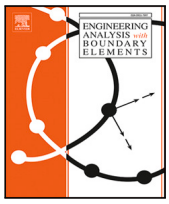
Engineering Analysis with Boundary Elements 169 (2024) 106015

Contents lists available at ScienceDirect

Engineering Analysis with Boundary Elements

journal homepage: www.elsevier.com/locate/enganabound



Fluid flow simulation with an $\mathcal{H}^2$-accelerated Boundary-Domain Integral Method

J. Tibaut [a,*], J. Ravnik [b], M. Schanz [a]

[a] Institute of Applied Mechanics, Graz University of Technology, Technikerstraße 4, A-8010 Graz, Austria
[b] Faculty of Mechanical Engineering, University of Maribor, Smetanova ulica 17, SI-2000 Maribor, Slovenia

ARTICLE INFO

Keywords:
Boundary-Domain Integral Method
Velocity–vorticity
Adaptive cross approximation
Modified Helmholtz equation
Yukawa potential
Fast multipole method
$\mathcal{H}$-structure

ABSTRACT

The development of new numerical methods for fluid flow simulations is challenging but such tools may help to understand flow problems better. Here, the Boundary-Domain Integral Method is applied to simulate laminar fluid flow governed by a dimensionless velocity–vorticity formulation of the Navier–Stokes equation. The Reynolds number is chosen in all examples small enough to ensure laminar flow conditions. The false transient approach is utilized to improve stability.

As all boundary element methods, the Boundary-Domain Integral Method has a quadratic complexity. Here, the $\mathcal{H}^2$-methodology is applied to obtain an almost linear complexity. This acceleration technique is not only applied to the boundary only part but more important to the domain related part of the formulation. The application of the $\mathcal{H}^2$-methodology does not allow to use the rigid body method to determine the singular integrals and the integral free term as done until now. It is shown how to apply the technique of Guigiani and Gigante to handle the strongly singular integrals in this application. Further, a parametric study shows the influence of the introduced approximation parameters. For this purpose the example of a lid driven cavity is utilized. The second example demonstrates the performance of the proposed method by simulating the Hagen–Poiseuille flow in a pipe. The third example considers the flow around a rigid cylinder to show the behavior of the method for an unstructured grid. All examples show that the proposed method results in an almost linear complexity as the mathematical analysis promisses.

1. Introduction

The Boundary-Domain Integral Method [1] is based on Green's second identity. When the fundamental solution of the partial differential equation is known, the volume integral in Green's second identity vanishes and a boundary only formulation is obtained. After discretization it is called Boundary Element Method. However, not for all operators the fundamental solution is known, e.g., for the Poisson equation. This equation can be handled with the fundamental solution of the Laplace operator resulting in an integral equation with a volume integral for the right hand side of Poisson's equation. The volume integral can be transformed into a boundary integral, e.g., with the dual reciprocity method. Al-Bayati and Wrobel employed this method to solve convection–diffusion problems with first-order chemical reactions [2]. Guo et al. [3] used the triple reciprocity method to solve the heat equation with heat sources. The transformation of the domain integral into a boundary integral does not improve the quadratic computational complexity but reduces the number of unknowns from $\mathcal{O}(m^2)$ to $\mathcal{O}(n^2)$, where n and m denote the number of the unknowns on the boundary and the domain, respectively.

The Boundary-Domain Integral Method is a numerical method that is used to solve boundary-volume problems. There are several authors who have employed the method for various differential equations, e.g., for inhomogeneous media governed by the diffusion equation [4,5] or Stokes flow with variable viscosity coefficients [6,7].

To accelerate the Boundary-Domain Integral Method the methods developed for the Boundary Element Method can be used, e.g., $\mathcal{H}$-matrices, Fast Multipole Method (FMM) or $\mathcal{H}^2$-matrices. An hierarchical partition of the initial matrix results in an $\mathcal{H}$-matrix where those matrix blocks fulfilling an admissibility condition are approximated [8]. Approximation techniques are, e.g., Adaptive cross approximation (ACA) [9] or Singular Value Decomposition (SVD) [10]. The most efficient is the Singular Value Decomposition. However, it is also the most computationally costly one. Another possibility is the Fast Multipole Method (FMM), where the integral kernel, i.e., the fundamental solution is approximated [11]. Darve [12]

* Corresponding author.
E-mail addresses: jan.tibaut@tugraz.at (J. Tibaut), jure.ravnik@um.si (J. Ravnik), m.schanz@tugraz.at (M. Schanz).

https://doi.org/10.1016/j.enganabound.2024.106015
Received 7 May 2024; Received in revised form 10 October 2024; Accepted 22 October 2024

used the FMM to solve the Maxwell equation or Messener and Schanz [13] applied it on the transient heat transfer within a time domain Boundary Element formulation. The Burton Miller formulation for acoustic analysis with FMM has been published by Jelich et al. [14] and Li et al. [15]. Further, fluid flow was solved with the velocity–vorticity form of the Navier–Stokes equations and an FMM accelerated Boundary-Domain Integral Method [16,17]. To accelerate the computation time within the FMM Wang et al. [18] used GPU's. More applications can be found in the review article of Nishimura [19].

The $\mathcal{H}^2$-method is as well based on the approximation of the integral kernel and $\mathcal{H}$-matrices. Börm and Hackbush presented this fast method in [20,21]. For this method the integral kernel is interpolated with the Lagrange polynomials and nested cluster bases are used. The $\mathcal{H}^2$-matrix has been proposed for different problems, e.g., Chai et al. [22] compared the $\mathcal{H}$-matrix and $\mathcal{H}^2$-approach for large scale electromagnetic analysis, Börm [23] solved high-frequency problems and Tibaut et al. [24] presented the fast Boundary-Domain Integral Method accelerated with the $\mathcal{H}^2$-method to solve the Poisson equation. It is important to note that the FMM and $\mathcal{H}^2$-matrices are similar. However, the $\mathcal{H}^2$-matrix is a more generic formulation.

In the paper at hand, a dimensionless velocity–vorticity formulation of the Navier–Stokes equations is solved with the Boundary-Domain Integral Method. To increase the stability of the formulation a false transient term is added. This time derivative is an under relaxation that increases the stability of the numerical simulation where the Boussinesq-Approximation for buoyancy is present in the partial differential equation. Further, this under-relaxation parameter change the characteristic form of the fundamental solution in the integral equation. In [24], it is shown how the accuracy of the integral kernel approximation is changed with the false transient approach. Other applications of the false transient approach can be found by Guj and Stella [25] or Behnia et al. [26]. After stating the set of governing equations, the Boundary-Domain Integral Method is formulated. To reduce the computational cost, essentially the storage, the $\mathcal{H}^2$-method is applied. The basics for the $\mathcal{H}^2$-method, i.e., the clustering and the interpolation procedure is recalled. The proposed formulation is then applied to the example of a lid-driven cavity, the Hagen–Poiseuille flow in a pipe and the channel flow with a cylindrical barrier.

2. Problem setting

To simulate stationary fluid flow the velocity–vorticity form of the Navier–Stokes equations is solved with the Boundary-Domain Integral Method. The mass and momentum conservation equations are considered.

2.1. Governing equations

Let $\Omega \subset \mathbb{R}^3$ be a bounded domain and $\Gamma := \partial\Omega$ its boundary with the outward normal $\vec{n}$. To solve the fluid flow, we consider the dimensionless velocity–vorticity formulation of the Navier–Stokes equations. The mass and momentum conservation equations are formulated with the dimensionless variables $\vec{v} \to \frac{\vec{v}^*}{v_0}, \vec{x} \to \frac{\vec{x}^*}{L}, \vec{\omega} \to \frac{\vec{\omega}^* L}{v_0}$ and $t \to \frac{v_0 t^*}{L}$, while the vorticity $\vec{\omega} = \vec{\nabla} \times \vec{v}$ is inserted into the equations. A full description of the formulation is given in [27]. The velocity–vorticity form of the equations results in the dimensionless form

$$\begin{aligned} \vec{\nabla}_x \times \vec{\omega} + \nabla_x^2 \vec{v} &= 0 \,, \\ (\vec{v} \cdot \vec{\nabla}_x)\vec{\omega} &= (\vec{\omega} \cdot \vec{\nabla}_x)\vec{v} + \frac{1}{Re} \nabla_x^2 \vec{\omega} \,, \end{aligned} \tag{1}$$

where $Re = \frac{v_0 L}{\vartheta_0}$ is the Reynolds number. Here, only smal Reynolds numbers are considered such that the fluid flow is laminar. The first equation in (1) is the kinematic equation obtained from the mass conservation equation and the second is the transport vorticity equation derived from the momentum conservation equation. To increase the stability of the fluid flow simulation, we add a so-called false transient term $\frac{\partial \vec{v}}{\partial t}$ to the kinematic equation [28]

$$\frac{\partial \vec{v}}{\partial t} = \nabla_x^2 \vec{v} + \vec{\nabla}_x \times \vec{\omega} \quad \forall\, (\vec{x}, t) \in \Omega \times (0, T). \tag{2}$$

Note, the false transient term $\frac{\partial \vec{v}}{\partial t}$ has no physical meaning nor any connection to a real time. The initial and boundary conditions are

$$\begin{aligned} \vec{v}(\vec{x}, t=0) &= 0, \quad \vec{x} \in \Omega \;\; t = 0, \\ \vec{\omega}(\vec{x}, t=0) &= \vec{\omega}, \quad \vec{x} \in \Omega \;\; t = 0, \\ \vec{v}(\vec{x}, t) &= \vec{v}, \quad \vec{x} \in \Gamma \times (0, T) \,, \end{aligned} \tag{3}$$

where T presents the final time and $\vec{v}$, $\vec{\omega}$ are the prescribed velocity and vorticity data. Note, the equation for $\vec{\omega}$ is not a boundary condition but a starting value for the solution of the problem. Let us discretize the time in steps $t_0 = 0,\ t_1,\ \ldots,\ t_N = T$ with constant time steps Δt. A time-discrete form of (2) is obtained by approximating the time derivative by a first-order finite difference scheme. This results for $n = 1,\ \ldots,\ N$ in

$$\frac{\vec{v}(\vec{x}, t_n) - \vec{v}(\vec{x}, t_{n-1})}{\Delta t} = \nabla_x^2 \vec{v}(\vec{x}, t_n) + \vec{\nabla}_x \times \vec{\omega}(\vec{x}, t_n) \,, \tag{4}$$

where $\vec{v}(\vec{x}, t_{n-1})$ denotes the value of the velocity field at the previous time step. Rearrangement and using the abbreviations $\mu^2 = \frac{1}{\Delta t}$, $b(\vec{x}, t_n) = \frac{1}{\Delta t}\vec{v}(\vec{x}, t_{n-1}) + \vec{\nabla}_x \times \vec{\omega}(\vec{x}, t_n)$ results in the Yukawa kinematic equation

$$\left(\nabla_x^2 - \mu^2\right) \vec{v}(\vec{x}, t_n) + b(\vec{x}, t_n) = 0. \tag{5}$$

This equation is also called modified Helmholtz kinematic equation. In case of large time steps $\Delta t \to \infty$, the parameter μ^2 tends to zero, i.e., $\mu^2 \to 0$. Thus in the limit, the kinematic equation in (1) is obtained.

2.2. The integral form of the governing equations

Green's second identity is applied to obtain the related integral equations to the transport vorticity equation in (1) and the modified Helmholtz kinematic equation (5). First, the integral equation to (5) is

$$c(\vec{y})\vec{v}(\vec{y}) + \int_\Gamma \vec{v}(\vec{x})q^*(\vec{y},\vec{x})d\Gamma = \int_\Gamma \vec{v}(\vec{x}) \times [\vec{n} \times \vec{\nabla}_x]u^*(\vec{y},\vec{x})d\Gamma + \int_\Omega \vec{\omega}(\vec{x}) \times \vec{\nabla}_x u^*(\vec{y},\vec{x})d\Omega + \\ + \mu^2 \int_\Omega u^*(\vec{y},\vec{x})\vec{v}(\vec{x},t_{n-1})d\Omega \,, \tag{6}$$

where $c(\vec{y})$ denotes the integral free term. The terms denoted with $()^*$ are the fundamental solutions, which are, as usual, defined as the solution of the governing equation with the Dirac distribution as source term, i.e., $b(\vec{x},t_n) = \delta(\vec{x}-\vec{y})$. The fundamental solution of the modified Helmholtz kinematic Eq. (5) and its normal derivative are

$$u^*(\vec{y},\vec{x}) = \frac{e^{-\mu r}}{4\pi r}, \quad q^*(\vec{y},\vec{x}) = \vec{n}(\vec{x}) \cdot \vec{\nabla} u^*(\vec{y},\vec{x}) \,, \tag{7}$$

with $r = |\vec{x}-\vec{y}|$. The fundamental solution $q^*(\vec{y},\vec{x})$ is the flux. In the case of large time steps, the parameter $\mu^2 \to 0$ and the fundamental solution in (7) tends to be the fundamental solution of the Laplace equation. It is important to mention that the parameter μ changes the shape of the fundamental solution [24]. However, while the parameter μ can enhance convergence it could also lead to stability issues. It is important to choose the optimal value of μ, whereas in the following $\mu < 20$ is used.

Additional to the integral Eq. (6) the tangential form of it is used in the subsequent solution procedure. It is obtained by applying the cross product of the normal vector at the source point $\vec{y}$ on (6). This results in

$$c(\vec{y})\vec{n}(\vec{y}) \times \vec{v}(\vec{y},t_n) + \vec{n}(\vec{y}) \times \int_\Gamma \vec{v}(\vec{x},t_n)q^*(\vec{y},\vec{x})d\Gamma = \vec{n}(\vec{y}) \times \int_\Gamma \vec{v}(\vec{x},t_n) \times [\vec{n} \times \vec{\nabla}_x]u^*(\vec{y},\vec{x})d\Gamma \\ + \vec{n}(\vec{y}) \times \int_\Omega \vec{\omega}(\vec{x},t_n) \times \vec{\nabla}_x u^*(\vec{y},\vec{x})d\Omega + \vec{n}(\vec{y}) \times \int_\Omega u^*(\vec{y},\vec{x})\mu^2\vec{v}(\vec{x},t_{n-1})d\Omega \quad \forall \vec{y} \in \Gamma. \tag{8}$$

The above equations are solved with a starting guess of the vorticity $\vec{\omega}$. To update this value an integral equation corresponding to the transport vorticity Eq. (1) is solved. Such an equation is obtained by applying Green's second identity on this equation and sorting the non-linear terms to the right hand side. This results in the boundary domain integral equation

$$c(\vec{y})\omega_j(\vec{y}) + \int_\Gamma \omega_j(\vec{x})q_L^*(\vec{y},\vec{x})d\Gamma = \int_\Gamma q_j^\omega(\vec{x})u_L^*(\vec{y},\vec{x})d\Gamma + \\ + Re \int_\Gamma \vec{n} \cdot \left[u_L^*(\vec{y},\vec{x})(\vec{v}\omega_j - \vec{\omega}v_j)\right] d\Gamma - Re \int_\Omega (\vec{v}\omega_j - \vec{\omega}v_j) \cdot q_L^*(\vec{y},\vec{x})d\Omega. \tag{9}$$

The fundamental solution $u_L^*(\vec{y},\vec{x})$ used in (9) is that of the Laplace operator. It can be obtained from (7) by setting the parameter $\mu = 0$.

The integral Eq. (9) is given in indical notation for better readability, where $j = 1, 2, 3$ and $q_j^\omega(\vec{x})$ is the flux component. After this step, the determined vorticity and velocity are used for the next iteration step, i.e., the three integral equations are solved again as described but with an updated vorticity and velocity field. This process is repeated until a prescribed precision is reached.

2.3. Spatial discretization of the integral formulation

For the spatial discretization of the above integral equations, first, the boundary is divided into N boundary elements and the domain Ω is divided into M domain cells

$$\Gamma = \bigcup_{i=1}^{N} \Gamma_i, \quad \Omega = \bigcup_{j=1}^{M} \Omega_j \,. \tag{10}$$

Second, the boundary data are approximated with continuous quadratic shape functions (9 nodes). The domain cells are approximated with a quadratic (27 nodes) interpolation. However, in this case a volume cell must be used, i.e., hexahedrons are assumed. The respective shape functions for the boundary element and domain cell are $\varphi(\vec{x})$, $\Phi(\vec{x})$. This results in

$$\vec{v}(\vec{x}) \approx \sum_{a=1}^{9} \vec{v}_a \varphi_a(\vec{x}), \qquad \vec{\omega}(\vec{x}) \approx \sum_{c=1}^{27} \vec{\omega}_c \Phi_c(\vec{x}). \tag{11}$$

Inserting these shape functions and the panelisation of the geometry in (6) results in the discrete integral equation

$$\sum_{a=1}^{n} \left[c(\vec{y})v_{ia}(\vec{x},t)\varphi(\vec{x}) + v_{ia}(\vec{x},t) \int_{\Gamma_a} \varphi(\vec{x})q^*(\vec{y},\vec{x})d\Gamma_a \right] = \\ \sum_{a=1}^{n} v_{ja}(\vec{x},t) \int_{\Gamma_a} \varphi(\vec{x}) \left[n_j \frac{\partial u^*(\vec{y},\vec{x})}{\partial x_i} - n_i \frac{\partial u^*(\vec{y},\vec{x})}{\partial x_j} \right] d\Gamma_a \\ - \sum_{a=1}^{n} v_{ka}(\vec{x},t) \int_{\Gamma_a} \varphi(\vec{x}) \left[n_k \frac{\partial u^*(\vec{y},\vec{x})}{\partial x_i} - n_i \frac{\partial u^*(\vec{y},\vec{x})}{\partial x_k} \right] d\Gamma_a + \sum_{c=1}^{m} \omega_k(\vec{x},t) \int_{\Omega_c} \Phi(\vec{x}) \frac{\partial u^*(\vec{y},\vec{x})}{\partial x_j} d\Omega_c \\ - \sum_{c=1}^{m} \omega_j(\vec{x},t) \int_{\Omega_c} \Phi(\vec{x}) \frac{\partial u^*(\vec{y},\vec{x})}{\partial x_k} d\Omega_c + \mu^2 \sum_{c=1}^{m} v_{ic}(\vec{x},t_{n-1}) \int_{\Omega_c} \Phi(\vec{x})u^*(\vec{y},\vec{x})d\Omega_c \quad \forall \vec{y} \in \Gamma. \tag{12}$$

The index a restricts the shape functions $\varphi(\vec{x})$ to the boundary element Γ_a, c restricts the shape function $\Phi(\vec{x})$ to the domain cell Ω_c. The upper limit of the first sum n indicates that this is the number of nodes on the boundary. In the last sum, m is the number of domain nodes. Hence, the sums over shape functions and nodes are collected into one sum. A collocation method is applied with the collocation points chosen at the nodes of the shape functions denoted as f. Based on these steps, from Eq. (12) evolves the discrete system of equations

$$[H]\{\vec{v}\} = \{\vec{v}\} \times [\vec{H}^t] + \{\vec{\omega}\} \times [\vec{D}] + \mu^2[B]\{\vec{v}_{n-1}\} \,, \tag{13}$$

with the elements of the matrices $[\vec{H}]$, $[\vec{H}^t]$, $[\vec{D}]$ and $[B]$

$$\begin{aligned} h_{fa} &= c(\vec{y}_f)\varphi_a(\vec{y}_f) + \int_{\Gamma_a} \varphi(\vec{x}) q^*(\vec{y}_f, \vec{x}) d\Gamma_a \,, \\ \vec{h}^t_{fa} &= \int_{\Gamma_a} \varphi(\vec{x}) [\vec{n} \times \vec{\nabla}_x] u^*(\vec{y}_f, \vec{x}) d\Gamma_a \,, \\ \vec{d}^t_{fc} &= \int_{\Omega_c} \boldsymbol{\Phi}(\vec{x}) \vec{\nabla}_x u^*(\vec{y}_f, \vec{x}) d\Omega_c \,, \\ b_{fc} &= \int_{\Omega_c} \boldsymbol{\Phi}(\vec{x}) u^*(\vec{y}_f, \vec{x}) d\Omega_c . \end{aligned} \tag{14}$$

The discretized form of (8)

$$[H]\left([\vec{n}] \times \{\vec{v}\}\right) = [\vec{n}] \times \{\vec{v}\} \times [\vec{H}^t] + [\vec{n}] \times \{\vec{\omega}\} \times [\vec{D}] + \mu^2 [\vec{n}] \times [B] \{\vec{v}_{n-1}\} . \tag{15}$$

is obtained in the same fashion. The elements in the respective matrices have the same form as presented in (14), while matrix $[\vec{n}]$ is a diagonal matrix with the elements of the normal vector $\vec{n}(\vec{y})$ on the main diagonal. To obtain the boundary vorticity the vector $\vec{\omega}$ is split into the boundary and inner domain part $\{\vec{\omega}\} = \{\vec{\omega}\}_\Gamma \cup \{\vec{\omega}\}_{\Omega/\Gamma}$. Reforming the term $[\vec{n}] \times [\vec{D}] \times \{\vec{\omega}\} = \left([\vec{n}] \cdot [\vec{D}]\right) \{\vec{\omega}\} - [\vec{D}] \{\vec{\omega}\}_n$ and rearranging equation (15) gives the final formulation

$$\begin{aligned} \left([\vec{n}] \cdot [\vec{D}]\right) \{\vec{\omega}\}_\Gamma = [\vec{D}] \{\vec{\omega}\}_n - \left([\vec{n}] \cdot [\vec{D}]\right) \{\vec{\omega}\}_{\Omega/\Gamma} - \left([\vec{n}] \cdot [\vec{H}^t]\right) \{\vec{v}\} + \\ + [\vec{H}^t] \{\vec{v}\}_n + [H]\left([\vec{n}] \times \{\vec{v}\}\right) - \mu^2 [\vec{n}] \times [B] \{\vec{v}_{n-1}\} , \end{aligned} \tag{16}$$

where $\{\vec{\omega}\}_n = \vec{n} \cdot \vec{\omega}$ and $\{\vec{v}\}_n = \vec{n} \cdot \vec{v}$ are the normal vorticitiy and velocity. The transport equation (9) is handled similarly. The integral equation (16) is solved to obtain the boundary vorticity $\{\vec{\omega}\}_\Gamma$. In the first iteration a guessed value for the domain values $\{\vec{\omega}\}_{\Omega/\Gamma}$ is used. Then the integral equation (15) is employed to find the velocity field $\{\vec{v}\}$. Lastly, the integral equation related to the transport vorticity equation (9) is used to solve the remaining unknown vorticity in the domain. With these domain vorticity values the procedure is repeated until a given precision is obtained. A detailed description of this formulation is presented in [29].

The integral equation (15) and the one corresponding to the transport vorticity equation (9) is solved using a subdomain technique, which is already an efficient technique presented in [30]. However, this method cannot be employed to solve the boundary vorticity with the integral equation (16). Hence, for this integral equation the $\mathcal{H}^2$-matrix technique is proposed.

2.4. Singular integrals

In the above integral equations singularities appear for $r = |\vec{x} - \vec{y}| \to 0$. Inspecting their behavior it is observed that the integrals in $[H]$, $[\vec{D}]$, $[B]$ are weakly singular but the integrals in $[\vec{H}^t]$ are strongly singular. The weakly singular integrals are solved using a transformation in polar coordinates for the surface integrals or spherical coordinates for the volume integrals (e.g., [31]). The integral free term $c(\vec{y})$ that is present in all integral formulations (6), (8) and (9) is solved with the formula from Mantič [32]. In principle the strongly singular integrals could be solved indirectly based on the rigid body technique [33]. However, this technique cannot be applied here as the integral kernels in the latter used $\mathcal{H}^2$-matrices are only exact in the near field (non-admissible blocks) but not in the rest of the matrix. Hence, the rigid body technique would suffer from the approximated matrix entries. Thus the strongly singular integrals have to be regularized. Guiggiani and Gigante [34] presented a technique for the elasticity problem. The same method is employed here due to the same structure of the strongly singular part of the integral kernel.

2.4.1. Strongly singular surface integrals

The strong singular surface integrals are the elements $\vec{h}^t_{fa}$ from (14) for the matrices $[\vec{H}^t]$. Evaluating the cross product or considering the third part in (12) expressions of the form

$$h^t_{ij} = \int_{\Gamma_a} \varphi(\vec{x}) \left[n_j \frac{\partial u^*(\vec{y}, \vec{x})}{\partial x_i} - n_i \frac{\partial u^*(\vec{y}, \vec{x})}{\partial x_j} \right] d\Gamma_a \tag{17}$$

can be identified. Comparing the structure of these terms with those handled in [34], it is clear that the proposed technique in this paper can be applied with some modifications. The principle steps are the same and will be shown next.

First, a transformation from cartesian to polar coordinates is introduced. Let us consider a surface element with the directions ξ_1 and ξ_2. A sketch of the polar transformation is illustrated in Fig. 1 (left). The polar coordinates

$$\xi_1 = \eta_1 + \rho \cos\vartheta, \quad \xi_2 = \eta_2 + \rho \sin\vartheta \,, \tag{18}$$

are used, where $\vec{\eta}$ are the coordinates of the collocation point $\vec{y}$ in the local coordinate system. Inserting the fundamental solution (7) and the coordinate transformation (18) into the kernel of the integral (17) results in the expression

$$\begin{aligned} F_{ij}(\rho, \vartheta) = \varphi(\vec{x}) \left[n_j(\vec{x}) \frac{\partial u^*(\vec{y}, \vec{x})}{\partial x_i} - n_i(\vec{x}) \frac{\partial u^*(\vec{y}, \vec{x})}{\partial x_j} \right] \rho = \\ = -\frac{e^{-\mu r}}{4\pi} \left(\frac{1}{r^2} + \frac{\mu}{r} \right) \left(n_j \frac{x_i - y_i}{r} - n_i \frac{x_j - y_j}{r} \right) \varphi(\vec{x}) \rho \,. \end{aligned} \tag{19}$$

By employing a Taylor expansion in the neighborhood of the singular point $\vec{\eta}$, the approximations and abbreviations

$$\begin{aligned} x_i - y_i = \rho \left[\frac{\partial x_i}{\partial \xi_1} \cos\vartheta + \frac{\partial x_i}{\partial \xi_2} \sin\vartheta \right] + \mathcal{O}(\rho^2) \,, \quad A_i(\vartheta) = \left[\frac{\partial x_i}{\partial \xi_1} \cos\vartheta + \frac{\partial x_i}{\partial \xi_2} \sin\vartheta \right] , \; i = 1, 2, 3, \\ A(\vartheta) = \left\{ \sum_{i=1}^{3} [A_i(\vartheta)]^2 \right\}^{1/2} , \quad r^2(\rho, \vartheta) = \rho^2 A^2(\vartheta) + \mathcal{O}(\rho^3) \,, \quad \frac{x_i - y_i}{r} = \frac{A_i(\vartheta)}{A(\theta)} + \mathcal{O}(\rho) \,, \end{aligned} \tag{20}$$

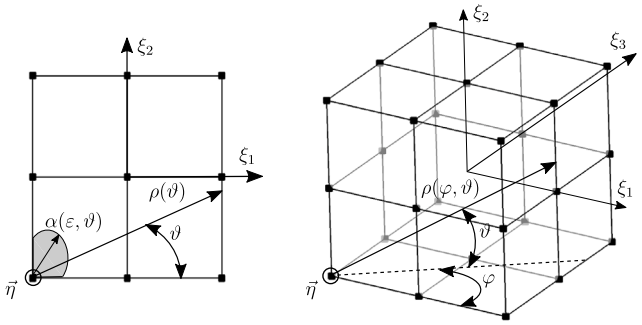


Fig. 1. The integration of the singular point in polar coordinates for surface integral (left) and spherical coordinates for volume integral (right). On the boundary element (right panel), the radius for the vanishing part $\alpha(\varepsilon,\vartheta)$ is illustrated [34].

can be found. Next, the radius $\alpha(\varepsilon,\vartheta)$ in the local panel (the gray area in Fig. 1) has to be determined. Following the above expression for $r^2(\rho,\vartheta)$ gives

$$\begin{aligned}\varepsilon^2 &= (x_k - y_k)(x_k - y_k) \rightarrow \varepsilon^2 = \rho^2 A^2(\vartheta) + \mathcal{O}(\rho^3)\\ \rho &= \alpha(\varepsilon,\vartheta) = \frac{\varepsilon}{A(\theta)} + \mathcal{O}(\varepsilon^2) = \varepsilon\beta(\vartheta) + \mathcal{O}(\varepsilon^2)\,,\end{aligned} \tag{21}$$

where ε is used later to shrink the gray area to zero. The first order terms in $F_{ij}(\rho,\theta)$ in the neighborhood of the singular point $\vec{\eta}$ can be given with the expressions from (20)

$$\begin{aligned}f_{ij}(\rho,\vartheta) &= \varphi(\vec{\eta})\left[n_j(\vec{\eta})\frac{\partial u^*(\vec{y},\vec{x})}{\partial x_i} - n_i(\vec{\eta})\frac{\partial u^*(\vec{y},\vec{x})}{\partial x_j}\right] =\\ &= -\frac{e^{-\mu\rho A(\vartheta)}}{4\pi}\left[\frac{n_j(\vec{\eta})A_i(\vartheta) - n_i(\vec{\eta})A_j(\vartheta)}{A^3(\vartheta)} + \rho\mu\left(\frac{n_j(\vec{\eta})A_i(\vartheta) - n_i(\vec{\eta})A_j(\vartheta)}{A^2(\vartheta)}\right)\right]\varphi(\vec{\eta}).\end{aligned} \tag{22}$$

Taking $\mu \rightarrow 0$ the same expression as in the original work [34] is obtained. Subtracting this first order representation of the kernel from F_{ij} gives an integrable function

$$F_{ij}(\rho,\vartheta) = \frac{1}{\rho}\left[f_{ij}(\vartheta) + \mathcal{O}(\rho)\right] \rightarrow F_{ij}(\rho,\vartheta) - \frac{1}{\rho}f_{ij}(\vartheta) = \mathcal{O}(1), \quad \rho \rightarrow 0\,. \tag{23}$$

For the integral (17) this results in two expressions

$$h^t_{ifa} = \int_{\vartheta_1}^{\vartheta_2}\int_0^{\rho(\vartheta)}\left[F_{ij}(\rho,\vartheta) - \frac{1}{\rho}f_{ij}(\vartheta)\right]d\rho d\vartheta + \lim_{\varepsilon\rightarrow 0}\int_{\vartheta_1}^{\vartheta_2}\int_{\alpha(\varepsilon,\vartheta)}^{\rho(\vartheta)}\frac{1}{\rho}f_{ij}(\vartheta)d\rho d\vartheta\,, \tag{24}$$

where the first can be integrated with standard Gaussian integration and the second has to be integrated analytically. This integral is split into two parts

$$\lim_{\varepsilon\rightarrow 0}\int_{\vartheta_1}^{\vartheta_2}\int_{\alpha(\varepsilon,\vartheta)}^{\rho(\vartheta)}\frac{1}{\rho}f_{ij}(\vartheta)d\rho d\vartheta = I_1 - I_2\,, \tag{25}$$

with the explicit expressions

$$\begin{aligned}I_1 &= \lim_{\varepsilon\rightarrow 0}\varphi(\vec{\eta})\int_{\vartheta_1}^{\vartheta_2}\int_{\alpha(\varepsilon,\vartheta)}^{\rho(\vartheta)} -\frac{e^{-\mu\rho A(\vartheta)}}{4\pi}\frac{1}{\rho}\left[\frac{n_j(\vec{\eta})A_i(\vartheta) - n_i(\vec{\eta})A_j(\vartheta)}{A^3(\vartheta)}\right]d\rho d\vartheta\,,\\ I_2 &= \lim_{\varepsilon\rightarrow 0}\varphi(\vec{\eta})\int_{\vartheta_1}^{\vartheta_2}\int_{\alpha(\varepsilon,\vartheta)}^{\rho(\vartheta)}\frac{\mu e^{-\mu\rho A(\vartheta)}}{4\pi}\left[\frac{n_j(\vec{\eta})A_i(\vartheta) - n_i(\vec{\eta})A_j(\vartheta)}{A^2(\vartheta)}\right]d\rho d\vartheta\,.\end{aligned} \tag{26}$$

Using partial integration results in

$$\begin{aligned}I_1 = \lim_{\varepsilon\rightarrow 0}\varphi(\vec{\eta})\int_{\vartheta_1}^{\vartheta_2}\left[\frac{n_j(\vec{\eta})A_i(\vartheta) - n_i(\vec{\eta})A_j(\vartheta)}{4\pi A^3(\vartheta)}\right]\Bigg[e^{-\mu\rho A(\vartheta)}\ln\rho(\vartheta) - e^{-\mu\alpha(\varepsilon,\vartheta)A(\vartheta)}\ln\alpha(\varepsilon,\vartheta)\\ + \mu A(\vartheta)\int_{\alpha(\varepsilon,\vartheta)}^{\rho(\vartheta)}e^{-\mu\rho A(\vartheta)}\ln\rho d\rho\Bigg]d\vartheta.\end{aligned} \tag{27}$$

The limit in (27) will be discussed for each of the three terms in the last bracket separately. The first term is independent from ε, hence it holds $\lim_{\varepsilon\rightarrow 0}e^{-\mu\rho A(\vartheta)}\ln\rho(\vartheta) = e^{-\mu\rho A(\vartheta)}\ln\rho(\vartheta)$. The second expression has to be rearranged

$$\begin{aligned}&\lim_{\varepsilon\rightarrow 0}\varphi(\vec{\eta})\int_{\vartheta_1}^{\vartheta_2}\left[\frac{n_j(\vec{\eta})A_i(\vartheta) - n_i(\vec{\eta})A_j(\vartheta)}{4\pi A^3(\vartheta)}\right]e^{-\mu\alpha(\varepsilon,\vartheta)A(\vartheta)}\ln\alpha(\varepsilon,\vartheta)d\vartheta =\\ &\varphi(\vec{\eta})\lim_{\varepsilon\rightarrow 0}e^{-\mu\varepsilon}\int_{\vartheta_1}^{\vartheta_2}\left[\frac{n_j(\vec{\eta})A_i(\vartheta) - n_i(\vec{\eta})A_j(\vartheta)}{4\pi A^3(\vartheta)}\right]\ln(\varepsilon\beta(\vartheta))d\vartheta =\\ &\varphi(\vec{\eta})\lim_{\varepsilon\rightarrow 0}e^{-\mu\varepsilon}\ln\varepsilon\int_{\vartheta_1}^{\vartheta_2}\left[\frac{n_j(\vec{\eta})A_i(\vartheta) - n_i(\vec{\eta})A_j(\vartheta)}{4\pi A^3(\vartheta)}\right]d\vartheta + \varphi(\vec{\eta})\int_{\vartheta_1}^{\vartheta_2}\left[\frac{n_j(\vec{\eta})A_i(\vartheta) - n_i(\vec{\eta})A_j(\vartheta)}{4\pi A^3(\vartheta)}\right]\ln\beta(\vartheta)d\vartheta\,,\end{aligned} \tag{28}$$

where $\alpha(\varepsilon,\vartheta)=\frac{\varepsilon}{A(\vartheta)}=\varepsilon\beta(\vartheta)$ is used, which is the linear part of the second line in (21). Higher order terms are neglected. The remaining term with a limit is zero due to symmetry conditions if all elements are considered, which are attached to the singular node. Let us assume that n elements k are connected to the singular node it holds

$$\sum_{k=1}^{n}\int_{\vartheta_k}^{\vartheta_{k+1}}\left[\frac{n_j(\vec{\eta})A_i(\vartheta)-n_i(\vec{\eta})A_j(\vartheta)}{4\pi A^3(\vartheta)}\right]d\vartheta=0. \tag{29}$$

This condition holds as long as the singular point is no free corner, i.e. the last element at this node closes up to the first element. This means that $\vartheta=[0,2\pi]$. This condition and a discussion about it can be found in [34,35].

The remaining expression is the third term in (27) with the integration over ϱ. To do this integration a substitution $\tau=\mu\rho A(\vartheta)$ is used $(\tau^*=\tau/(\mu A(\vartheta)),\alpha^*=\alpha/(\mu A(\vartheta)))$

$$\lim_{\varepsilon\to0}\int_{\alpha(\varepsilon,\vartheta)}^{\rho(\vartheta)}e^{-\mu\rho A(\vartheta)}\ln\rho d\rho=\frac{1}{\mu A(\vartheta)}\lim_{\varepsilon\to0}\left[\int_{\alpha^*(\varepsilon,\vartheta)}^{\tau^*}e^{-\tau'}\ln\tau' d\tau'-\ln(\mu A(\vartheta))\int_{\alpha*(\varepsilon,\vartheta)}^{\tau^*}e^{-\tau'}d\tau'\right]. \tag{30}$$

Now, the limit can be performed in the last integral, which causes the lower limit to be zero. An analytical solution of this term is straight forward. Further, the first integral is extended to infinity and the additional part is subtracted. These manipulations result in

$$\lim_{\varepsilon\to0}\int_{\alpha(\varepsilon,\vartheta)}^{\rho(\vartheta)}e^{-\mu\rho A(\vartheta)}\ln\rho d\rho=\frac{1}{\mu A(\vartheta)}\left[\int_0^{\infty}e^{-\tau'}\ln\tau' d\tau'-\int_{\tau^*}^{\infty}e^{-\tau'}\ln\tau' d\tau'+\ln(\mu A(\vartheta))\left[e^{-\rho(\vartheta)}-1\right]\right]. \tag{31}$$

The first integral can be identified as the negative Euler–Mascheroni constant $\gamma=0.5772156649$ and the second integral is solved using Mathematica to

$$\int_{\tau^*}^{\infty}e^{-\tau'}\ln\tau' d\tau'=e^{-\tau^*}\ln\tau^*+\Gamma(0,\tau^*),$$

with the upper incomplete Gamma function $\Gamma(0,\tau^*)$. Taking this result and transforming the variables back the result is

$$\lim_{\varepsilon\to0}\int_{\alpha(\varepsilon,\vartheta)}^{\rho(\vartheta)}e^{-\mu\rho A(\vartheta)}\ln\rho d\rho=\frac{1}{\mu A(\vartheta)}\left[-\gamma-e^{-\rho(\vartheta)}\ln\rho(\vartheta)-\Gamma(0,\rho(\vartheta))+\ln(\mu A(\vartheta))\left[e^{-\rho(\vartheta)}-1\right]\right]. \tag{32}$$

Collecting all intermediate steps the sought integral is

$$\begin{aligned}I_1=&\frac{\varphi(\vec{\eta})}{4\pi}\int_{\vartheta_1}^{\vartheta_2}\left[\frac{n_j(\vec{\eta})A_i(\vartheta)-n_i(\vec{\eta})A_j(\vartheta)}{A^3(\vartheta)}\right]\\&\left[e^{-\mu\rho(\vartheta)A(\vartheta)}\ln\rho(\vartheta)-\ln\beta(\vartheta)-\gamma-e^{-\rho(\vartheta)}\ln\left(\frac{\rho(\vartheta)}{\mu A(\vartheta)}\right)-\Gamma(0,\rho(\vartheta))-\ln(\mu A(\vartheta))\right]d\vartheta.\end{aligned} \tag{33}$$

The integration over ϑ can now be performed numerically with a standard Gauss–Legendre quadrature.

The second integral I_2 form (25) is regular thus a conventional Gauss–Legendre quadrature can be employed to solve the integral. However, also an analytical solution can be obtained resulting in

$$I_2=\frac{\varphi(\vec{\eta})}{4\pi}\int_{\vartheta_1}^{\vartheta_2}\left[\frac{n_j(\vec{\eta})A_i(\vartheta)-n_i(\vec{\eta})A_j(\vartheta)}{A^2(\vartheta)}\right][e^{-\mu A(\vartheta)\rho(\vartheta)}-1]d\vartheta. \tag{34}$$

It should be remarked that $\mu\to0$ results in the formula proposed by Guiggiani and Gigante in [34]. This can easily be checked by setting $\mu=0$ in (34) and in (27). Unfortunately, this is not possible in (33) as this representation of I_1 has been derived under the condition that $\mu\neq0$. Else the used coordinate transformation will not work.

2.4.2. Volume integrals

The volume integrals in (14) are weakly singular, hence, a coordinate transformation is sufficient to obtain numerically computable integrals. We transform the cartesian coordinates into spherical coordinates to solve the integrals. The spherical coordinates are

$$\xi_1=\eta_1+\rho\sin\vartheta\cos\varphi,\quad \xi_2=\eta_2+\rho\cos\vartheta\sin\varphi,\quad \xi_3=\eta_3+\rho\cos\vartheta, \tag{35}$$

where the radius ρ and angles ϑ, φ are shown in Fig. 1 on the sketch of the cell. The transformation of the coordinates takes place from the singular point with the coordinates $\vec{\eta}$.

Numerical tests have shown that the integration can be improved by using the generalized Telles' method for evaluating integrals [36]. This method narrows the Gaussian integral points to the singular point. The formulation

$$s_T=s_0+2^{1-q}(s-s_0)^q, \tag{36}$$

was implemented, where q is an odd integer which was set to 5, s is the Gaussian point, s_0 is the coordinate of the singular point in local coordinates and s_T is the new Gaussian point. In order to implement this in the integrals the radius ρ is evaluated for the new Gaussian point, while ϑ and φ are unchanged.

3. $\mathcal{H}$-Structure

To obtain a fast Boundary-Domain formulation, a sparse matrix formulation is proposed based on $\mathcal{H}^2$-matrices, where the integral kernel in the matrix is interpolated (see [37]). The matrices $[H]$, $[\vec{H}^t]$, $[\vec{D}]$ and $[B]$ in the modified Helmholtz kinematic equation (16) are approximated. First, an hierarchical partition of the geometry is introduced, i.e. an $\mathcal{H}$-matrix is formed, which is the basis to determine admissible and non-admissible blocks for approximations.

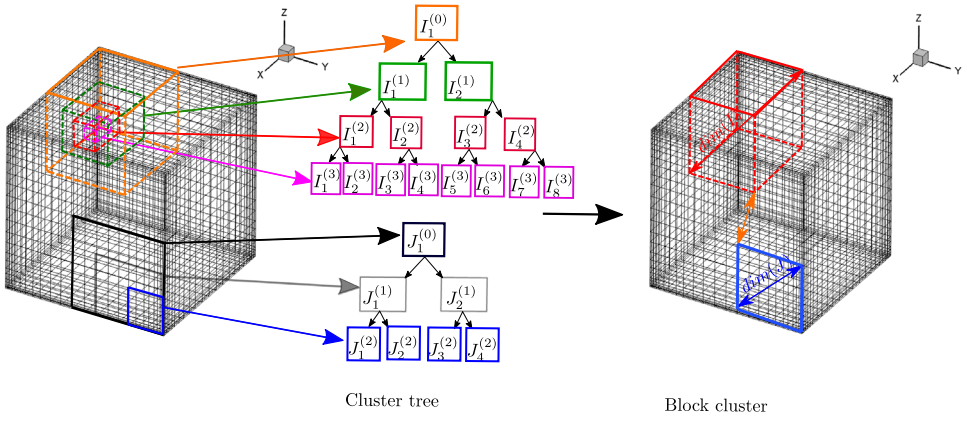


Fig. 2. Building the cluster tree (left) and the block cluster (right).

3.1. The cluster tree and block cluster tree

Cluster trees have to be formed as basis of the $\mathcal{H}$-matrix. The root cluster contains all boundary elements, which is then subdivided in 8 equally sized subblocks. These subblocks are then again subdivided. This procedure is repeated until a predefined level or a predefined size of the cluster, i.e. a predefined number of elements, is achieved. In each level the latter criterion is applied to stop subdivision of this specific cluster. For the proposed boundary domain integral formulation two kind of cluster trees are established. One for the boundary nodes and one for the domain nodes of the cells. A part of the cluster trees is illustrated in Fig. 2 on the left side, where J_j^p is the boundary cluster, I_j^p is the domain cluster and p is the level, while j is the cluster number. After the cluster trees are obtained, block clusters are formed. It depends on the matrix, which cluster trees are combined to form block cluster trees, e.g., for the boundary matrices $[H]$, $[\vec{H}^t]$ the clusters of two boundary cluster trees are combined and for the matrices $[\vec{D}]$, $[B]$ clusters from boundary and domain cluster trees are combined. Block clusters resemble smaller blocks of the matrix and are tested for admissibility. Exemplarily, let us consider the domain matrix $[B]$ with size $m \times n$ having block clusters of the size $\hat{m} \times \hat{n}$. The admissibility conditions

$$\begin{aligned} &\|[\hat{B}]_{\hat{m}\times\hat{n}}\| \leq 10^{-15}, \\ min\{dim(I), dim(J)\} \leq \eta\ dist(I,J) \quad \text{or} \quad &max\{dim(I), dim(J)\} \leq \eta\ dist(I,J), \\ &dist(I_i, J_j) \leq dist_m \end{aligned} \tag{37}$$

are applied. The first condition in (37) tests whether the entries in this block are such small that they can be neglected. If the Frobenius norm of the block cluster is smaller than the prescribed value, the block cluster is considered empty and is no longer used. The block clusters that are not empty are tested for admissibility. Two admissibility conditions were employed to observe which is optimal, where $dim(I), dim(J)$ are the diameters of the clusters, min and max are the minimal and maximal diameters, $dist(I,J)$ is the smallest distance between the two clusters and η is a user-defined parameter that determines the size of admissible block clusters. The last condition in (37) is somehow unusual. Based on the user-defined distance $dist_m$ it is decided whether admissible blocks are approximated by a standard ACA [9] or by interpolation with nested basis vectors, i.e. following the $\mathcal{H}^2$ approach. This criterion ensures that clusters of the domain near the boundary are approximated sufficiently good. More details and tests of this $\mathcal{H}$-structure can be found in [24].

3.2. Approximation of the integral kernel

Let us consider an admissible block cluster that fulfills the second condition form (37). The blocks in $[H]$ and $[\vec{H}^t]$ are from the boundary block cluster $J^k \times J^k$ and those in $[\vec{D}]$ and $[B]$ are from boundary-domain block cluster $J^k \times I^k$. If not the third criterion in (37) holds, the kernel functions $u^*(\vec{y}, \vec{x})$ in those blocks are interpolated

$$u^*(\vec{y}, \vec{x}) \approx \sum_{\iota=1}^{\alpha^3} \sum_{\kappa=1}^{\beta^3} \mathcal{L}_\iota(\vec{y})\, u^*(\vec{y}_\iota, \vec{x}_\kappa)\, \mathcal{L}_\kappa(\vec{x}) \ , \tag{38}$$

with Lagrange interpolation functions $\mathcal{L}(\vec{y})$, $\mathcal{L}(\vec{x})$ of order α, β evaluated at the zeros of the Tschebyschev nodes. In $[H]$, $[H_i^t]$ and $[D_i]$ the derivatives of the fundamental solution are present. In this case the derivative of the Lagrange polynomial has to be used, which gives the interpolations

$$q^*(\vec{y}, \vec{x}) \approx \sum_{\iota=1}^{\alpha^3} \sum_{\kappa=1}^{\beta^3} \mathcal{L}_\iota(\vec{y})\, u^*(\vec{y}_\iota, \vec{x}_\kappa)\, n_i \frac{\partial \mathcal{L}_\kappa(\vec{x})}{\partial x_i} \ , \quad \vec{q}_i(\vec{y}, \vec{x}) \approx \sum_{\iota=1}^{\alpha^3} \sum_{\kappa=1}^{\beta^3} \mathcal{L}_\iota(\vec{y})\, u^*(\vec{y}_\iota, \vec{x}_\kappa) \frac{\partial \mathcal{L}_\kappa(\vec{x})}{\partial x_i}. \tag{39}$$

Replacing the fundamental solution $u^*(\vec{y}, \vec{x})$ and its derivatives in (14) with the Lagrange interpolations from (38) and (39) gives the matrices

$$[\hat{H}] = [\hat{U}][\hat{S}][\hat{V}_H] \ , \qquad [\hat{H}_i^t] = [\hat{U}][\hat{S}][\hat{V}_{Hi}^t] \ , \qquad [\hat{D}_i] = [\hat{U}][\hat{S}][\hat{V}_{Di}] \ , \qquad [\hat{B}] = [\hat{U}][\hat{S}][\hat{V}_B]. \tag{40}$$

The matrix $[\hat{U}]$ is the Lagrange interpolation matrix on the boundary, $[\hat{S}]$ is the fundamental solution matrix and the $[\hat{V}]$ matrices consist of the Lagrange interpolation under the integral. A detailed explanation on the formulation is presented in [24]. The matrix $[\hat{S}]$ is stored in memory once for all the matrices. To further reduce the computational cost, nested cluster basis are employed

$$\mathcal{L}_\iota(\vec{x}) = \sum_{\lambda=1}^{\gamma^3} \mathcal{L}_\iota(\vec{x}_\lambda)\, \mathcal{L}'_\lambda(\vec{x}), \qquad \vec{n}(\vec{x}) \cdot \vec{\nabla}_x \mathcal{L}_\kappa(\vec{x}) = \sum_{\lambda=1}^{\gamma^3} \mathcal{L}_\kappa(\vec{x}_\lambda)\, (\vec{n}(\vec{x}) \cdot \vec{\nabla}_x \mathcal{L}'_\lambda(\vec{x})) \ , \tag{41}$$

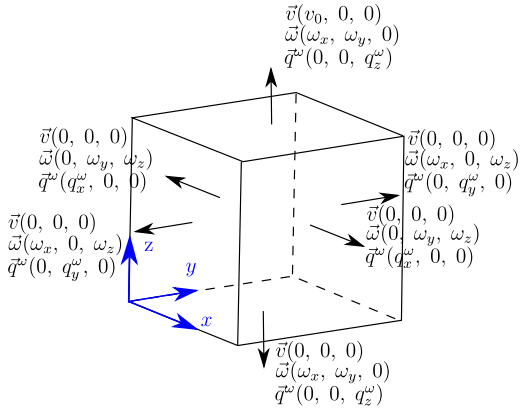


Fig. 3. The velocity, vorticity and vorticity flux boundary conditions for the 'lid-driven cavity'.

where $\mathcal{L}'(\vec{x})$ is again a Lagrange interpolation [21]. By employing the nested cluster basis, the matrices $[\hat{U}]$, $[\hat{V}_H]$, $[\hat{V}^t_{Hi}]$, $[\hat{V}_{Di}]$, $[\hat{V}_B]$ are transformed into the matrices $[T]$, $[T_H]$, $[T_{Hi}]$, $[T_{Di}]$. Details and studies on the parameters to be selected can as well be found in [24].

4. Lid-driven cavity, Hagen–Poiseuille and channel flow with cylindrical barrier test case

The proposed method is applied to three-dimensional fluid flow problems. Three exemplary test cases are selected, a lid-driven cavity, a pipe and a channel with a cylindrical barrier. For the lid-driven cavity the first, experimental studies have been performed by Kosseff and Street in [38] and numerical simulations can be found in [39], while for the pipe flow, analytical solutions can be used for comparison. The last example is the channel flow over a cylindrical barrier. This example is often employed by different authors, e.g., Kanaris et al. [40] and Ooi. [41] numerically simulated the turbulent flow and heat transfer around the cylinder in a channel, while [42] employed the test example to simulate blood flow around a thrombus. A benchmark test of it was performed by Shäfer and Turek in [43]. They have shown that the example can be solved by different numerical methods. For all cases observed in this research the fluid flow is solved in the laminar flow regime. To solve the boundary vorticity Eq. (16) the proposed $\mathcal{H}^2$-matrix is used for all boundary and domain matrices. To solve the linear system of equations GMRES with a simple Jacobi-preconditioner has been applied. For the subsequent presented tests, the error introduced by the $\mathcal{H}^2$-method is measured using a relative root mean square error defined by

$$RMS_v = \left(\frac{\sum_{i=1}^{n} (v_i - \tilde{v}_i)^2}{\sum_{i=1}^{n} (v_i)^2} \right)^{\frac{1}{2}}, \quad RMS_\omega = \left(\frac{\sum_{i=1}^{n} (\omega_i - \tilde{\omega}_i)^2}{\sum_{i=1}^{n} (\omega_i)^2} \right)^{\frac{1}{2}}. \tag{42}$$

In (42), ω_i, v_i are the absolute values of the vorticity and velocity at all nodes x_i that were solved with the original matrices formulation and $\tilde{\omega}_i, \tilde{v}_i$ denote the solution obtained with the $\mathcal{H}^2$-formulation. Hence, these values measures the error introduced due to the approximations in the $\mathcal{H}^2$-matrices in comparison to the dense matrix solution. But this error does not measure the overall quality of the solution.

4.1. Example: Lid driven cavity

The first test case considers a three-dimensional cubic geometry, where the top lid of the cube is driven at a velocity $v_x = v_0$ and all other walls have no-slip conditions. The coordinate system is located in the lower left corner such that $\{x, y, z\} \in [0, 1]$ holds. The selected boundary conditions for the velocity $\vec{v}$, vorticity $\vec{\omega}$ and vorticity flux q_j^{ω} are sketched in Fig. 3. The normal components of the boundary vorticity are set to zero whereas the tangential components are unknown. The boundary values of the vorticity flux, i.e., q_j^{ω} in (9), are selected opposite, i.e., the normal components are unknown and the tangential components are set to zero.

4.1.1. Validation with results from literature

We present the results obtained with the $\mathcal{H}^2$-technique in Fig. 4. The fluid flow was solved for Reynolds numbers 100, 400 and 1000 while the interpolation order for the $\mathcal{H}^2$-matrix was $\alpha = \beta = \gamma = 4$ and the ACA stopping condition was $\varepsilon = 10^{-8}$. The norm of the velocity $\vec{v}$ is presented for a slice of the domain at $y = 0.5$. The velocity norms $\vec{v}$ are very close to results obtained with a dense calculation, i.e., without the $\mathcal{H}^2$-technique. The top lid of the cavity has the maximum velocity, while the vortex of the fluid has the smallest velocity recognizable by the dark blue color. With increasing Reynolds number the vortex moves towards the center of the domain as also observed in the experiments in [38] and in the simulations by Yang et al. [39].

The x- and z-components of the velocity are plotted versus the x- and z-coordinate, respectively, in Fig. 5. Different to the colored plots in Fig. 4 the interpolation order for the $\mathcal{H}^2$-matrix formulation was changed. The different colored lines in Fig. 5 correspond to different interpolation orders of the kernel expansion. Increasing the interpolation order brings the results closer to the numerical results of Yang et al. [39]. This behavior is expected and confirms that the proposed formulation is able to produce the same results as other numerical techniques. In case of Yang et al. [39] it is an finite volume calculation.

The above shown results have been obtained with $\mu = 0$, i.e., the false transient approach is not used. Applying this technique improves the numerical behavior as presented in Fig. 6, where the same profiles as in Fig. 5 are presented for different values of μ. It can be observed that with much less interpolation order of the $\mathcal{H}^2$-matrices the sought solution is obtained. This effect can be attributed to the changed kernel function as an increase of μ adds a kind of damping which decreases the values of the kernel. Hence, more blocks are either not considered at all or can be easier approximated by lower order polynomials. Overall, it can be concluded that the $\mathcal{H}^2$-formulation gives sufficiently accurate results, which are comparable to the numerical results from literature.

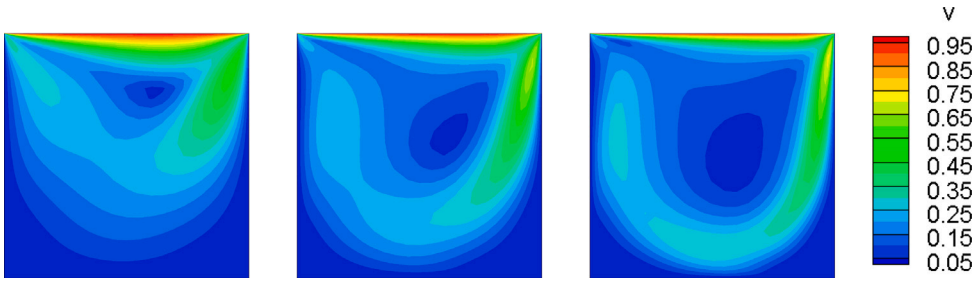


Fig. 4. The norm of the velocity $\vec{v}$ in the x-z-plane at $y = 0.5$ solved for Reynold's numbers 100 (left), 400 (middle) and 1000 (right). The domain mesh had 53^3 nodes and parameter $\mu = 0$ was selected.

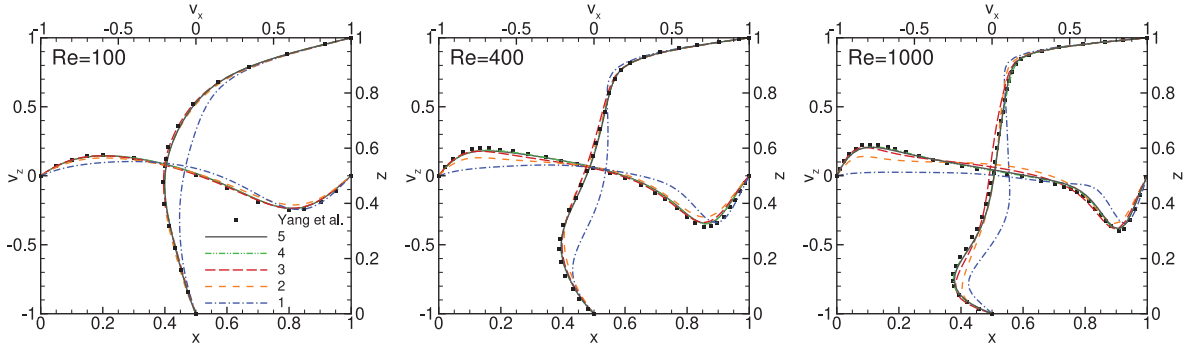


Fig. 5. The velocity profiles v_x and v_z along the respective x- and z-axes at $y = 0.5$ solved with different interpolation orders of the kernel expansion. Parameter $\mu = 0$ and domain mesh was 57^3.

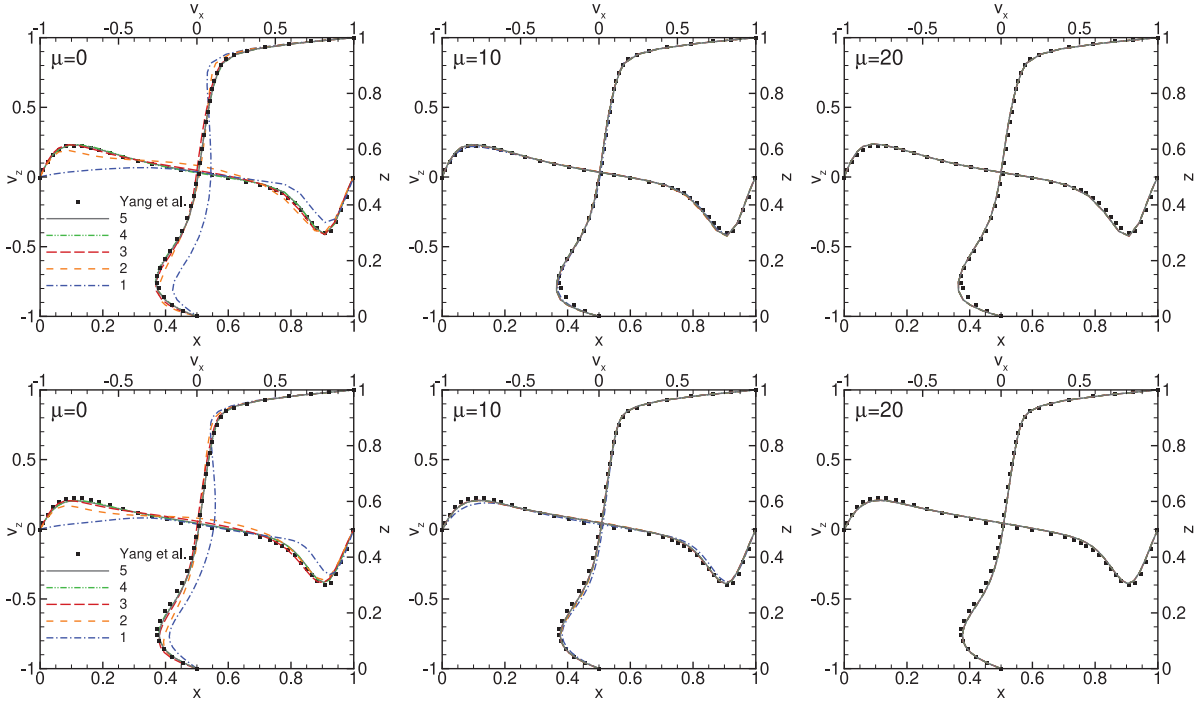


Fig. 6. The velocity profiles v_x and v_z along the respective x- and z-axes at $y = 0.5$ solved with different interpolation orders of the kernel expansion and different values of μ. The top line results are obtained with a domain mesh of 25^3 nodes, where the bottom line is for 53^3 domain nodes. The Reynolds number is 1000.

4.1.2. Behavior of GMRES

In this subsection, we investigate the convergence of the GMRES-solver with the Jacobian preconditioner. In Fig. 7, we present the number of GMRES iterations necessary to obtain the boundary vorticity ω_x from (16) in each iteration for two different meshes (upper and lower rows). The difference between the left and right column is the stopping condition ϵ^{GMRES}. The number of GMRES iterations decreases with the number of iterations to obtain convergence of the ω_x solution. The maximal number of iterations was 3000. Clearly, for $\epsilon^{GMRES} = 10^{-4}$ the GMRES solver

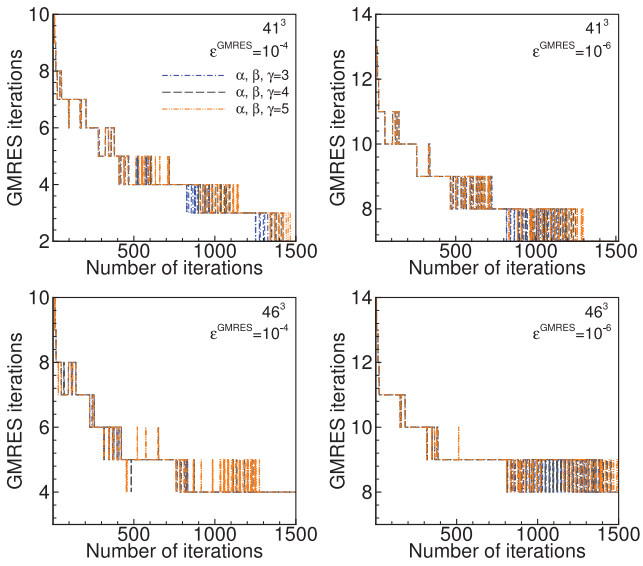


Fig. 7. Number of iterations depending on the interpolation order and GMRES stopping condition ϵ^{GMRES} with Reynolds number 1000.

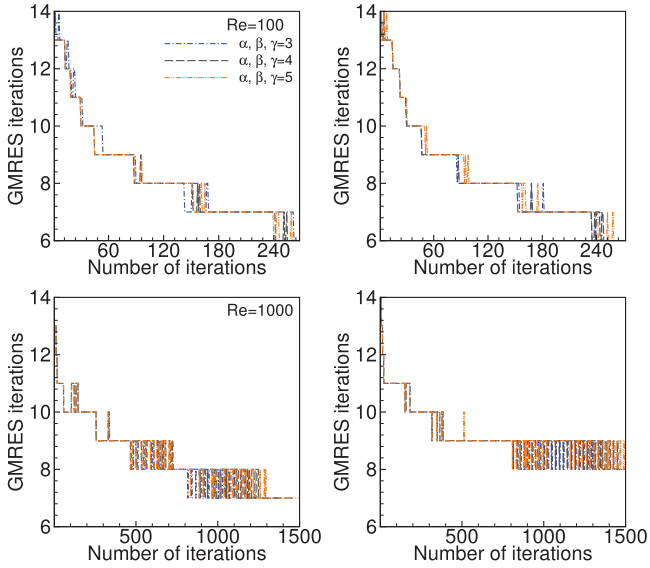


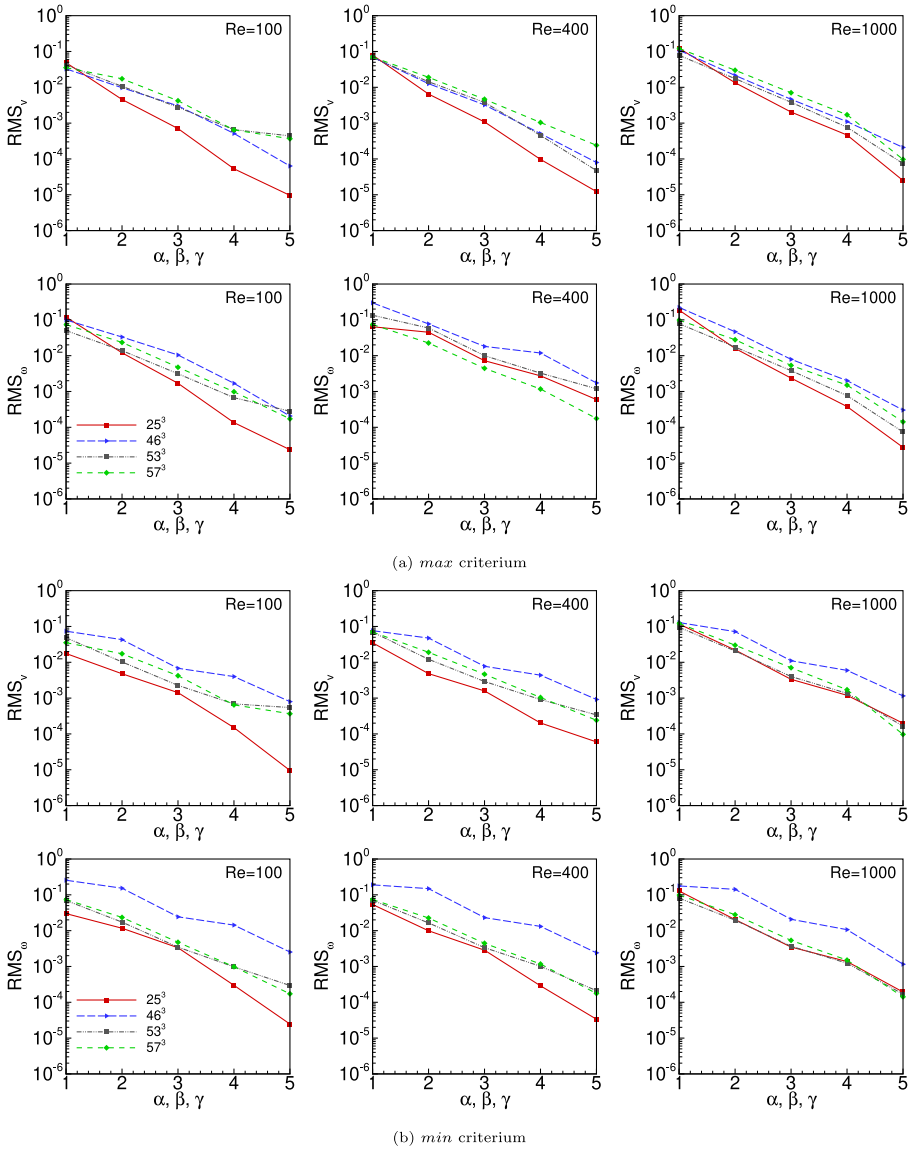
Fig. 8. Number of iterations depending on the interpolation order and Reynolds number for mesh densities 41^3 (left) and 46^3 (right) with $\epsilon^{GMRES} = 10^{-6}$.

needs less iterations than for $\epsilon^{GMRES} = 10^{-6}$. Further, in Fig. 8 the Reynolds number is varied. The number of iterations increases for a larger Reynolds number. However, the number of GMRES iterations did not change.

4.1.3. *Parameters of the $\mathcal{H}^2$-matrix*

Next, the different parameters introduced by the $\mathcal{H}^2$-matrix are studied using the RMS values defined in (42). Note again, this value measures the error introduced by the $\mathcal{H}^2$-matrix and not that of the solution. In Figs. 9(a) and 9(b) the RMS_v and RMS_ω are presented for different mesh densities and Reynolds numbers. The difference between the two subfigures is that in the first (i.e., Fig. 9(a)), the *max* and in the second (i.e., Fig. 9(b)), the *min* criteria from (37) are applied. It can be observed that increasing the interpolation order for the kernel interpolation improves the results independently. As well, this test show the expected result that the used mesh does not influence this tendency. Larger meshes increases the amount of admissible block clusters but this does not influence the error introduced by the kernel expansion. Comparing the error level for different Reynolds numbers shows that a higher Reynolds number requires an increase in the interpolation order. This is most probably related to the solution behavior visible in Fig. 4. For higher Reynolds numbers higher gradients are visible, which should require a better approximation. For this example and the used meshes there is no difference in the accuracy between the *min* and *max* admissibility condition in (37). However, in [24], it is observed that the *min* condition decreases the memory requirement.

In Fig. 10, the RMS_ω presented are obtained from two meshes with the same amount of nodes but different sized elements. The mesh on the left side in Fig. 10 denoted A has evenly sized elements, while the mesh on the right side denoted B has elements decreasing in size towards the edge of the domain. The latter is done for an improved computation of the boundary layer. The RMS_ω-values for mesh B are smaller which does not mean that the solution quality is better but the approximation of the $\mathcal{H}^2$-matrix introduces a smaller error. This effect is a consequence of the



(a) *max* criterium

(b) *min* criterium

Fig. 9. The RMS_v (top) and RMS_ω (bottom) for different mesh densities and Reynolds numbers. The parameters $\mu = 0$ and $\eta = 1$ are used. No ACA approximation was employed.

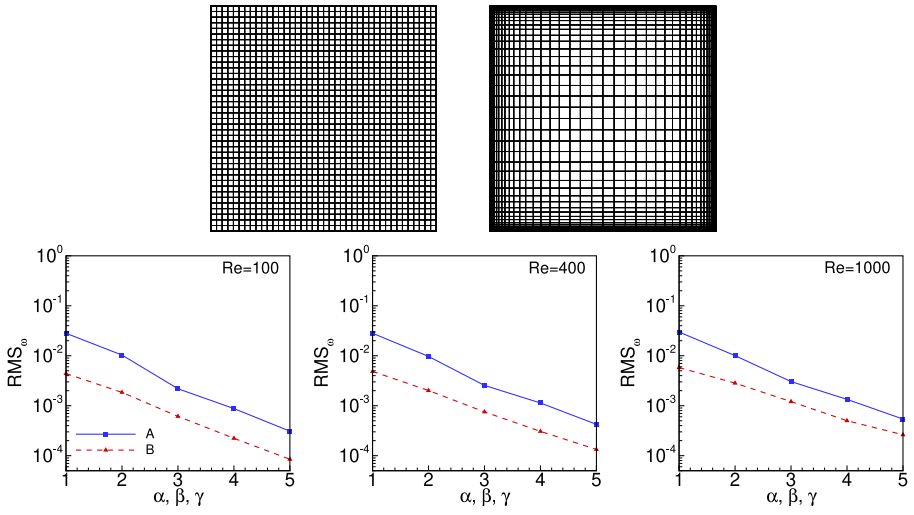


Fig. 10. The x-y cross section of two meshes (above), where both meshes have 41^3 nodes. The RMS_ω obtained for both meshes with the parameter $\mu = 0$.

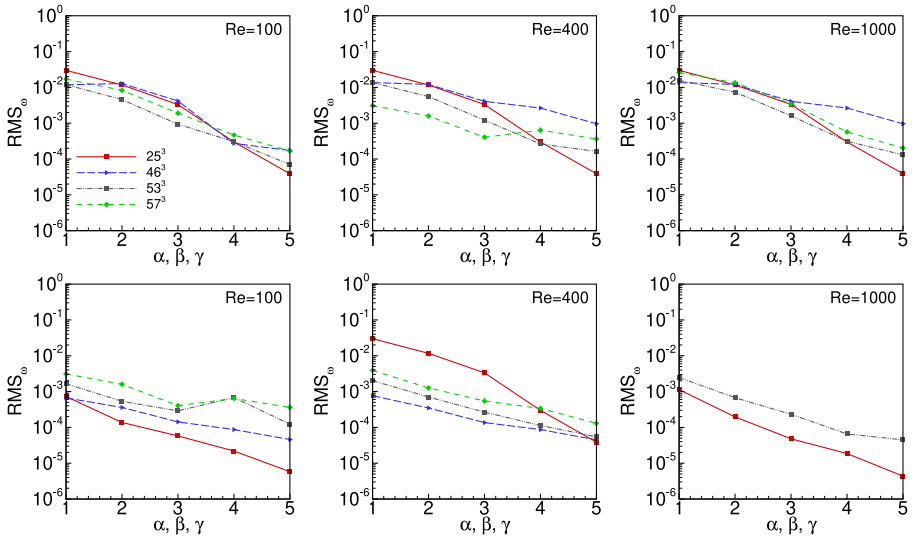


Fig. 11. The RMS_ω for the values of the parameter $\mu = 10$ (top) and 20 (below) at different Reynolds numbers and $\eta = 1$. The min criterium was employed.

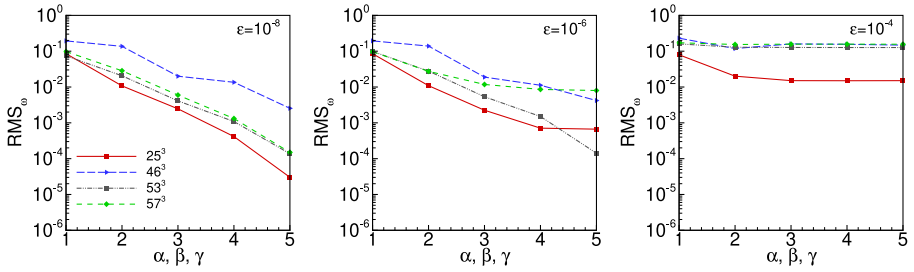


Fig. 12. RMS_ω for parameter $\mu = 0$ and Re=1000: Influence of the ACA stopping condition.

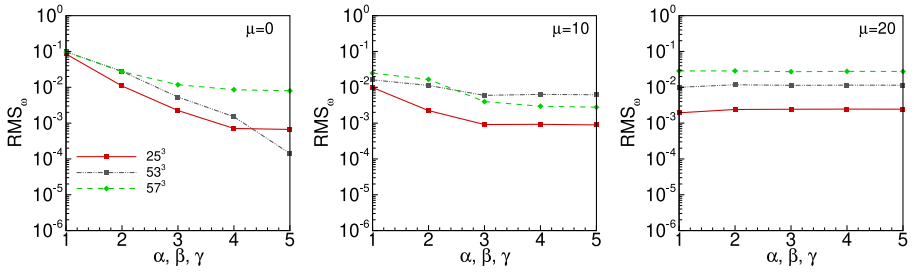


Fig. 13. RMS_ω for different μ and Re=1000. The ACA stopping condition was $\varepsilon = 10^{-6}$.

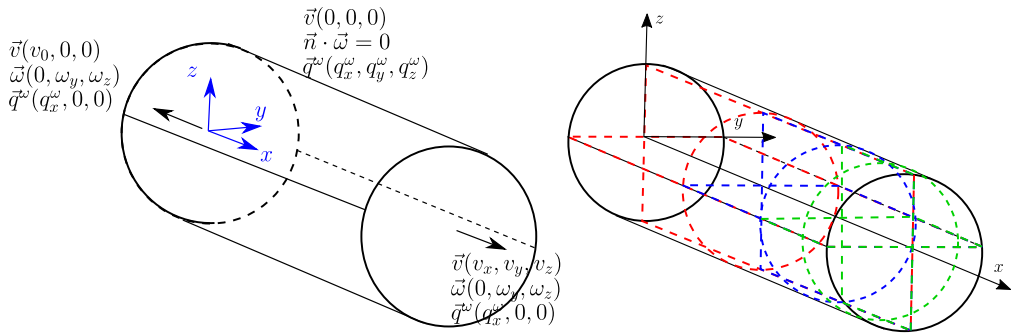
admissibility conditions used. There are more elements near the wall and, consequently, the amount of inadmissible block clusters increases. Thus less elements are approximated.

When the interpolation order of the approximation is increased the storage demand increases as well. Another method to improve the quality of the results is to change the parameter μ in the modified Helmholtz fundamental solution (7). This parameter changes the shape of the fundamental solution and its influence was already observed in Fig. 6. In Fig. 11, the RMS_ω obtained at different Reynolds numbers and three different values of μ are presented.

It is visible that for a larger μ the approximation accuracy increases, while the interpolation order does not change. Thus the memory demand does not increase for a fixed mesh density. In Fig. 11, the *min* admissibility criterium was used. It is remarked that similar results can be obtained for the *max* criterium not displayed here. More of these studies can be found in [24], where the approximation of the domain integral by $\mathcal{H}^2$-matrices is studied.

4.1.4. The ACA approximation

In the above proposed $\mathcal{H}^2$-matrix based methodology at two places the ACA is used to reduce the storage requirement. It is used in the $\mathcal{H}$-matrix blocks (third condition in (37)) and in the $\mathcal{H}^2$-method for the compression of matrix $[\hat{S}]$ in (40). In the above tests the latter compression is not included. Here, the influence of the precision/stopping criterion of ACA ε is studied. In Fig. 12, we present the RMS when the ACA is employed using different stopping conditions. Different to Fig. 12 in Fig. 13 the parameter of the false transient μ is changed and the ACA stopping condition has been fixed at $\varepsilon = 10^{-6}$.



(a) Geometry and boundary conditions (b) Partitioning of the domain

Fig. 14. The velocity, vorticity and vorticity flux boundary conditions for 'pipe flow' and the domain partitioning.

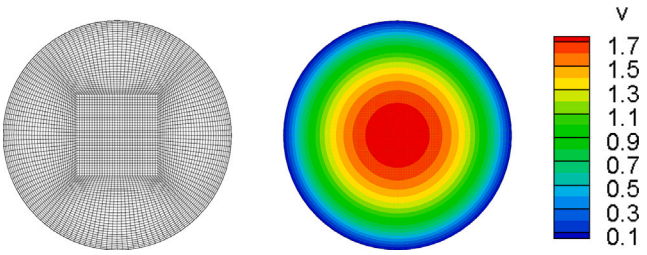


Fig. 15. The meshed outlet surface of the pipe (left) and solved absolute velocity on the outlet at $x = L$, $\alpha = \beta = \gamma = 4$ and $\epsilon = 10^{-8}$ (right).

The stopping condition of the ACA determines the rank of the low-rank matrix blocks. In both figures, it can be observed that the approximation quality stops at a distinct value independent of increasing the interpolation order. This effect shows simply that from this point on the error introduced by the ACA dominates. This holds true as well for the second test in Fig. 13 where the ACA precision was kept constant but the false transient parameter μ is increased. In the test above where no ACA has been used the increase of μ was beneficial. Here, it is only beneficial up to the point where the ACA error dominates.

4.2. Example: Hagen–Poiseuille flow in a pipe

The second test case is the Hagen–Poiseuille flow in a three-dimensional pipe. The solution can be obtained analytically from the in-compressible Navier–Stokes equations. For the geometry a cylinder is meshed that resembles the inner part of the pipe. The cylinder and the prescribed boundary conditions are sketched in Fig. 14(a). The coordinate system is placed in the middle of the inlet surface at $x = 0$. On the inlet surface the fluid enters the pipe with prescribed average velocity v_0 in x-direction. On the curved cylinder wall the normal vorticity is set to zero, $\omega_n = \vec{n}\cdot\vec{\omega} = 0$. In the test presented below, the diameter of the pipe was kept constant $D = 0.01$ m, while its length L has been varied such that the ratio L/D was changed from 1.0 to 4.0, i.e., the length of the pipe is $0.01\ \text{m} \leq L \leq 0.04\,\text{m}$.

The cylinder is a lengthy geometry in the x-direction. This makes it difficult to form a suitable cluster tree. To have a balanced $\mathcal{H}$-structure the domain is splited into substructures by bisection in all three directions. The subdomains evolving from this bisection have similar sizes in all three dimensions, see Fig. 14(b) for an example where the different colored boxes sketch the subdomains. For each subdomain a cluster tree is made and all cluster trees together give the final cluster tree. These substructures are not only used to obtain equal sized block cluster but also to simplify parallelization. Each substructure can be sent to each processor and forms its cluster tree that is used to obtain the block clusters for the $\mathcal{H}$-structure. The advantage is that the matrix–vector product is gathered from each processor to a root processor that can solve the linear system of equations. This technique is not only applied to the lengthy pipe structure but as well to the cube example to allow for parallel computing.

In Fig. 15, we show the meshed outlet surface of the pipe (left) and the absolute value of the velocity computed with the $\mathcal{H}^2$-formulation (right). To have a simple mesh generation process we used a square in the middle of the pipe. This enabled to have hexaeders as mesh elements for the domain cells. The red color stands for the maximal velocity and the blue for the minimal velocity $v = 0$ m/s. It should be reminded that the maximal velocity is two times larger than the prescribed average velocity v_0 in the inlet.

Similar studies concerning the approximation quality of the $\mathcal{H}^2$-matrix as above for the cube are performed next for this elongated structure. In Fig. 16, we show the RMS_v and RMS_ω for different Reynolds numbers. It shows that, as expected, the approximation accuracy increases for a higher interpolation order. As already presented for the Lid-driven cavity also in this example the Reynolds number has an impact on the approximation accuracy. When the Reynolds number is increased the interpolation order has to be increased.

Further, we observe the RMS_ω when the pipe length is changed. For that the length-diameter ratio L/D was changed from 1.0 to 4.0 while the mesh density was kept constant. In Figs. 17, the RMS_ω is shown for Reynolds numbers Re = 1 and Re = 100 and different length-diameter ratios L/D. The length of the pipe does not affect the approximation quality substantially. Only for the ratio $L/D = 1$ the RMS values are by a factor of 10 higher. This is, essentially, caused by the construction of the test, where the same meshes are used for all length-diameter ratios. This results for the longer pipe in larger elements, which is convenient for the approximation. It must be recalled that these results are only possible with the above introduced substructuring.

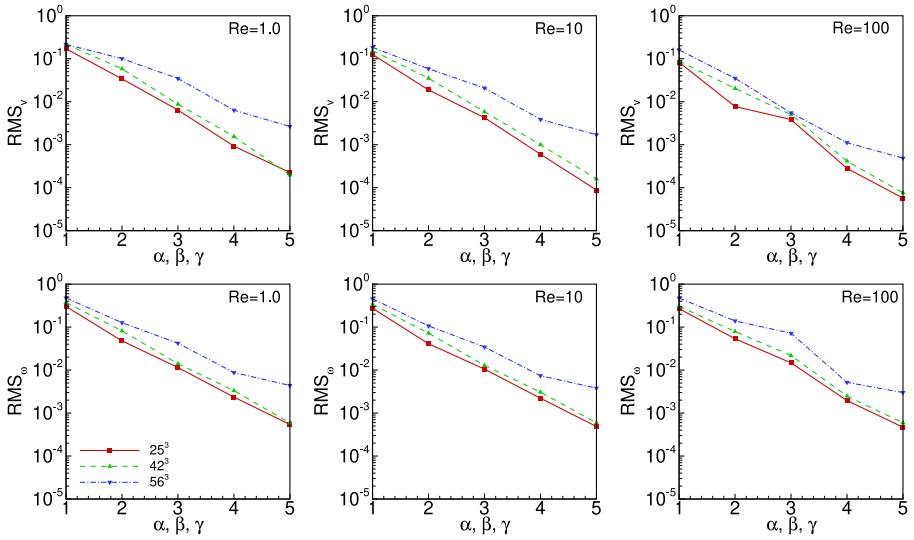


Fig. 16. The RMS_v (top) and RMS_ω (bottom) for parameter $\mu = 0$ at different Reynolds numbers. For the $\mathcal{H}$-structure the *min* criteria was employed with $\eta = 5$. The ACA stopping condition was set to $\varepsilon = 10^{-8}$. The length of the pipe was equal to its diameter $L/D = 1.0$.

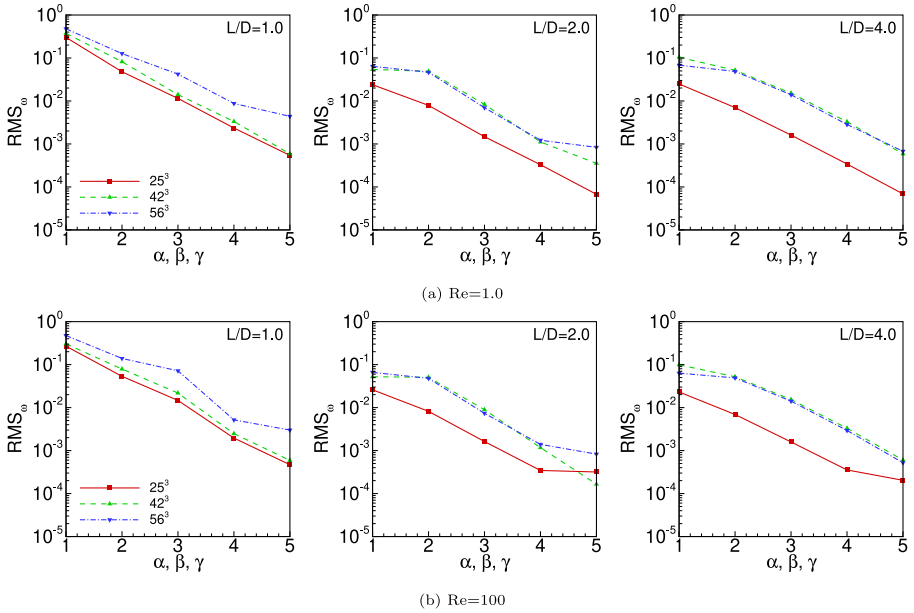


Fig. 17. The RMS_ω using the parameter $\mu = 0$. For the $\mathcal{H}$-structure the *min* criteria was employed. The ACA stopping condition was set to $\varepsilon = 10^{-8}$. The length of the pipe was changed while its diameter was kept constant.

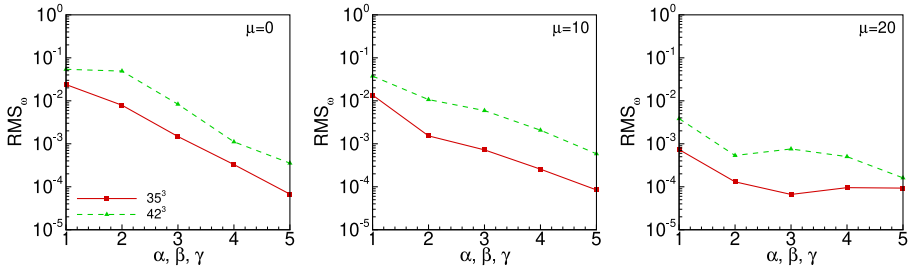


Fig. 18. The RMS_ω for different parameters μ at Re=1.0. For the $\mathcal{H}$-structure the *min* criteria was employed with $\eta = 5$. The ACA stopping condition was set to $\varepsilon = 10^{-8}$. The ratio was $L/D = 2.0$.

In Fig. 18, we show the approximation accuracy depending on μ. The tendency is the same as in the lid-driven case shown in Fig. 11. For larger values of μ the approximation error is reduced. The reasoning is again the smoother behavior of the fundamental solution. The observed results show that the proposed method can be employed on curved and lengthy geometries.

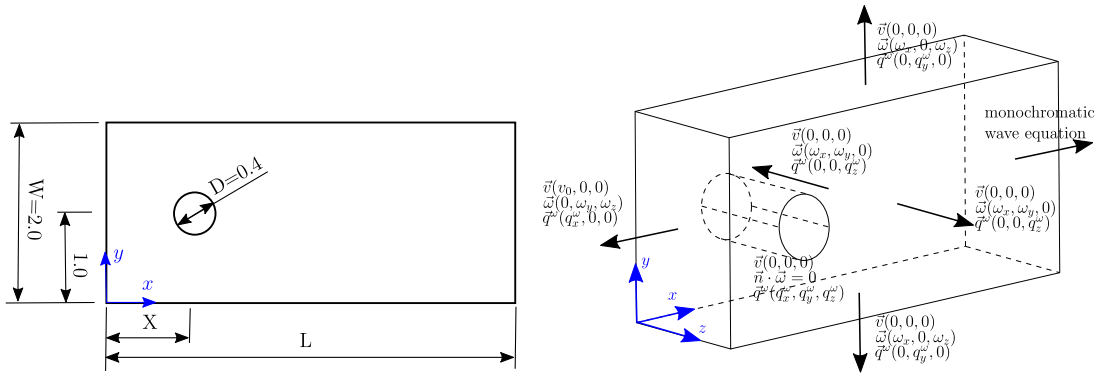


Fig. 19. Dimensions of the domain (left) and boundary conditions (right), where the hight of the domain is $H = 1.0$.

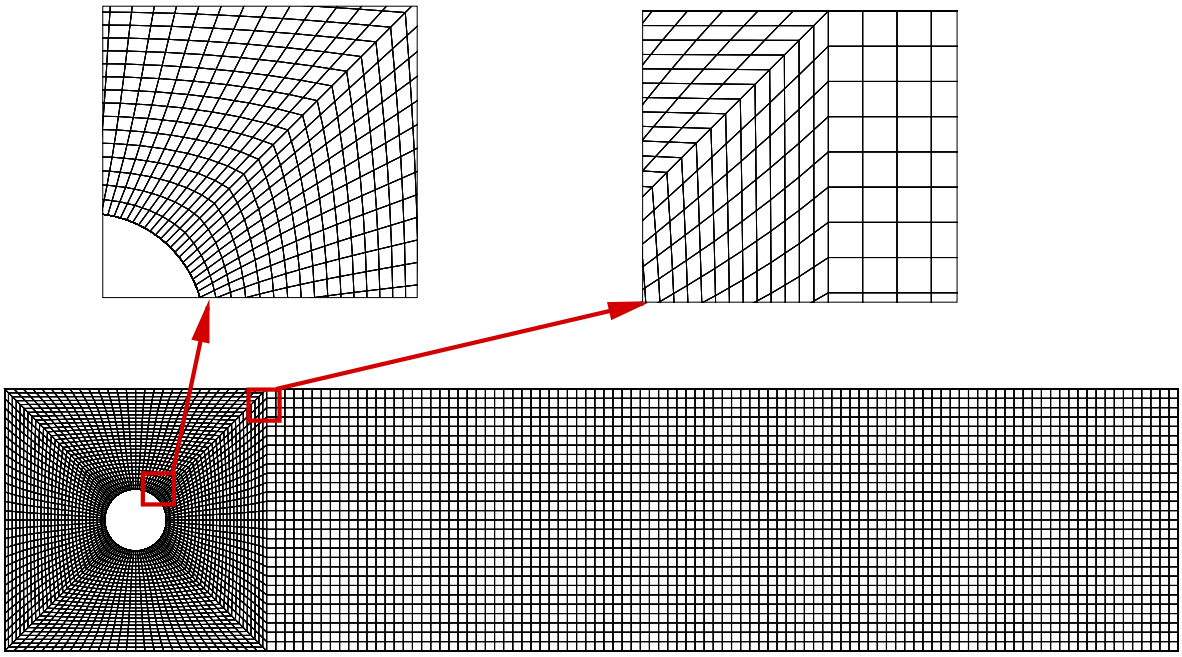


Fig. 20. Meshed channel for dimensions $X = 1.0$, $L = 10$ with 65^3 nodes.

4.3. Example: Channel flow with cylindrical barrier

For the last example, we present the channel flow over a cylindrical barrier. The intention of this example is to show that the proposed methodology can also be applied on more challenging geometries as those from above. The above reported experiences to adjust the parameters of the $\mathcal{H}^2$-approach are applied here.

The geometry with its dimensions is shown in Fig. 19 (left). Note, all dimensions are given in a dimensionless form. The cylindrical obstacle is placed at $X = 1.0$. The applied boundary conditions are given in Fig. 19 (right). The inlet boundary condition is on the plane $y-z$ at $x = 0$, with the prescribed inlet velocity $v_0 = 1.0$. The side and cylinder walls are no-slip walls. For the cylinder surface and the side walls a zero normal vorticity boundary condition $\vec{n} \cdot \vec{\omega} = 0$ is prescribed. To ensure that the vortices formed behind the cylinder can leave the computational domain without reflections, a convective outflow boundary condition is used. As presented in [44], such a condition can be obtained with a monochromatic wave equation.

This example simulated with other numerical methods can be found in [43] beside some other benchmark problems. Different to the benchmark [43], here, we consider a smaller domain to show the influence of the $\mathcal{H}^2$-approach. In Fig. 20, the meshed domain is presented. The mesh is formed with hexahedrons. Two red marked areas, one at the cylinder and the other further away from the cylinder, are magnified to show the mesh details in these locations. The elements around the cylinder are deformed, while further away from the cylinder the elements are perfectly shaped cubes.

To determine the convergence of the solution we compute the drag coefficient (see, e.g. [45])

$$C_D = \frac{(p_1 - p_2)W \cdot H}{\frac{1}{2}\rho A_\ell \cdot v_0}, \tag{43}$$

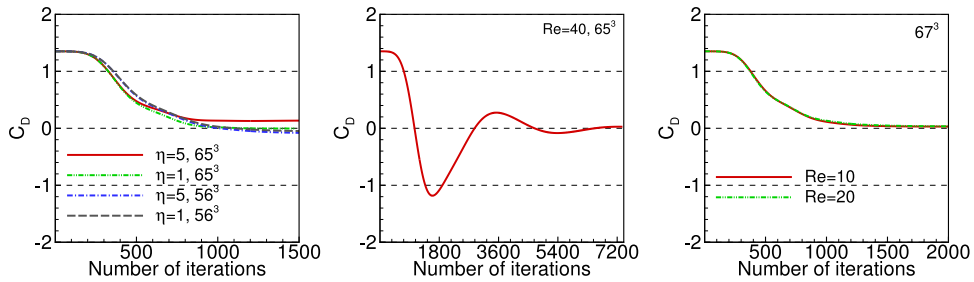


Fig. 21. The C_D coefficient: (left) for the domain $X = 1.0, L = 10$ at $Re = 10$ and different approximation parameter, (middle) for the domain $X = 1.0, L = 10$ at $Re = 40$ and $\eta = 1$, $\alpha = \beta = \gamma = 5$, (right) for the domain $X = 2.0$, $L = 12$ and $\eta = 1$, $\alpha = \beta = \gamma = 5$.

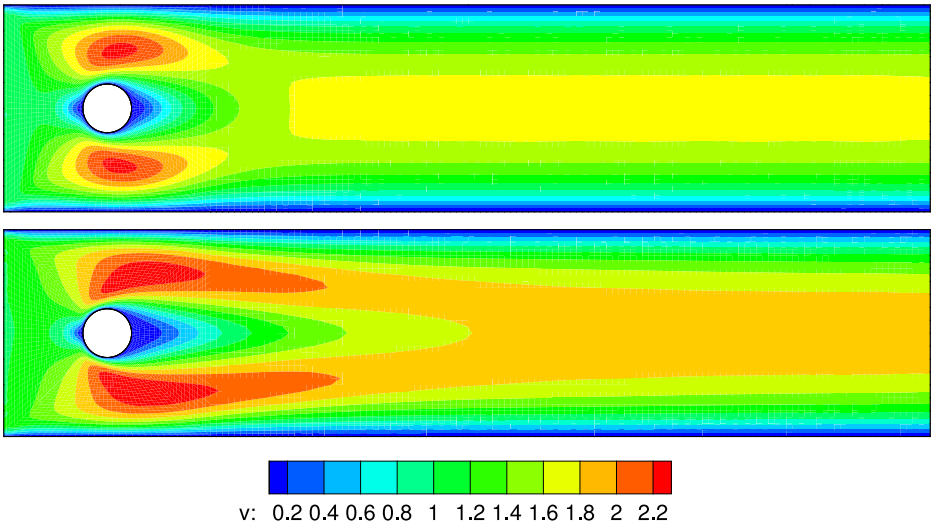


Fig. 22. The velocity contour for $Re = 10$ (top) and $Re = 40$ (bottom) for the domain length $X = 1$ and $L = 10$ using the approximation parameters $\eta = 1$, $\alpha = \beta = \gamma = 5$ and the mesh with 65^3 nodes.

which is the pressure difference between the inlet p_1 and outlet p_2. To obtain these the dynamic pressure formula $p = \frac{\rho v^2}{2}$ is used, where $v = \frac{1}{A}\int(\vec{n}\cdot\vec{v})dA$ is the average velocity on each surface. Further v_0 is the velocity on the inlet and $A_\ell = D \cdot H$ is the surface area of the cylinder cross section in the y-direction.

In Fig. 21 we show the drag coefficients C_D for different approximation parameters, Reynolds numbers and channel dimensions. On the left panel we investigate the influence of the presented $\mathcal{H}^2$-methodology and mesh density on the solution of C_D at $Re = 10$. Clearly, the approximation parameters influence the solution. For $\eta = 5$ and the interpolation order $\alpha = \beta = \gamma = 3$ (red or blue line) the C_D coefficient differs for different meshes, while for $\eta = 1$, $\alpha = \beta = \gamma = 5$ the drag coefficients of both meshes are similar. These results reveal that the choice $\eta = 1$ and $\alpha = \beta = \gamma = 5$ are good approximation parameters. Using this set of parameters the channel flow is solved for $Re = 40$ (middle panel in Fig. 21). The number of iterations for this Reynolds number is very high. However, the C_D converges to a constant value. On the right panel in Fig. 21 results are presented for changed channel dimensions $X = 2.0$, $L = 12$ and an increased mesh density of 67^3. The Reynolds number is varied from $Re = 10$ to $Re = 20$ resulting in similar drag coefficients. As already observed for the shorter channel the values of the drag coefficient are very small. It is important to note that the values of the drag coefficient depend on the channel dimensions as shown in [43,44].

Next, the velocity contours are presented. In Fig. 22, the absolute value of the velocity in the x-y-plane at $z = 0.5$ are given for two different Reynolds numbers. Differences are clearly visible, where the solution preserves the expected symmetry. Changing the dimensions of the channel in Fig. 23 as well the absolute value of the velocity of the fluid flow is shown for two Reynolds numbers. Comparing the velocity contours for $Re = 10$ between Figs. 22 and 23 shows a difference in the flow field around the cylinder. This is due to the different X-values, the distance between the inlet and the cylinder (see Fig. 19).

4.4. Computational costs

The aim of the presented method is to get a data-sparse representation of the underlying domain-boundary element formulation. I.e., the main argument was to save memory to store the matrices, which in a standard formulation increases quadratically with the unknowns. In Fig. 24, the memory usage is compared between three different cases.

In the first picture the storage for the formulation with and without ACA is compared. It shows that this compression pays for larger mesh sizes. The reason is the increased amount of admissible block clusters in larger meshes. Hence, there are more $\mathcal{H}^2$-blocks but as well more blocks with an ACA only compression. As well for the latter, the size is larger and, hence, the ACA is more effective. Further, the min and max criteria in the admissibility condition gives obviously different results, where the max criterion is more conservative from a storage point of view as well as from a mathematical point of view. The storage is presented in the middle picture. The third picture shows the influence of the parameter μ of the

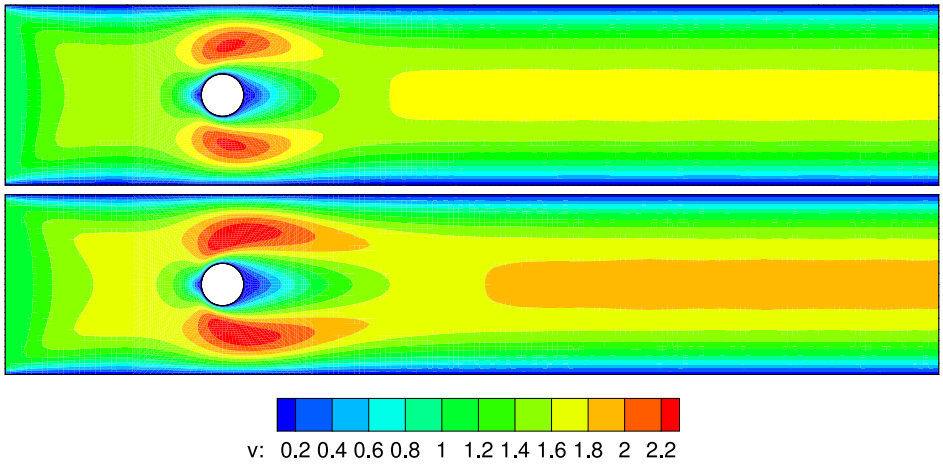


Fig. 23. The velocity contour for $Re = 10$ (top) and $Re = 20$ (bottom) for the domain length $X = 2$ and $L = 12$ using the approximation parameters $\eta = 1$, $\alpha = \beta = \gamma = 5$ and the mesh with 67^3 nodes.

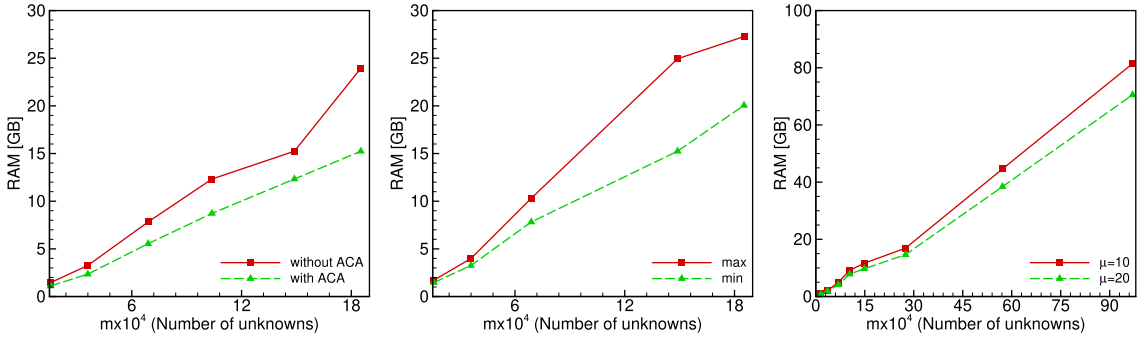


Fig. 24. Used memory of the boundary vorticity equation compared with and without ACA for $\mu = 0$ (left), the min and max criteria for the $\mathcal{H}$-structure (middle) and for the values of $\mu = 10$ and 20 (right). The ACA stopping condition was at 10^{-8} and the interpolation order $\alpha = \beta = \gamma = 3$.

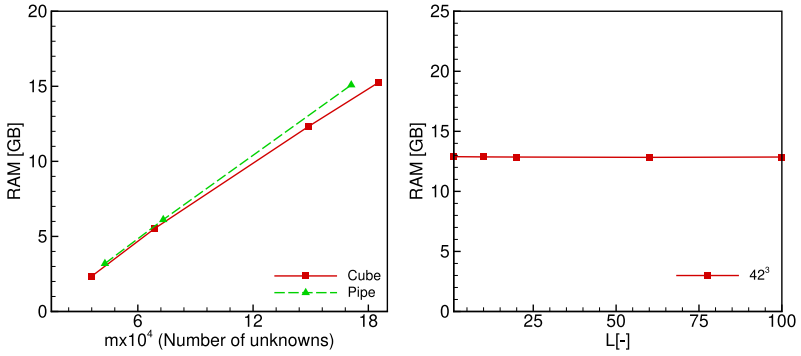


Fig. 25. Used memory for the pipe compared with the cube (left) and the memory for different pipe lengths. The ACA stopping condition was at 10^{-8} and the interpolation order $\alpha = \beta = \gamma = 3$.

false transient approach. There is a slight difference in the tendency showing that in the higher damped case (larger μ) more entries can be better approximated.

The above results are for the cube from the Lid-driven cavity test. In Fig. 25, we compare the memory used by the cube example with that of the pipe. This is shown on the left. The differences are small showing that the substructure technique works. This is further confirmed in the right picture in Fig. 25 where the memory usage is plotted for different pipe length but using the same mesh. It is confirmed that the approximation works in the same way also for the elongated structure.

There are no complexity estimates available for this specific application of $\mathcal{H}^2$-matrices but it can be expected that an almost linear behavior can be obtained. In Fig. 26, the used memory is plotted versus the number of unknowns. The black lines give indication of the orders $O(nm), \mathcal{O}(m \log m)$, and $\mathcal{O}(m)$. For the dense calculation, the green line, with n collocation points and m domain nodes the memory requirement scales with $O(nm)$. The other lines in blue, orange, and red are results for the $\mathcal{H}^2$-matrices. The notation $ACA - [U][S][V]$ indicates a computation where the $[U][S][V]$-matrices from (40) are stored and the nested basis is established during the matrix–vector product. The alternative implementation is denoted with $ACA - T$, where the $[T]$-matrices, i.e. the nested basis, are precomputed and stored instead of the $[U], [V]$-matrices. Obviously, the memory

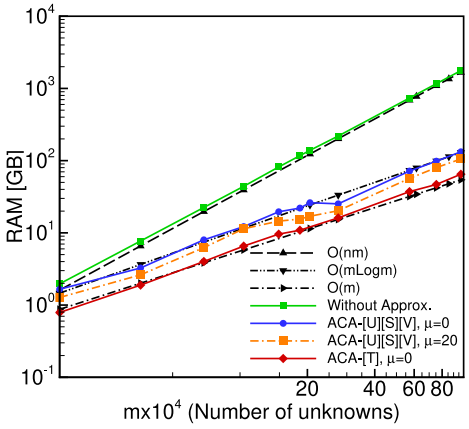


Fig. 26. The memory used to solve the boundary vorticity with and without approximation for a cube geometry. The ACA stopping condition is set to 10^{-8} and the interpolation order is $\alpha = \beta = \gamma = 4$.

demand drops when an approximation is employed. The complexity of the $ACA{-}[U][S][V]$ line indicates a linear logarithmic complexity $\mathcal{O}(m \log m)$, whereas the complexity of the $ACA - T$ version is almost linear $\mathcal{O}(m)$. Hence, the $ACA - T$ version is preferable from the storage view point but needs more computation time up to a factor of 2. Certainly, the latter is implementation dependent. Further, it can be observed that a larger factor μ reduces storage but does not change the complexity. To give some numbers: the quadratic complexity of the dense computation would results in 1.8 TB of memory to store all matrices for a mesh with around 10^6 nodes. With the nested cluster bases and ACA the memory is reduced to 50 GB. This is a reduction to ≈2%.

5. Conclusions

Laminar fluid flow was solved numerically with a fast Boundary-Domain Integral method. To reduce the computational cost the $\mathcal{H}^2$-methodology was employed. This method does not allow to use the rigid body method to compute the strong singular integrals. Hence, a method based on the technique of Guigiani and Gigante has been developed to handle the strong singular integrals. The main focus was on the application of $\mathcal{H}^2$-matrices to obtain a data sparse formulation. Different examples were solved to observe the influence of the approximation on the results. Especially, the interaction of the false transient approach and of the Reynolds number with the data sparse method has been studied. It can be concluded that the proposed technique is able to efficiently solve laminar fluid flow problems. Certainly, it must be remarked that the parameter of the proposed method must be carefully adjusted. For higher Reynolds numbers the interpolation order has to be increased, which decreases the efficiency of the method. It is not clear if the method could be employed to solve turbulent fluid flow. For that additional investigations of this approximation method are necessary. Further, parallel computation is not straight forward despite that the block clusters can be easily distributed on different processors. But the block cluster size is strongly varying and, consequently, a load balancing is difficult.

CRediT authorship contribution statement

J. Tibaut: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization. **J. Ravnik:** Validation, Software, Funding acquisition. **M. Schanz:** Writing – review & editing, Writing – original draft, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors acknowledge the financial support from the Slovenian Research Agency (research core funding No. P2-0196). The authors would like to acknowledge the use of HPC resources provided by the ZID of Graz University of Technology.

Data availability

No data was used for the research described in the article.

References

[1] Kuhn G, Skerget L, Hriberšek M. Computational fluid dynamics by boundary – Domain integral method. Internat J Numer Methods Engrg 1999;46(8):1291–311.
[2] Al-Bayati SA, Wrobel LC. Numerical modelling of convection-diffusion problems with first-order chemical reaction using the dual reciprocity boundary element method. Internat J Numer Methods Heat Fluid Flow 2021;32(5):1793–823.
[3] Guo S, Wu Q, Gu J, Wang W, Li X. An improved implementation of triple reciprocity boundary element method for three-dimensional steady state heat conduction problems. Eng Anal Bound Elem 2019;107:1–11.
[4] Fresneda-Portillo C. Boundary-domain integral equations for the diffusion equation in inhomogeneous media based on a new family of parametrices. Complex Var Elliptic Equ 2019;0–15.
[5] Fresneda-Portillo C, Woldemicheal Z. A new family of boundary-domain integral equations for the Dirichlet problem of the diffusion equation in inhomogeneous media with H-1(ω) source term on Lipschitz domains. Math Methods Appl Sci 2021;44(February 2020):9817–30.
[6] Mikhailov SE, Mohamed NA. Numerical solution and spectrum of boundary-domain integral equation for the Neumann BVP with a variable coefficient. Int J Comput Math 2012;89(11):1488–503.
[7] Mikhailov SE, Portillo CF. Boundary-domain integral equations equivalent to an exterior mixed bvp for the variable-viscosity compressible stokes pdes. Commun Pure Appl Anal 2021;20(3):1103–33.
[8] Hackbusch W. Sparse matrix arithmetic based on H-matrices. Part I: Introduction to H-matrices. Computing 1999;62(2):89–108.
[9] Bebendorf M, Grzibovski R. Accelerating Galerkin BEM for linear elasticity using adaptive cross approximation. Math Methods Appl Sci 2006;29(14):1721–47.
[10] Kalman D. A singularly valuable decomposition: The SVD of a matrix. College Math J 1996;27(1):2.
[11] Rokhlin V. Diagonal forms of translation operators for the Helmholtz equation in three dimensions. 1992.
[12] Darve E. The fast multipole method: Numerical implementation. J Comput Phys 2000;160(1):195–240.
[13] Messner M, Schanz M, Tausch J. An efficient Galerkin boundary element method for the transient heat equation. SIAM J Sci Comput 2014;37(3):1554–76.
[14] Jelich C, Zhao W, Chen H, Marburg S. Fast multipole boundary element method for the acoustic analysis of finite periodic structures. Comput Methods Appl Mech Engrg 2022;391:114528.
[15] Li J, Chen W, Qin Q. A modified dual-level fast multipole boundary element method based on the Burton – Miller formulation for large-scale three-dimensional sound field analysis. Comput Methods Appl Mech Engrg 2018;340:121–46.
[16] Sellountos EJ. A single domain velocity–vorticity fast multipole boundary domain element method for two dimensional incompressible fluid flow problems. Eng Anal Bound Elem 2019;106:359–70.
[17] Sellountos EJ. A single domain velocity - vorticity Fast Multipole Boundary Domain Element Method for three dimensional incompressible fluid flow problems, part II. Eng Anal Bound Elem 2020;114:74–93.
[18] Wang Y, Wang Q, Deng X, Xia Z, Yan J, Xu H. Graphics processing unit (GPU) accelerated fast multipole BEM with level-skip M2L for 3D elasticity problems. Adv Eng Softw 2015;82:105–18.
[19] Nishimura N. Fast multipole accelerated boundary integral equation methods. Appl Mech Rev 2002;55(4):299–324.
[20] Börm S, Hackbusch W. H2-matrix approximation of integral operators by interpolation. Appl Numer Math 2002;43(1–2):129–43.
[21] Börm S. Approximation of integral operators by H2-matrices with adaptive bases. Computing 2005;74(3):249–71.
[22] Chai W, Jiao D. H- and H2-matrix-based fast integral-equation solvers for large-scale electromagnetic analysis. IET Microw Antennas Propag 2010;4(10):1583–96.
[23] Börm S. Directional H2-matrix compression for high-frequency problems. Numer Linear Algebra Appl 2017;24(6):1–28.
[24] Tibaut J, Schanz M, Ravnik J. Fast Boundary-Domain Integral Method with the H2-matrix formulation for large scale numerical investigations. Eng Anal Bound Elem 2022;138:1–12.
[25] Guj G, Stella F. A vorticity-velocity method for the numerical solution of 3D incompressible flows. J Comput Phys 1993;106(2):286–98.
[26] Behnia M, Stella F, Guj G. A numerical study of three-dimensional combined buoyancy and thermocapillary convection. Int J Multiph Flow 1995;21(3):529–42.
[27] Škerget L, Hriberšek M, Žunič Z. Natural convection flows in complex cavities by BEM. Internat J Numer Methods Heat Fluid Flow 2003;13(5–6):720–35.
[28] Tibaut J, Ravnik J. Fast boundary-domain integral method for heat transfer simulations. Eng Anal Bound Elem 2019;99:222–32.
[29] Ravnik J, Škerget L. A numerical study of nanofluid natural convection in a cubic enclosure with a circular and an ellipsoidal cylinder. Int J Heat Mass Transfer 2015;89:596–605.
[30] Ravnik J, Škerget L, Žunič Z. Combined single domain and subdomain BEM for 3D laminar viscous flow. Eng Anal Bound Elem 2009;33(3):420–4.
[31] Itagaki M, Sahashi N, Itagaki M, Keikaku K. Three-dimensional isoparametric boundary element method for solving neutron diffusion equations three-dimensional isoparametric boundary element method for solving neutron diffusion equations. J Nucl Sci Technol 1996;33:7–16.
[32] Mantič V. A new formula for the C-matrix in the somigliana identity. J Elasticity 1993;33(3):191–201.
[33] Guiggiani M, Casalini P. Rigid-body translation with curved boundary elements. Appl Math Model 1989;13(2):365–8.
[34] Guiggiani M, Gigante A. A general algorithm for multidimensional Cauchy principal value integrals in the boundary element method. Trans ASME 1990;57(December):906–15.
[35] Guiggiani M, Krishnasamy G, Rudolphi TJ, Rizzo FJ. A general algorithm for the numerical solution of hypersingular boundary integral equations. J Appl Mech ASME 1992;59:604–14.
[36] Johnston PR, Elliott D. A generalisation of Telles' method for evaluating weakly singular boundary element integrals. J Comput Appl Math 2001;131(1–2):223–41.
[37] Börm S. Efficient numerical methods for non-local operators, In: EMS tracts in mathematics, second ed. vol. 14, Zuerich, Switzerland: European Methematical Society Publishing House; 2013.
[38] Koseff JR, Street RL. The lid-driven cavity flow: A synthesis of qualitative and quantitative observations. J Fluids Eng 1984;106(4):390–8.
[39] Yang JY, Yang SC, Chen YN, Hsu CA. Implicit weighted ENO schemes for the three-dimensional incompressible Navier-Stokes equations. J Comput Phys 1998;146(1):464–87.
[40] Kanaris N, Grigoriadis D, Kassinos S. Three dimensional flow around a circular cylinder confined in a plane channel. Phys Fluids 2011;23(6).
[41] Ooi A, Lu W, Chan L, Cao Y, Leontini J, Skvortsov A. Turbulent flow over a cylinder confined in a channel at Re = 3,900. Int J Heat Fluid Flow 2022;96(February).
[42] Mukherjee D, Shadden SC. Modeling blood flow around a thrombus using a hybrid particle–continuum approach. Biomech Model Mechanobiol 2018;17(3):645–63.
[43] Schäfer M, Turek S, Durst F, Krause E, Rannacher R. Benchmark computations of laminar flow around a cylinder. In: Hirschel EH, editor. Flow simulation with high-performance computers II: DFG priority research programme results 1993–1995. Wiesbaden: Vieweg+Teubner Verlag; 1996, p. 547–66.
[44] Ravnik J, Škerget L, Hriberšek M, Žunič Z. Numerical simulation of dilute particle laden flows by wavelet BEM-fem. Comput Methods Appl Mech Engrg 2008;197(6–8):789–805.
[45] Sharma B, Barman RN. Steady laminar flow past a slotted circular cylinder. Phys Fluids 2020;32(7).