretization procedure, the following types of integrals must be calculated and stored into matrices

$$[H]=\int_{\Gamma}\varphi_i\boldsymbol{\nabla}u^{\star}\cdot\mathbf{n}d\Gamma,\quad[G]=\int_{\Gamma}\phi_i u^{\star}d\Gamma, \tag{17}$$

$$[\mathbf{A}]=\int_{\Gamma}\varphi_i\mathbf{n}u^{\star}d\Gamma,\qquad[\mathbf{D}]=\int_{\Omega}\Phi_i\boldsymbol{\nabla}u^{\star}d\Omega,\quad[B]=\int_{\Omega}\Phi_i u^{\star}d\Omega, \tag{18}$$

where the square brackets denote matrices of integrals. The matrix $[B]$ is based on the same domain shape functions as matrices $[\mathbf{D}]$. The reason four matrices are needed steams from the fact that $[B]$ includes integrals featuring the fundamental

solution and [**D**] feature the three components of the gradient of the fundamental solution.

Let the curly brackets denote vectors of nodal values of functions. Then, the discrete version of equation (15) may be written as

$$[H]\{\alpha u\} = [G]\{\alpha q\} - [\mathbf{A}] \cdot \{\mathbf{v}u\} + [\mathbf{D}] \cdot \{(\boldsymbol{\nabla}\alpha + \mathbf{v})\, u\} + [B] \left\{ f + g + \beta \left(\frac{\alpha}{\alpha(\boldsymbol{\xi})} - 1 \right) u \right\}. \tag{19}$$

The $c(\boldsymbol{\xi})$ term was added to the diagonal of the $[H]$ matrix in order to reproduce the $\alpha(\boldsymbol{\xi})c(\boldsymbol{\xi})u(\boldsymbol{\xi})$ of equation (15) term correctly. In Eq. (7) we introduced a finite difference approximation of the time derivative $\partial u/\partial t$ defining u as the unknown in the next time step and u' and u'' as known fields in the two previous time steps. Thus, in Eq. (19) all u are unknown and the fields u' and u'', which are denoted by g, are known. Starting from the known initial condition, we solve (19) repeatedly to advance the solution through time.

The problem is significantly simplified in cases, when the diffusion coefficient is a constant, i.e.

$$[H]\{\alpha(\boldsymbol{\xi})u\} = [G]\{\alpha(\boldsymbol{\xi})q\} - [\mathbf{A}] \cdot \{\mathbf{v}u\} + [\mathbf{D}] \cdot \{\mathbf{v}u\} + [B]\{f + g\}, \tag{20}$$

and even further, when pure diffusive problems are considered ($\mathbf{v} = 0$)

$$[H]\{\alpha(\boldsymbol{\xi})u\} = [G]\{\alpha(\boldsymbol{\xi})q\} + [B]\{f + g\}. \tag{21}$$

Considering Eq. (19) one can take several approaches to placement of collocation points and setting up the system of linear equations, which enables the solution of unknown function and flux values. The simplest approach is the Boundary-Domain Integral Method (BDIM). For each collocation point, we calculate the integrals in full and store all values in large full matrices. Next, we consider a domain-decomposition based Boundary-Domain Integral Method (SD-BDIM), where BDIM is applied to subdomains, which are then connected via compatibility conditions. When subdomains are mesh cells, such an approach yields a finite-element like matrix structure with sparse matrices. Finally, we consider ACA and wavelet compression techniques in combination with $\mathcal{H}$ matrix structure to give a sparse representation of BDIM integral matrices. The approaches are described in more detail in the following subsections.

3.1 Boundary-Domain Integral Method (BDIM)

Since the final integral form of the governing equation (15) features domain integrals and since we seek the solution in the domain and at the boundary, we mesh the whole domain. The computational mesh includes boundary elements at the boundary and domain elements in the domain. The boundary elements are quadrilaterals with quadratic interpolation of function (nine function nodes per element) and linear interpolation of flux (four discontinuously placed flux nodes per element). The domain cells are hexahedra with 27 nodes for quadratic interpolation of function.

The collocation points are placed at the boundary into function or flux nodes depending on the prescribed boundary conditions. Furthermore, they are placed

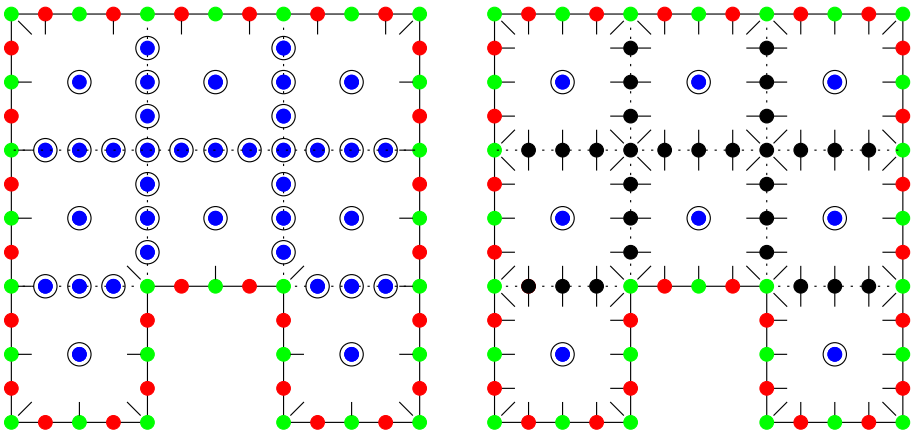

Fig. 1 A representation set up of the system of linear equations (2D cross-section). Green dots represent function nodes at the wall, red dots are flux nodes at the wall. Blue dots are nodes in the domain, where the function values is calculated explicitly. Short lines represent nodes, where a source point is placed and eq. (15) is solved. Circles represent nodes, where function values is obtained by explicitly solving eq. (15). In the black nodes, compatibility boundary conditions between subdomain are prescribed. DBIM is shown in the left panel, SD-BDIM in the left.

into domain nodes as well, since we would like to compute the solution in the domain as well. Due to collocation point placement the integral matrices (17) and (18) have the following structure: boundary $\times$ boundary (matrices $[\mathbf{A}]$ for flux collocation nodes), domain $\times$ domain (matrices $[\mathbf{D}]$, $[B]$), domain $\times$ boundary (matrices $[H]$, $[G]$, and matrices $[\mathbf{A}]$ for function collocation nodes).

The final system of linear equations is set up in such a way that its solution gives the unknown function and/or flux values at the boundary and the unknown function values in the domain. In the left panel of Fig. 1, we present an illustration of the set up of the system of linear equations. Due to clarity only a 2D cross-section through a domain is shown. Due to the relationship between the number of source points placed at the boundary (denoted by lines in Fig. 1) and in the domain (circles) we obtain a rectangular (domain $\times$ boundary) shape of matrices.

The system is solved by LU decomposition with diagonal preconditioning. A two-dimensional version of this approach has been proposed by Škerget et al. [37] and was extended to three dimensions in this paper.

3.2 Adaptive Cross Approximation ($\mathcal{H}$-ACA)

The adaptive cross approximation is an algebraic method, which provides a low rank approximation of matrices. It is usually applied on a hierarchical $\mathcal{H}$ matrix structure and accelerates matrix-vector multiplication from $O(N^2)$ to $O(N \log N)$. When used to accelerate an existing algorithm, such as BDIM in this paper, it is very easy to integrate into the procedure. Appropriate matrix storage data structure has to be set up at the start of the algorithm, to store low rank approximations. Then, only a routine, which performs ACA based matrix-vector multipli-

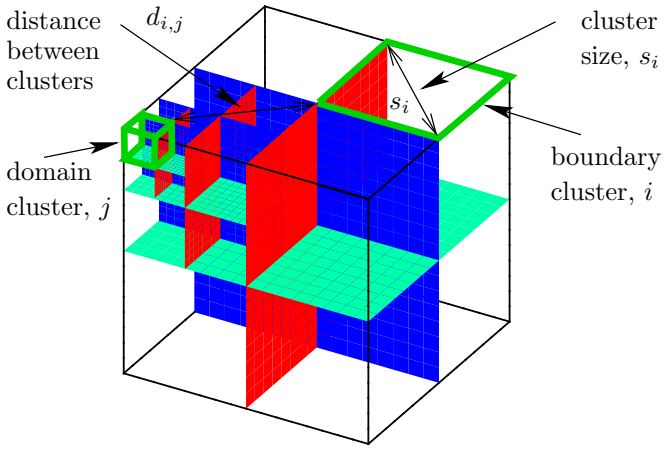

Fig. 2 A representation of the clustering algorithm. Hierarchical subdivision of the domain is shown together with an example of a boundary and a domain cluster. Cluster size and distance between clusters is also shown.

cation is needed to replace the standard multiplication routines in the main part of the algorithm.

The ACA can be use to approximate whole matrices, or it can be used only on parts of the matrix. The fundamental solution of the modified Helmholtz equation and its gradient form the integrands in the integrals, which make up fully populated integral matrices. Since the fundamental solution diminishes with distance (especially at large μ_0) it is beneficial to use the acceleration methods on smaller parts matrix parts, which correspond to a cluster of collocation points and a cluster of domain elements or boundary elements. When the clusters are separated by a large distance, we can expect to achieve high compression rates, since all integral values in a matrix part will take similar values and will be thus easier to approximate. We will take advantage of this fact by recursively hierarchically decomposing the domain into pieces, while at the same time examining the corresponding matrix parts for the expected quality of approximation. The idea of hierarchical decomposition was first proposed by Hackbusch [19].

Hierarchical decomposition of the domain can be achieved using a top-down approach (Ravnik et al. [33]) or bottom-up approach (Tibaut et al. [38]). In this work, we use the top-down approach. First, we encompass the entire computational domain in a parallelepiped. Then, using a recursive algorithm, we cut the domain through its centre of gravity using subsequent divisions by x, y and z planes. Since each cut produces two smaller clusters we form a binary cluster tree. Two such trees are produced: a boundary cluster tree and a domain cluster tree. Figure 2 shows an example of the clustering procedure and introduces the cluster size s_i and the distance between two clusters $d_{i,j}$.

Three types of matrices need to be approximated: boundary $\times$ boundary type, domain $\times$ domain type, and domain $\times$ boundary type. Thus, we make three combinations of cluster trees by pairing branches of the first tree with all branches in the second tree at the same level and at the next level. Thus, in case of domain $\times$ boundary type matrix, we end up with a boundary-domain cluster tree, whose

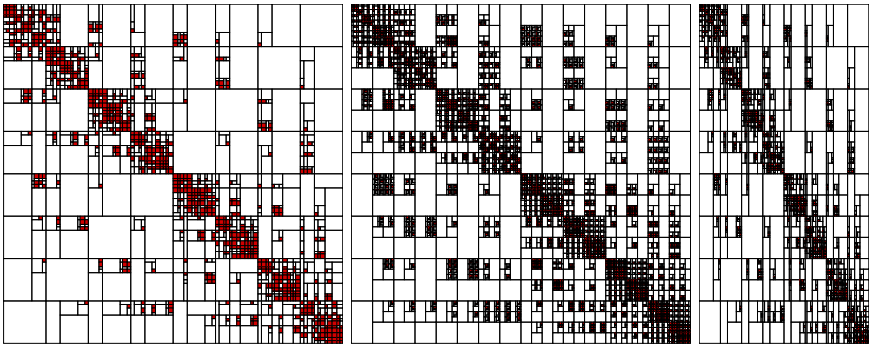

Fig. 3 Representation of admissible (white) and inadmissible (red) blocks in boundary × boundary matrix (left), domain × domain matrix (centre) and domain × boundary matrix (right). The computational domain was a cube meshed with 25^3 equidistant nodes, the admissibility parameter was $\eta = 3$.

branches include clusters of boundary nodes and domain nodes and correspond to a part of the hierarchical matrix structure. For tree branch (cluster pair), we decide weather it can be approximated (i.e. it is admissible) or it needs to be subdivided by progressing deeper into the tree. The admissibility criterion for clusters i and j is defined as

$$min(s_i, s_j) < \eta d_{i,j}, \tag{22}$$

where η is the admissibility parameter. In the case when the admissibility criterion is not satisfied even at the last level of the tree, such cluster pairs are inadmissible and will not be approximated. In Figure 3 an example of the hierarchically subdivided matrix structure is shown for all three cases: boundary × boundary, domain × domain, domain × boundary.

With the hierarchical matrix structure in place, we perform a low-rank approximation of each admissible matrix part using the cross approximation algorithm proposed by Bebendorf [7,8]. Inadmissible parts are not approximated and are stored in full. Let $[M]$ be a matrix part, which corresponds to an admissible cluster pair. The cross approximation algorithm sets up two low-rank matrices $[A]$ and $[B^T]$, which approximate a matrix part $[M]$ as

$$[M] \approx [A][B^T]. \tag{23}$$

If $[M]$ is of the size $(m \times n)$, then the size of $[A]$ and $[B^T]$ are $(m \times r)$ and $(r \times n)$, respectively. The rank of the approximation is r. Compression is achieved, when $r < mn/(m+n)$ [35]. In order to choose the rank r of the low-rank matrices, several approaches have been proposed [21,22]. Bebendorf [7] proposed an adaptive determination of the stopping criteria that is most widely used also by other authors. In this work, we propose to determine r by examining the accuracy of matrix times vector multiplication. For this, we introduce a new parameter, ϵ, which represents the target accuracy of matrix times vector multiplications. We propose the following implementation of the cross approximation algorithm:

1. $\{u\} = \mathrm{rnd}(0,1)$, $\{v\} = [M]\{u\}$
2. $[R]^0 = [M]$, $r = 0$

3. While $n > \epsilon$
   - $(i^*, j^*)^r = ArgMax(abs([R]^r))$
   - $\gamma = ([R]^r_{i^*,j^*})^{-1}$
   - $\{a\}^{r+1} = \gamma [R]^r_{i,j^*},\ \{b\}^{r+1} = ([R]^r_{i^*,j})^T$
   - $[R]^{r+1} = [R]^r - \{a\}^{r+1}\{b\}^{r+1}$
   - $n = ||[A]^{r+1}[B^T]^{r+1}\{u\} - \{v\}||$
   - $r = r + 1$

At the start of the algorithm, we define a random vector $\{u\}$ and multiply it with the matrix part and store the result in $\{v\}$. In the second step, we set up the residual matrix $[R]$, which is at the beginning equal to the matrix part $[M]$. We iterate while the random vector multiplication norm n is larger than the user-prescribed accuracy ϵ. Within each iteration, we choose the largest entry in the residual matrix $[R]^r_{i^*,j^*}$, extract the row $\{a\}$ and column $\{b\}$ around it and add them to $[A]$ and $[B]$. At the end of the loop, we update the random vector multiplication norm n by comapring the current approximation of matrix-vector mulitplication and the stored values in $\{v\}$. The relative RMS norm (25) is used to estimate the norm n. Finally, we update the rank r.

The final matrix is stored within a recursive hierarchical matrix structure, where each admissible branch stores the low rank matrices $[A]$ and $[B]$ and each inadmissible branch stores the matrix part $[M]$ in full. The matrix-vector multiplication algorithm recursively traverses the hierarchical structure and for each branch computes either $[M]\{u\}$ or $[A][B^T]\{u\}$ and sums up the final result vector.

3.3 Wavelet compression ($\mathcal{H}$-WT) and (FM-WT)

The fast wavelet transform (FWT) algorithm of Beylkin et al. [9] uses a pyramidal scheme to transform a vector into a wavelet basis. It employs compactly supported wavelets with m vanishing moments (Daubechies [16]). Ravnik at al. [31] proposed an algorithm to apply the FWT to vectors of arbitrary length. This method is used here to perform wavelet transform of matrix parts, whose sizes vary and are not a power of 2. In this work, the Haar wavelets ($m = 1$), which have a constant scaling function and non-overlapping support, are used.

For each admissible matrix block $[M]$ we should be able to calculate a matrix vector product, i.e. $[M]\{x\}$. Here $\{x\}$ represents appropriate nodal values corresponding to the matrix block. The FWT for vector of arbitrary length works by first expanding the vector to 2^n components, performing the Haar wavelet transform and contracting the vector:

1. $\{\tilde{x}\} = E\{x\}$
2. $\{x'\} = H\{\tilde{x}\}$
3. $\{\hat{x}\} = C\{x'\}$

The expansion, the Haar wavelet transform and the contraction are linear processes and can be represented by a matrices C, H and E. The expansion is done by duplicating the last values in $\{x\}$ vector, so that the length of the expanded vector $\{\tilde{x}\}$ is 2^n. When the Haar transform is applied to such a vector, one finds in $\{x'\}$ exactly the same number of non-zero components as are components in $\{x\}$. Finally, the zero values are omitted by the contraction operation C and the final result $\{\hat{x}\}$.

Let $W = CHE$ denote the complete wavelet matrix and W^{-1} its inverse. We may write

$$[M]\{x\} = \{f\} \quad \rightarrow \quad \underbrace{W[M]W^{-1}}_{[\hat{M}]}\underbrace{W\{x\}}_{\{\hat{x}\}} = \underbrace{W\{f\}}_{\{\hat{f}\}}, \tag{24}$$

where $[\hat{M}]$ is the matrix of wavelet coefficients and $\{\hat{x}\}$ and $\{\hat{f}\}$ are the wavelet transformed vectors. In absolute sense small values in the matrix $[\hat{M}]$ may be zeroed out (Bucher et al. [10]) without diminishing the accuracy of the final matrix vector product. The matrix times vector multiplication is performed in three steps

1. $\{\hat{x}\} = W\{x\}$,
2. $\{\hat{f}\} = [\hat{M}]\{\hat{x}\}$,
3. $\{f\} = W^{-1}\{\hat{f}\}$.

In the first step a wavelet transform of the vector of nodal values $\{x\}$ is performed. Secondly, the wavelet transformed vector $\{\hat{x}\}$ is multiplied by the matrix of wavelet coefficients. The final result is obtained in step three, where the inverse wavelet transform is performed.

The thresholding of the matrix $[\hat{M}]$ is done by comparing the absolute value of matrix elements to the average matrix element multiplied by a factor κ. Changing κ enables achievement of different compression rates and different final accuracy of the matrix times vector product. The thresholded matrix block $[\hat{M}]$ is stored in compressed row storage format as to free computer memory and so that the zero valued elements are not stored. The matrix-vector multiplication algorithm thus recursively traverses the hierarchical tree and performs compressed row storage times vector multiplications in each admissible branch and normal matrix-vector multiplications in inadmissible branches.

The same algorithm can be used on the whole matrices instead on the matrix parts in the hierarchical matrix structure. We denote this approach as FM-WT.

3.4 Subdomain Boundary-Domain Integral Method (SD-BDIM)

The SD-BDIM decomposes the entire domain into subdomains. In this work, subdomains are defined as the domain mesh elements. To set up the system of linear equations, the BDIM algorithm is used on each subdomain. Subdomains that share nodes are subject to compatibility boundary conditions. Source points are placed in each node of each subdomain. In the right panel of Fig. 1 we present equations in the system with short lines originating from nodes. Since neighbouring elements share nodes, we have several equations for each node. Thus, the final system is over-determined. It is solved in a least squares manner. The final system is also sparse, since BDIM is applied on subdomains and not on the whole domain. This approach has been proposed by Ravnik et al. [32], more details can be found in the reference. We included this approach for comparison purposes only.

4 Numerical examples

In order to verify the integral formulation and the proposed acceleration algorithms and ascertain their weaknesses and strengths, we have performed several numerical

experiments. Test cases with known analytical solutions are chosen, so we can report the difference between numerical and analytical solution using the following relative error norm

$$||u_n - u_a|| = \left(\frac{\sum_i (u_{n,i} - u_{a,i})^2}{\sum_i u_{a,i}^2} \right)^{\frac{1}{2}}, \tag{25}$$

where u_n is the numerical solution, u_a the analytical solution and the index i represents the nodal index.

The choices of the admissibility parameter η, the rank determination parameter ϵ, and the wavelet thresholding parameter κ lead to different matrix structures and ultimately to a different number of non-zero elements in final representations of matrices. In order to have an unified way of comparing results, let us define the *compression ratio* φ as the ratio between the number of elements in approximated matrices and the number of elements in the full matrices.

4.1 Test $\mathcal{A}$

We solve the transport equation (6) in a cubic domain $\Omega = (0,1)^3$. The flow velocity of the incompressible fluid is changing throughout the domain and is represented by the following function: $\mathbf{v} = (x, y, -2z)$. The fluid diffusion coefficient is also variable and takes the following values $\alpha = 1 + xyz$. There are sources in the domain $f = -5 + 2x^2 + 2y^2 - 12xyz - 4z^2$. The initial condition is $u|_{t=0} = x^2 + y^2 + z^2$. We choose a combination of Dirichlet and Neumann boundary conditions: at two opposite walls of the domain we prescribe a known function ($u = u_a$) and on the other four walls known flux ($q = \mathbf{n} \cdot \boldsymbol{\nabla} u_a$). Under these conditions, the analytical solution is quadratic in space and linear in time and reads as $u_a = x^2 + y^2 + z^2 + t$.

The problem was solved using three time step values, $\Delta t = 10^{-1}$, $\Delta t = 10^{-2}$, and $\Delta t = 10^{-3}$. Using these time steps the maximal value of the modified Helmholtz fundamental solution parameter in the computational domain is $\mu_0 = 3.9$, $\mu_0 = 12.2$, and $\mu_0 = 38.7$.

Three computational meshes were used, which had 9^3, 17^3, and 25^3 nodes. The node distribution in all meshes was uniform. Hierarchical matrix structure was obtained using five admissibility parameters, $\eta = 1, \ldots, 5$. Several choices of the rank determination parameter ϵ and wavelet thresholding parameter κ were made to yield different compression ratios.

In Figure 4, we present error norms versus compression ratio for all discretization approaches for different mesh densities, time steps and admissibility parameters. We observe that in all cases at high compression ratio the error norm of all fast techniques is equal to the BDIM error norm, which uses the original full matrices. Let us define the *optimal compression ratio* φ_o as the lowest compression ratio when the error norm of a fast technique is still at the same order of magnitude as the error obtained by BDIM. Up until the optimal compression ratio, the error introduced by matrix approximation is smaller than the errors introduced by the integration scheme, by the linear equations system solver, by the discretization, and by the interpolation. Below optimal compression ratio the error norm increases dramatically due to the poor accuracy of matrix time vector product made by approximated matrices. The optimal compression ratio is expected to

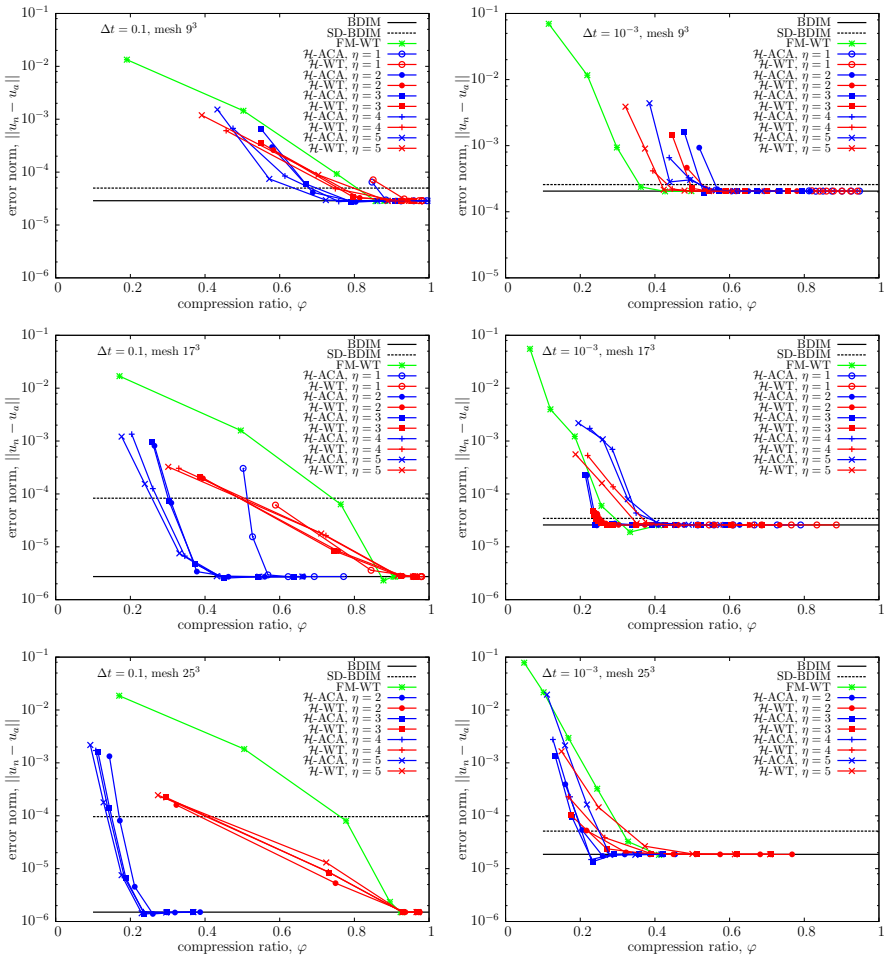

Fig. 4 Test case $\mathcal{A}$ error norms obtained using time step sizes $\Delta t = 10^{-1}$, $\Delta t = 10^{-3}$ and all three mesh densities: 9^3, 17^3, and 25^3. The results were evaluated at $t = 2$.

depend on the ability of the computational mesh to capture the solution field (the local cell Péclet number). Furthermore, since the time step chosen determines the shape of the fundamental solution, short time steps can lead to inaccuracies when computing integrals. On the other hand, short time steps (and large μ value) lead to a better approximation due to a more local character of the fundamental solution.

The results show the following properties of the optimal compression ratio. The best results are achieved by using $\eta = 2$ and $\eta = 3$, while at $\eta = 1$ the results are poor. At small admissibility parameter the distance between admissible clusters is small and thus the non-local nature of the fundamental solution prevents accurate results at low compression ratios. At large value of the admissibility parameter, the number of admissible blocks is too small to enable low compression ratios.

The wavelet approximation algorithm was used on the whole matrix (FM-WT) as well as combined with the $\mathcal{H}$ matrix structure ($\mathcal{H}$-WT). Usage of wavelet compression on the whole matrix (FM-WT) works similarly as the $\mathcal{H}$ matrix version for coarse meshes only. For dense meshes, usage of hierarchical matrix structure gives significantly better results.

The mesh density affects both the optimal admissibility ratio and the accuracy of the simulation in a positive way. The denser the computational mesh, the lower the optimal compression ratio and the more sparse the approximated matrices. Figure 5 explores the mesh dependence in more detail.

We observe better optimal compression ratio when a smaller time step is used (Figure 6). Smaller time step leads to larger μ value and the fundamental solution becomes steeper and in effect more local. This enables better approximation at a given accuracy. It can also be seen that the $\mathcal{H}$-ACA approach is less affected by the time step parameter and yields similar accuracy for all time steps, while $\mathcal{H}$-WT shows poor performance with long time steps.

Figure 7 shows the memory requirements of the $\mathcal{H}$-ACA and $\mathcal{H}$-WT at optimal compression ratio compared to BDIM, which uses full matrices. The poor performance of $\mathcal{H}$-WT at long time steps is observed, while $\mathcal{H}$-ACA and $\mathcal{H}$-WT as short time step perform similarly. Nearly linear dependence of the number of nodes is also confirmed.

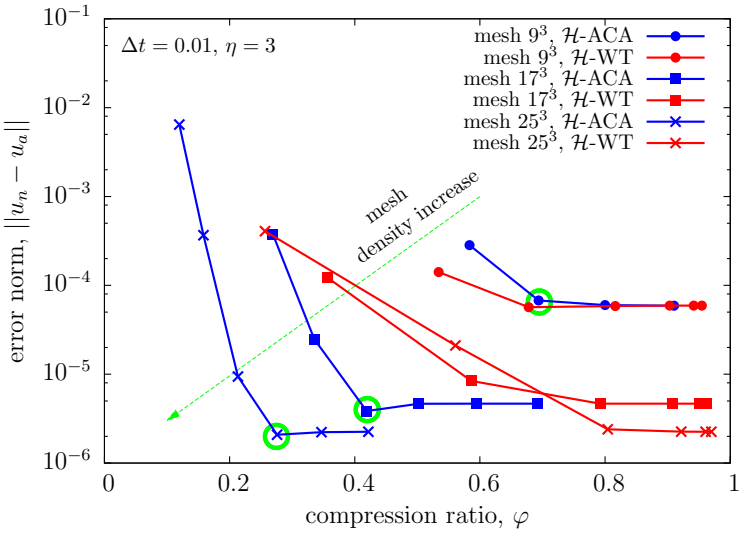

Fig. 5 Influence of the computational mesh density for test $\mathcal{A}$. Error norms for $\Delta t = 0.01$ and $\eta = 3$ are shown for ACA and wavelet compression techniques. An increase in mesh density clearly indicates that the optimal compression ratio, at which results accuracy is not diminished (denoted by green circles), decreases. The results were evaluated at $t = 2$.

In Figure 8 we present the time evolution of the error norms. We observe a high value of the error norm at the beginning of the simulation. This is due to the fact, that the initial approximation of the time derivative of the unknown function

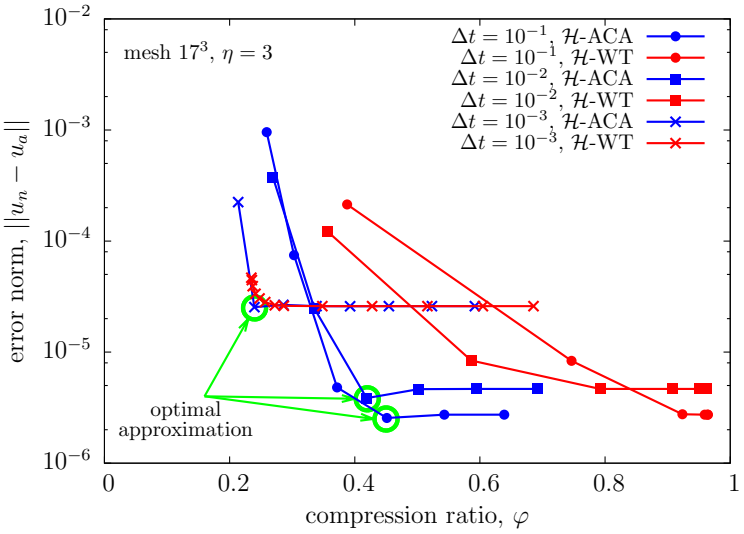

Fig. 6 Influence of the time step for test $\mathcal{A}$. Error norms for 17^3 mesh and $\eta = 3$ are compared for ACA and wavelet approximation techniques. A decrease in time step results in a more local character of the fundamental solution and thus, is easier to approximate. The results were evaluated at $t = 2$.

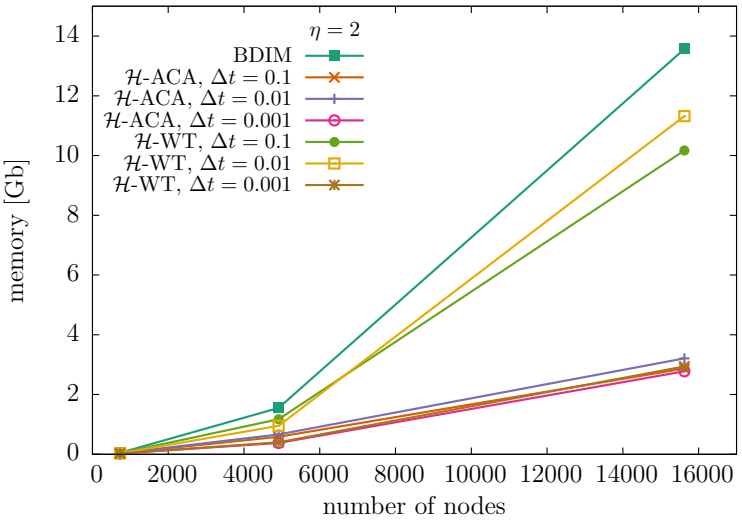

Fig. 7 Memory usage for approximated matrices shown versus the number of nodes in the computational mesh for the simulation at the optimal compression ratio, i.e. at the smallest memory usage, which does not affect the accuracy of simulation results.

$\partial u/\partial t|_{t=0}$ is unknown and an incorrect value of $\partial u/\partial t|_{t=0} = 0$ is used by the finite difference time stepping procedure. As time progresses is error norm decreased and reaches a stable value at about $t = 1$. When sub-optimal compression ratios are used, the algorithm converges to a higher stable error norm. When optimal compression ratio is used (or lower) the error norm time traces collapse on top of the time trace of the original BDIM algorithm, which does not use approximation techniques. This clearly demonstrates, that when optimal compression is used, the approximation part of the algorithm has a negligible influence on result accuracy. Comparing the results of the ACA and wavelet approximation techniques, we can observe similar behaviour - both techniques at optimal compression ratio do not affect the accuracy of simulation results. As already demonstrated, for this test case, the ACA technique exhibits a lower optimal compression ratio. Furthermore, we confirm, that lower time step size leads to better optimal compression ratio.

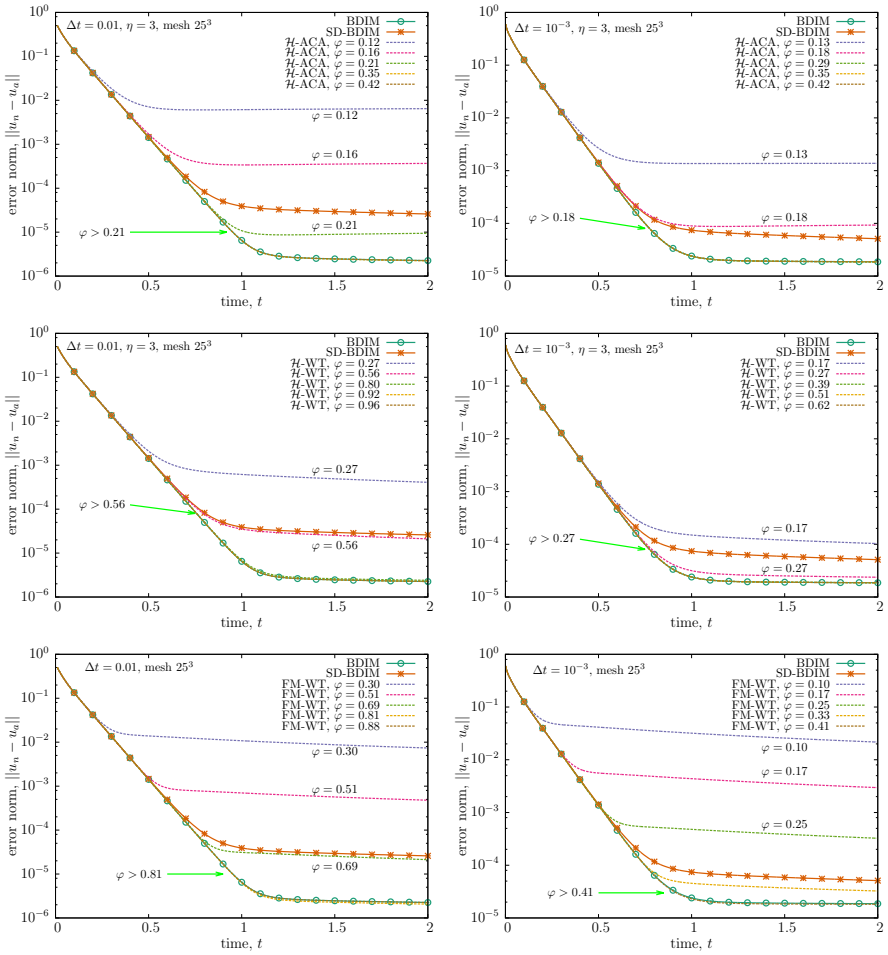

Fig. 8 Time evolution of the test case $\mathcal{A}$ error norms using 25^3 mesh and $\eta = 3$. Time step sizes were $\Delta t = 10^{-2}$ (left column) and $\Delta t = 10^{-3}$ (right column). Results of $\mathcal{H}$-ACA (top row) and wavelet transform techniques $\mathcal{H}$-WT (middle row) and FM-WT (bottom row) are presented.

4.2 Test $\mathcal{B}$

With the following series of tests, we highlight the importance of the variable diffusivity. By modelling diffusivity with a oscillatory trigonometric function as $\alpha = 1 + \sin(\nu xyz)$, where ν is the frequency of oscillation, we are able to asses the challenging cases when the mesh is barely adequate to capture the oscillatory nature of diffusivity. We consider a cubic domain $\Omega = (0,1)^3$ and solve the transport equation (6) with a prescribed velocity field $\mathbf{v} = (x, y, -2z)$. The domain sources are $-11 + 2x^2 + 2y^2 - 4z^2 - 6xyz\nu\cos(\nu xyz) - 6\sin(\nu xyz)$. Dirichlet boundary conditions are used at two opposite walls of the domain and Neumann boundary conditions are prescribed at all other walls. The initial condition is $u|_{t=0} = x^2 + y^2 + z^2$. The problem has an analytical solution $u_a = x^2 + y^2 + z^2 + t$ and was solved for $\nu = \pi/2, 3\pi/2$, and $5\pi/2$.

Since the spatial rate of change of diffusivity is governed by the frequency of oscillation ν, we present in Figure 9 the error norms versus the compression ratio obtained on a single mesh and using the same time step for different values of the frequency of oscillation. Since the computational mesh is kept the same for all cases, we observe a decrease of solution accuracy with increasing frequency of oscillations. This is expected and is a direct consequence of the ability of the mesh to capture the variations in diffusivity accurately. However, when we take a look at the accuracy of the ACA and wavelet approaches, we observe that the optimal compression ratio decreases with increasing frequency of oscillations. In order words, the inability of the computational mesh to capture variations in the diffusivity yields low accuracy of the final solution and thus enables lower rank in the ACA approach and more thresholded elements in the wavelet approach and thus the memory required for the storage of approximated matrices is lower. The conclusion of this analysis is that the accuracy of sparse approximation of integral matrices should be in sync with the overall accuracy of the numerical method.

We have shown in the analysis of results of the test case $\mathcal{A}$ that the temporal evolution of the error norm is unaffected by the introduction of approximation techniques at optimal compression level and below. In Figure 10 we present time traces of the error norm at approximately optimal compression ratio for different values of the ν parameter. We can confirm, that the solution accuracy remains unaffected by the approximation techniques. However, when sub-optimal compression ratio is used (see case $\mathcal{H}$-ACA, $\nu = \pi/2$ in the left panel of Fig. 10), a deterioration of the solution is observed.

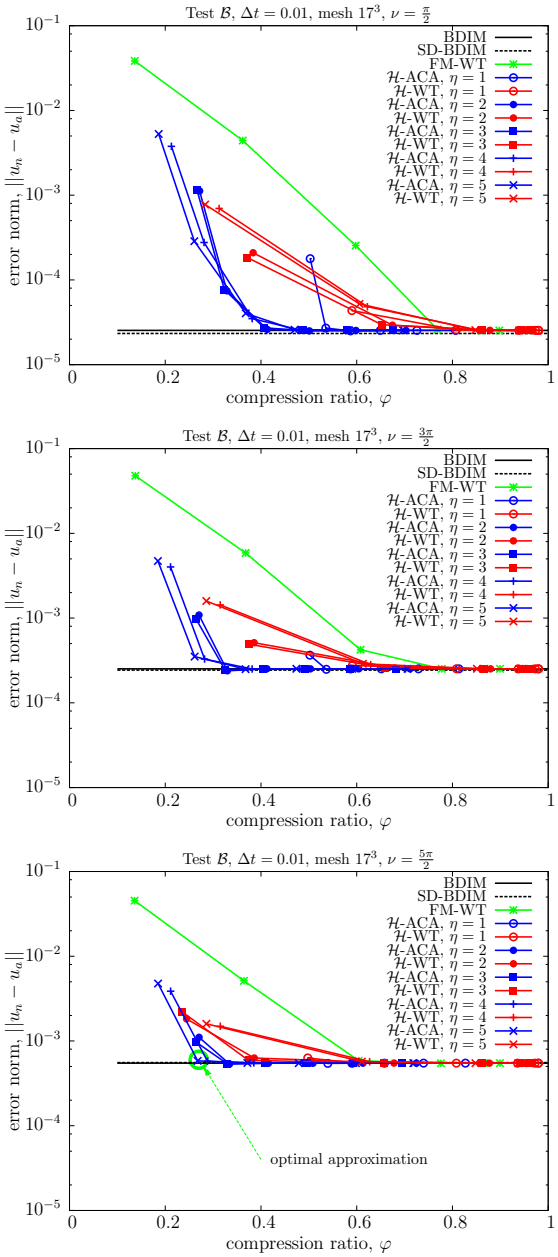

Fig. 9 Error norms versus compression ratio for test $\mathcal{B}$. The 17^3 mesh and $\Delta t = 0.01$ were used to perform simulations. Three models for diffusivity are considered, $\nu = \pi/2$ (top panel), $\nu = 3\pi/2$ (middle panel), and $\nu = 5\pi/2$ (bottom panel). The results were evaluated at $t = 2$.

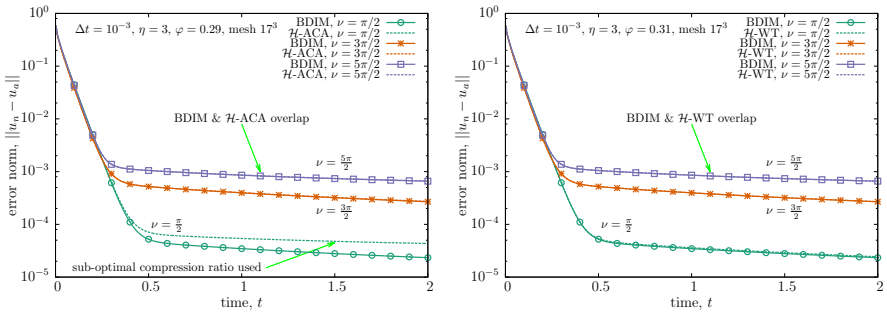

Fig. 10 Time evolution of the test case $\mathcal{B}$ error norms obtained using 17^3 mesh, $\eta = 3$ at approximately optimal compression ratio. Results of $\mathcal{H}$-ACA (left panel) and wavelet transform techniques $\mathcal{H}$-WT (right panel) are presented for different values of the ν parameter.

4.3 Test $\mathcal{C}$

The convection-diffusion type problems are specially challenging in the cases when convection dominates over diffusion. Let us denote the ratio between the characteristic time scale for diffusion and the characteristic time scale for convection as the Péclet number, $\mathcal{P}$. At high Péclet numbers, convection dominates and solutions are characterised by high gradient regions which can cause numerical difficulties.

In order to test the approximation methods developed in this paper, we propose the following test case. Let us consider a domain $\Omega = (0,1)^3$ where the fluid velocity in x direction is proportional to the Péclet number value, $\mathbf{v} = (\mathcal{P}, 1, 1)$. The diffusivity varies in the domain as $\alpha = 1 + xyz$ and is independent of the Péclet number. When the following type sources are present in the domain: $f = 1 + \mathcal{P}(-1 + \mathcal{P} - x\mathcal{P} + xyz\mathcal{P})x^{\mathcal{P}-2}$. At $x = 0$ and $x = 1$ Dirichlet type boundary conditions are used, all other walls have Neumann zero flux boundary conditions prescribed. Initially, the function is $u_|t = 0 = -x^{\mathcal{P}}$. Under these conditions, the solution of the transport equation (6) is $u_a = t - x^{\mathcal{P}}$. At high Péclet number, the x component of the gradient of the solution becomes large, i.e. $\partial u_a/\partial x|_{x=1} = -\mathcal{P}$, and thus presents a challenge for the numerical method.

We solved this problem by considering $\mathcal{P} = 2$, 5, and 10 using the 17^3 mesh. Since elements in the mesh had equals sizes, this means that the distance of the first node from the wall, where the large gradient must be captured, was 1/16.

In Figure 11 flux at the wall $\partial u_a/\partial x|_{x=1}$ is shown for all different methods. We observe poor performance of the subdomain based method SD-BDIM at high Péclet values. This is due to the finite element like setup of the system of linear equations, which is limited to individual subdomains and thus the size of the mesh elements at the wall plays a vital role. The fast approaches all produce very good result up to optimal compression ratio. Approximating more results in fast deterioration of the flux value. The flux at the wall is found to be independent of the admissibility parameter, since all results for $\eta = 1, \ldots 5$ are the same up to the optimal compression ratio.

In Figure 12 we explore the relationship between the optimal approximation and the Péclet number. Since the accuracy of simulation result is lower at a high Péclet number, the optimal compression ratio is better for high Péclet number cases, since a higher error of the approximated matrix-vector product is allowed. This relationship is indeed revealed for both $\mathcal{H}$-ACA and $\mathcal{H}$-WT as shown in Figure 12.

We introduced the parameter ϵ to determine the rank of the low-rank ACA matrices based on the error norm of approximated matrix times random vector multiplication. Figure 12 proves, that optimal approximation is achieved when the parameter ϵ is chosen such, that it is of the same order of magnitude than the accuracy of the original numerical scheme, which uses non-approximated full matrices.

This finding also explains the fact that using a denser mesh enables better compression ratio for the same accuracy. If the desired accuracy is not changed, using a denser mesh gives high accuracy results, which can be exploited by using a sub-optimal approximation to yield a better compression ratio and still satisfy the final accuracy requirement.

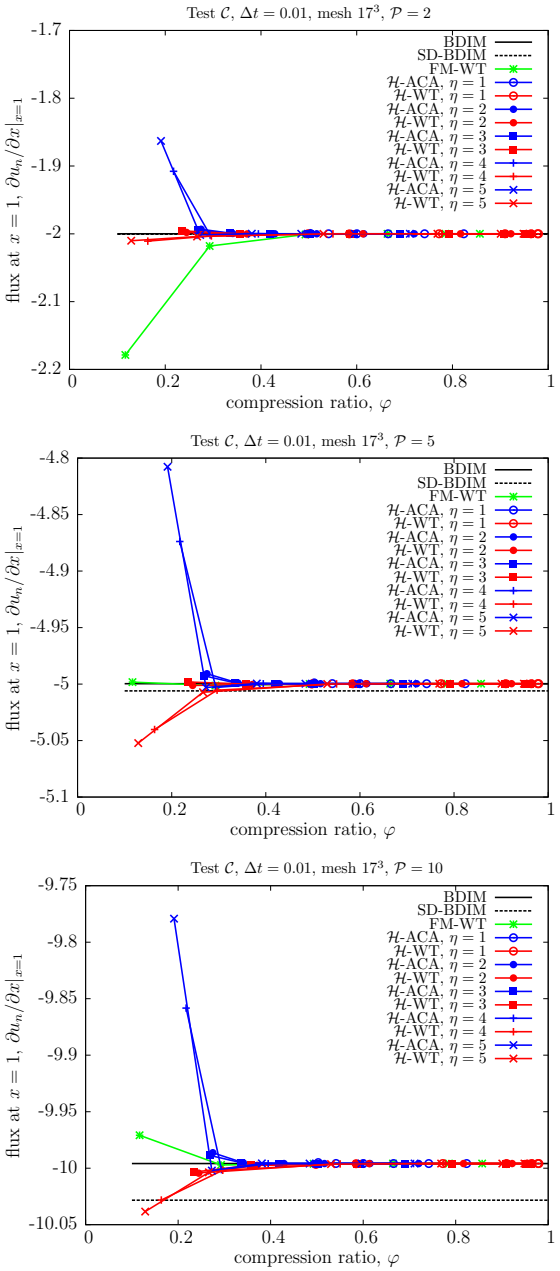

Fig. 11 Simulation results for normal gradient at $x = 1$ for test $\mathcal{C}$. Analytical value of the gradient is $-\mathcal{P}$. Top panel: $\mathcal{P} = 2$, middle panel: $\mathcal{P} = 5$, and bottom panel $\mathcal{P} = 10$. The results were evaluated at $t = 2$.

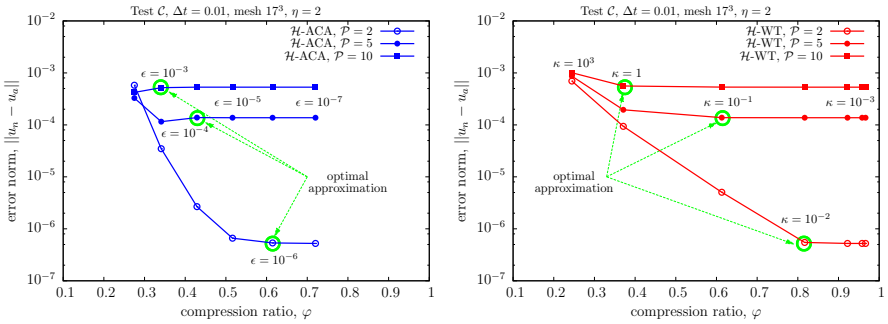

Fig. 12 Relationship between optimal approximation and the Péclet number. Results of test $\mathcal{C}$ at $\Delta t = 0.01$, $\eta = 2$ using the 17^3 mesh are shown. Left panel: $\mathcal{H}$-ACA, right panel: $\mathcal{H}$-WT. User prescribed values of the ACA rank parameter ϵ and wavelet thresholding parameter κ are also shown. The results were evaluated at $t = 2$.

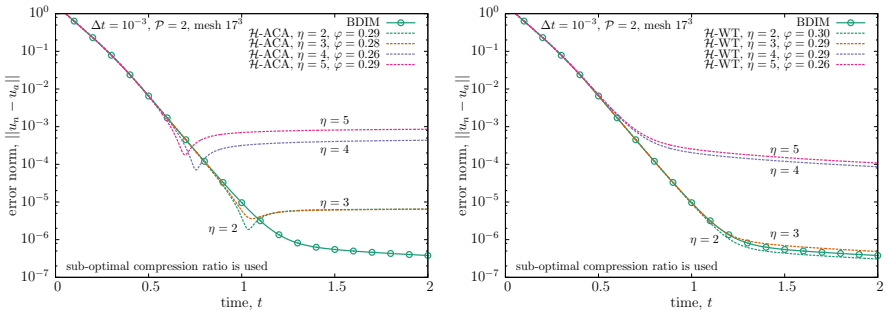

Fig. 13 Dependence of the time evolution of the error norm on the admissibility parameter η. Results of test $\mathcal{C}$ at $\Delta t = 10^{-3}$, $\mathcal{P} = 2$ using the 17^3 mesh are shown. Left panel: $\mathcal{H}$-ACA, right panel: $\mathcal{H}$-WT.

In the right panel of Figure 12 we observe that the wavelet thresholding parameter κ needed to achieve optimal approximation increases with the Péclet number. The reason is the same as with the ACA method, the accuracy of the solution deteriorates with the Péclet number, thus we are able to use lower compression ratios to match the error introduced by wavelet approximation of matrix-vector products to the error of the method itself.

In Fig. 13 we explore how the admissibility parameter affects the time evolution of the error norms. Choosing a sub-optimal compression ratio of approximately $\varphi \approx 0.3$, we ran simulations at different admissibility parameter values. In all cases, we observe a decrease of the error norm up to a certain level. It is clear, that the choice of η determines the evolution of the errors norm. The best results are obtained with $\eta = 2$ and $\eta = 3$ for both approximation techniques: ACA and wavelet transform.

5 Conclusions

The focus of this study was twofold: to derive an integral formulation of the unsteady convection-diffusion equation based on the modified Helmholtz fundamental solution and to investigate the applicability of ACA and wavelet matrix approximation techniques. The modified Helmholtz fundamental solution is characterised by an additional parameter, which depends on the time step and diffusivity. We established that by manipulating this parameter one can manipulate the optimal compression ratio and thus choose the most efficient approximation method to perform simulations. This is the main advantage of the proposed integral formulation in comparison, for example, to integral formulations derived using the Laplace fundamental solution.

Numerical tests have shown how the optimal compression ratio depends on the time step size, the diffusivity, the mesh density and the admissibility parameter. We provided guidelines on how to choose the user-prescribed parameters (admissibility parameter, ACA rank parameter and wavelet thresholding parameter) to reach the optimal compression ratio. We showed that in order to reach the optimal compression ratio is user-prescribed parameters must be in sync with the expected discretization error and with the linear system solver accuracy.

Acknowledgements The authors acknowledge the financial support from the Slovenian Research Agency (research core funding No. P2-0196) and the Deutsche Forschungsgemeinschaft (project STE 544/58).

References

1. Aimi, A., Diligenti, M., Lunardini, F.: Panel clustering method and restriction matrices for symmetric Galerkin BEM. Numerical Algorithms **40**(4), 355–382 (2005). DOI 10.1007/s11075-005-8136-x
2. AL-Jawary, M., Ravnik, J., Wrobel, L., Škerget, L.: Boundary element formulations for numerical solution of two-dimensional diffusion problems with variable coefficients. Computers and Mathematics with Applications **64**(8), 2695–2711 (2012). DOI 10.1016/j.camwa.2012.08.002. URL http://linkinghub.elsevier.com/retrieve/pii/S0898122112005135
3. AL-Jawary, M., Wrobel, L.C.: Numerical solution of two-dimensional mixed problems with variable coefficients by the boundary-domain integral and integro-differential equation methods. Engineering Analysis with Boundary Elements **35**, 1279–1287 (2011)
4. AL-Jawary, M., Wrobel, L.C.: Numerical solution of the twodimensional Helmholtz equation with variable coefficients by the radial integration boundary integral and integro-differential equation methods. International Journal of Computer Mathematics **89**(11), 1463–1487 (2012). DOI 10.1080/00207160.2012.667087
5. AL-Jawary, M., Wrobel, L.C.: Radial integration boundary integral and integro-differential equation methods for two-dimensional heat conduction problems with variable coefficients. Engineering Analysis with Boundary Elements **36**(5), 685–695 (2012). DOI 10.1016/j.enganabound.2011.11.019. URL http://dx.doi.org/10.1016/j.enganabound.2011.11.019
6. Antunes, P.R.: A numerical algorithm to reduce ill-conditioning in meshless methods for the Helmholtz equation. Numerical Algorithms pp. 1–19 (2018). DOI 10.1007/s11075-017-0465-z
7. Bebendorf, M.: Approximation of boundary element matrices. Numerische Mathematik **86**(4), 565–589 (2000). DOI 10.1007/PL00005410
8. Bebendorf, M., Rjasanow, S.: Adaptive low-rank approximation of collocation matrices. Computing **70**(1), 1–24 (2003). DOI 10.1007/s00607-002-1469-6

9. Beylkin, G., Coifman, R., Rokhlin, V.: Fast Wavelet Transforms and Numerical Algorithms 1. Wavelets and Applications **20**(2), 368–393 (1991). DOI 10.1002/cpa.3160440202
10. Bucher, H.F., Wrobel, L.C., Mansur, W.J., Magluta, C.: A Novel Approach to Applying Fast Wavelet Transforms in Boundary Element Method. El. J. of Bound. Elements **BETEQ 2001**, 187–195 (2002)
11. Bucher, H.F., Wrobel, L.C., Mansur, W.J., Magluta, C.: On the block wavelet transform applied to the boundary element method. Engineering Analysis with Boundary Elements **28**(6), 571–581 (2004). DOI 10.1016/j.enganabound.2003.10.002
12. Cheng, A.D.A.H.D., Cheng, D.T.: Heritage and early history of the boundary element method. Engineering Analysis with Boundary Elements **29**(3), 268–302 (2005). DOI 10.1016/j.enganabound.2004.12.001. URL http://linkinghub.elsevier.com/retrieve/pii/S0955799705000020
13. Cheng, H., Huang, J., Leiterman, T.J.: An adaptive fast solver for the modified Helmholtz equation in two dimensions. Journal of Computational Physics **211**(2), 616–637 (2006). DOI 10.1016/j.jcp.2005.06.006
14. Clavero, C., Jorge, J.C.: A fractional step method for 2D parabolic convection-diffusion singularly perturbed problems: uniform convergence and order reduction. Numerical Algorithms **75**(3), 809–826 (2017). DOI 10.1007/s11075-016-0221-9
15. Cunha, C., Carrer, J., Oliveira, M., Costa, V.: A study concerning the solution of advection - diffusion problems by the Boundary Element Method. Engineering Analysis with Boundary Elements **65**, 79–94 (2016). DOI 10.1016/j.enganabound.2016.01.002. URL http://www.sciencedirect.com/science/article/pii/S0955799716000047
16. Daubechies, I.: Orthonormal bases of compactly supported wavelets. Comm. Pure Appl. Math. **41**, 909–996 (1988)
17. Dehghan, M.: Numerical solution of the three-dimensional advection-diffusion equation. Applied Mathematics and Computation **150**(1), 5–19 (2004). DOI 10.1016/S0096-3003(03)00193-0. URL http://www.sciencedirect.com/science/article/pii/S0096300303001930
18. Greengard, L., Rokhlin, V.: A fast algorithm for particle simulations. J. Comput. Phys. **73**, 325–348 (1987)
19. Hackbusch, W.: A Sparse Matrix Arithmetic Based on H-Matrices . Part I : Introduction to H-Matrices. Computing **108**, 89–108 (1999). DOI 10.1007/s006070050015
20. Hackbusch, W., Nowak, Z.P.: On the fast multiplication in the boundary element method by panel clustering. Numerische Mathematik **54**, 463–491 (1989)
21. Heldring, A., Ubeda, E., Rius, J.M.: On the convergence of the ACA algorithm for radiation and scattering problems. IEEE Transactions on Antennas and Propagation **62**(7), 3806–3809 (2014). DOI 10.1109/TAP.2014.2316293
22. Heldring, A., Ubeda, E., Rius, J.M.: Stochastic estimation of the frobenius norm in the ACA convergence criterion. IEEE Transactions on Antennas and Propagation **63**(3), 1155–1158 (2015). DOI 10.1109/TAP.2014.2386306
23. Karageorghis, A.: The method of fundamental solutions for elliptic problems in circular domains with mixed boundary conditions. Numerical Algorithms **68**(1), 185–211 (2015). DOI 10.1007/s11075-014-9900-6
24. Kumar, A., Jaiswal, D.K., Kumar, N.: Analytical solutions of one-dimensional advection-diffusion equation with variable coefficients in a finite domain. J. Earth Syst. Sci. **118**(5), 539–549 (2009)
25. Mikhailov, S.E.: Localized boundary-domain integral formulations for problems with variable coefficients. Engineering Analysis with Boundary Elements **26**, 681–690 (2002)
26. Mikhailov, S.E., Nakhova, I.S.: Mesh-based numerical implementation of the localized boundary domain integral equation method to a variable-coefficient Neumann problem. Journal of Engineering Mathematics **51**, 251–259 (2005)
27. Partridge, P.W., Brebbia, C.A., Wrobel, L.C.: The dual reciprocity boundary element method. Computational Mechanics Publications Southampton, U.K. ; Boston : Computational Mechanics Publications ; London ; New York (1992)
28. Pudykiewicz, J.A.: Numerical solution of the reaction-advection-diffusion equation on the sphere. Journal of Computational Physics **213**(1), 358–390 (2006). DOI 10.1016/j.jcp.2005.08.021. URL http://www.sciencedirect.com/science/article/pii/S0021999105003840
29. Ravnik, J., Škerget, L.: A gradient free integral equation for diffusion - convection equation with variable coefficient and velocity. Eng. Anal. Bound. Elem. **37**, 683–690 (2013). DOI 10.1016/j.enganabound.2013.01.012

30. Ravnik, J., Škerget, L.: Integral equation formulation of an unsteady diffusion-convection equation with variable coefficient and velocity. Computers and Mathematics with Applications **66**(0), 2477–2488 (2014). DOI http://dx.doi.org/10.1016/j.camwa.2013.09.021
31. Ravnik, J., Škerget, L., Hriberšek, M.: The wavelet transform for BEM computational fluid dynamics. Eng. Anal. Bound. Elem. **28**, 1303–1314 (2004)
32. Ravnik, J., Škerget, L., Žunič, Z.: Velocity-vorticity formulation for 3D natural convection in an inclined enclosure by BEM. International Journal of Heat and Mass Transfer **51**(17-18), 4517–4527 (2008). DOI 10.1016/j.ijheatmasstransfer.2008.01.018
33. Ravnik, J., Škerget, L., Žunič, Z.: Fast single domain-subdomain BEM algorithm for 3D incompressible fluid flow and heat transfer. International Journal for Numerical Methods in Engineering **77**(12), 1627–1645 (2009). DOI 10.1002/nme.2467. URL http://doi.wiley.com/10.1002/nme.2467
34. Remešikova, M.: Numerical solution of two-dimensional convection-diffusion-adsorption problems using an operator splitting scheme. Applied Mathematics and Computation **184**(1), 116–130 (2007). DOI 10.1016/j.amc.2005.06.018. URL http://www.sciencedirect.com/science/article/pii/S0096300306004905
35. Rjasanow, S., Steinbach, O.: The Fast Solution of Boundary Integral Equations. Mathematical and Analytical Techniques with Applications to Engineering. Springer US, Boston, MA (2007). DOI 10.1007/0-387-34042-4. URL http://link.springer.com/10.1007/0-387-34042-4
36. Sakai, K., Kimura, I.: A numerical scheme based on a solution of nonlinear advection-diffusion equations. Journal of Computational and Applied Mathematics **173**(1), 39–55 (2005). DOI 10.1016/j.cam.2004.02.019. URL http://www.sciencedirect.com/science/article/pii/S0377042704001323
37. Škerget, L., Hriberšek, M., Kuhn, G.: Computational Fluid Dynamics By Boundary Domain Integral Method. International Journal for Numerical Methods in Engineering **46**(8), 1291–1311 (1999). DOI 10.1002/(SICI)1097-0207(19991120)46:8¡1291::AID-NME755¿3.0.CO;2-O
38. Tibaut, J., Škerget, L., Ravnik, J.: Acceleration of a BEM based solution of the velocity-vorticity formulation of the Navier-Stokes equations by the cross approximation method. Engineering Analysis with Boundary Elements **82**, 17–26 (2017). DOI 10.1016/j.enganabound.2017.05.013
39. Yang, K., Gao, X.W.X.W.: Radial integration BEM for transient heat conduction problems. Eng. Anal. Bound. Elem. **34**(6), 557–563 (2010). DOI 10.1016/j.enganabound.2010.01.008. URL http://dx.doi.org/10.1016/j.enganabound.2010.01.008