INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng (2008)
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme.2467

Fast single domain–subdomain BEM algorithm for 3D incompressible fluid flow and heat transfer

Jure Ravnik*,†, Leopold Škerget and Zoran Žunič

Faculty of Mechanical Engineering, University of Maribor, Smetanova 17, 2000 Maribor, Slovenia

SUMMARY

In this paper acceleration and computer memory reduction of an algorithm for the simulation of laminar viscous flows and heat transfer is presented. The algorithm solves the velocity–vorticity formulation of the incompressible Navier–Stokes equations in 3D. It is based on a combination of a subdomain boundary element method (BEM) and single domain BEM. The CPU time and storage requirements of the single domain BEM are reduced by implementing a fast multipole expansion method. The Laplace fundamental solution, which is used as a special weighting function in BEM, is expanded in terms of spherical harmonics. The computational domain and its boundary are recursively cut up forming a tree of clusters of boundary elements and domain cells. Data sparse representation is used in parts of the matrix, which correspond to boundary-domain clusters pairs that are admissible for expansion. Significant reduction of the complexity is achieved.

The paper presents results of testing of the multipole expansion algorithm by exploring its effect on the accuracy of the solution and its influence on the non-linear convergence properties of the solver. Two 3D benchmark numerical examples are used: the lid-driven cavity and the onset of natural convection in a differentially heated enclosure. Copyright © 2008 John Wiley & Sons, Ltd.

Received 21 April 2008; Accepted 7 August 2008

KEY WORDS: boundary element method; fast multipole method; fluid flow; heat transfer; velocity–vorticity

1. INTRODUCTION

Development of numerical algorithms capable of simulating fluid flow and heat transfer has a long-standing tradition. A vast variety of methods was developed and their characteristics were examined. In this paper we present a boundary element method (BEM)-based numerical technique, which is capable of simulating laminar flow and heat transfer in 3D. The BEM uses the fundamental solution of the differential operator and Green's theorem to rewrite a partial differential equation

*Correspondence to: Jure Ravnik, Faculty of Mechanical Engineering, University of Maribor, Smetanova 17, 2000 Maribor, Slovenia.
†E-mail: jure.ravnik@uni-mb.si

into an equivalent boundary integral equation. After discretization of only the boundary of the problem domain, a fully populated system of equations emerges. The number of degrees of freedom is equal to the number of boundary nodes. This reduction in the dimensionality of the problem is a major advantage over the volume-based methods. Fundamental solutions are known for a wide variety of differential operators [1], making BEM applicable for solving a wide range of problems.

Unfortunately, integral equations of nonhomogeneous and non-linear problems, such as fluid flow, include a domain term. In this paper, we solve the velocity–vorticity formulation of incompressible Navier–Stokes equations. The formulation joins the Poisson type kinematics equation with diffusion-advection-type equations of vorticity and heat transport. These equations are nonhomogenous and non-linear. In order to write discrete systems of linear equations for such equations, matrices of domain integrals must be evaluated. Such domain matrices, since they are full and unsymmetrical, require a lot of storage space and algebraic operations with them require a lot of CPU time. Thus, the domain matrices present a bottleneck for any BEM-based algorithm effectively limiting the maximal usable mesh size through their cost in storage and CPU time.

The dual reciprocity BEM [2] is one of the most popular techniques to eliminate the domain integrals. It uses expansion of the nonhomogenous term in terms of radial basis functions. Several other approaches that enable construction of data sparse approximations of fully populated matrices are also known. Hackbusch and Nowak [3] developed a panel clustering method, which also enables approximate matrix vector multiplications with decreased amount of arithmetical work. A class of hierarchical matrices was introduced by Hackbusch [4] with the aim of reducing the complexity of matrix–vector multiplications. Bebendorf and Rjasanow [5] developed an algebraic approach for solving integral equations using collocation methods with almost linear complexity. Ravnik *et al.* [6] developed a wavelet compression method and used it for compression of single domain BEM in 2D.

The algorithm proposed in this paper tackles the domain integral problem using two techniques: a fast multipole method (FMM)-based single domain BEM is employed for the solution of the kinematics equation and subdomain BEM is used for diffusion-advection-type equations.

In the subdomain BEM, integral equations are written for each subdomain (mesh element) separately. We use continuous quadratic boundary elements for the discretization of function and discontinuous linear boundary element for the discretization of flux. By the use of discontinuous discretization of flux, all flux nodes are within boundary elements where the normal and the flux are unambiguously defined. The corners and edges, where the normal is not well defined, are avoided. The singularities of corners and edges were dealt with special singular shape functions by Ong and Lim [7] and by the use of additional nodes by Gao and Davies [8]. By the use of a collocation scheme, a single linear equation is written for every function and flux node in every boundary element. By using compatibility conditions between subdomains, we obtain an over-determined system of linear equations, which may be solved in a least-squares manner. The governing matrices are sparse and have similar storage requirements as the finite element method. Subdomain BEM was applied on the Laplace equation by Ramšak and Škerget [9] and on the velocity–vorticity formulation of Navier–Stokes equations by Ravnik *et al.* [10, 11].

The second part of the algorithm uses (FMM)-based single domain BEM. FMM is used to provide a sparse approximation of the fully populated BEM domain matrices. The storage requirements of the sparse approximations scale linearly with the number of nodes in the domain, which is a major improvement over the quadratic complexity of the full BEM matrices. The technique eliminates the storage and CPU time problems associated with the application of BEM on nonhomogenous partial differential equations.

DOI: 10.1002/nme

The origins of the FMM can be found in a fast algorithm for particle simulations developed by Greengard and Rokhlin [12]. The algorithm decreases the amount of work required to evaluate mutual interaction of particles by reducing the complexity of the problem from quadratic to linear. Ever since, the method was used by many authors for a wide variety of problems using different expansion strategies. Recently, Bui *et al.* [13] combined FMM with the Fourier transform to study multiple bubbles dynamics. Gumerov and Duraiswani [14] applied the FMM for the biharmonic equation in 3D. The boundary integral Laplace equation was accelerated with FMM by Popov *et al.* [15]. In contrast to the contribution of this paper, where the subject of study is the application of FMM to obtain a sparse approximation of the domain matrix, the majority of work done by other authors dealt with coupling BEM with FMM for the boundary matrices. Only recently, Ravnik *et al.* [16] compared wavelet and fast multipole data sparse approximations for boundary-domain integral equations of Poisson type.

The rest of the paper is organized as follows: the second section gives the governing equations, the third section presents the numerical solution, the fourth section analyses the results of simulations and the last section summarizes the main findings.

2. GOVERNING EQUATIONS

Incompressible fluid flow and heat transfer are governed by laws of conservation of mass, momentum and energy. A system of partial differential equations, known as the Navier–Stokes equations, links the velocity, pressure and temperature fields. In this paper, we are considering an alternative approach: the velocity–vorticity formulation of Navier–Stokes equations. This system of partial differential equations consists of the kinematics equation, the vorticity transport equation and the energy equation linking the velocity, vorticity and temperature fields.

The kinematics equation is a vector elliptic partial differential equation of Poisson type and links the velocity and vorticity fields for every point in space and time. It is equivalent to the Biot–Savart law, which connects the electric current and magnetic field density. For an incompressible fluid the kinematics equation can be stated as

$$\nabla^2 \mathbf{v} + \nabla \times \boldsymbol{\omega} = 0 \tag{1}$$

where we must bear in mind that both velocity, $\mathbf{v}$, and vorticity, $\boldsymbol{\omega} = \nabla \times \mathbf{v}$, fields are divergence free. We will nondimensionalize all quantities by

$$\mathbf{v} \to \frac{\mathbf{v}}{v_0}, \quad \mathbf{g} \to \frac{\mathbf{g}}{g_0}, \quad \mathbf{r} \to \frac{\mathbf{r}}{\mathscr{L}}, \quad \boldsymbol{\omega} \to \frac{\boldsymbol{\omega}\mathscr{L}}{v_0}, \quad t \to \frac{v_0 t}{\mathscr{L}}, \quad T \to \frac{T - T_0}{\Delta T} \tag{2}$$

where $\mathscr{L}$ is a characteristic length scale, t is time, T is temperature, $\mathbf{r}$ is a position vector, $g_0 = 9.81\,\mathrm{m/s^2}$ and ΔT is the characteristic temperature difference.

The kinetic aspect of fluid movement is governed by the vorticity transport equation. Buoyancy is modelled with the Boussinesq approximation. Density variations with temperature $\rho(T) = \rho_0[1 - \beta_T(T - T_0)]$ are considered only in the buoyancy term and defined by the thermal volume expansion coefficient β_T and the temperature difference. Using these assumptions we write the vorticity

DOI: 10.1002/nme

transport equation as

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\mathbf{v}\cdot\nabla)\boldsymbol{\omega} = (\boldsymbol{\omega}\cdot\nabla)\mathbf{v} + \frac{1}{Re}\nabla^2\boldsymbol{\omega} - \frac{Ra}{Pr\,Re^2}\nabla\times T\mathbf{g} \tag{3}$$

with Reynolds, Prandtl and Rayleigh numbers defined as

$$Re = \frac{v_0 \mathscr{L}}{\nu_0}, \quad Pr = \frac{\nu_0}{\alpha_0}, \quad Ra = \frac{g_0 \beta_T \Delta T\, \mathscr{L}^3}{\nu_0 \alpha_0} \tag{4}$$

where ν_0 is viscosity and α_0 is diffusivity. Equation (3) equates the advective vorticity transport on the left-hand side with the vortex twisting and stretching term, the diffusion term and the buoyancy term on the right-hand side. We further assume that no internal energy sources are present in the fluid. We will not deal with high velocity flow of highly viscous fluid; hence, we will neglect irreversible viscous dissipation. With this, the internal energy conservation law, written with temperature as the unknown variable, reads as

$$\frac{\partial T}{\partial t} + (\mathbf{v}\cdot\nabla)T = \frac{1}{Re\,Pr}\nabla^2 T \tag{5}$$

The partial differential equations (1), (3) and (5) form a non-linear system for the unknown velocity, vorticity and temperature fields. The problem is in a given domain uniquely defined by specifying the Reynolds, Prandtl and Rayleigh numbers and the appropriate boundary conditions.

3. NUMERICAL METHOD

We will apply a combination of subdomain BEM and single domain BEM for the solution of the governing equations. The Dirichlet and/or Neumann boundary conditions for velocity and temperature are given. They are used to obtain solutions of the kinematics equation (1) for domain velocity values and energy equation (5) for domain temperature values. The boundary conditions for vorticity, which are needed to solve the vorticity transport equation (3), are unknown. We will use the fast multipole single domain BEM on the kinematics equation to obtain the remaining unknown boundary vorticity values. The outline of the algorithm is as follows

- initialization, calculate integrals and fast multipole expansion terms
- begin non-linear loop
  - calculate boundary vorticity values by solving the kinematics equation (1) by fast multipole single domain BEM (Section 3.1)
  - calculate domain velocity values by solving the kinematics equation (1) by subdomain BEM (Section 3.4)
  - solve the energy equation (5) using the new velocity field for domain temperature values by subdomain BEM (Section 3.3)
  - solve vorticity transport equation (3) by subdomain BEM for domain vorticity values using the boundary values from the solution of the kinematics equation and new velocity and temperature fields (Section 3.2)
  - check convergence—repeat steps in the non-linear loop until convergence of all field functions is achieved

DOI: 10.1002/nme

- end non-linear loop
- output results

The steps of the algorithm are explained in detail in the following subsections.

3.1. Fast multipole single domain BEM

Consider an arbitrary domain Ω with a position vector $\mathbf{r} \in \mathbb{R}^3$ and its boundary $\Gamma = \partial\Omega$. The integral form of the kinematics equation without derivatives of the velocity and vorticity fields takes the following form (for derivation, see Ravnik *et al.* [17] Equations (19)–(24)):

$$c(\boldsymbol{\xi})\mathbf{v}(\boldsymbol{\xi}) + \int_\Gamma \mathbf{v}\ \nabla u^\star \cdot \mathbf{n}\, \mathrm{d}\Gamma = \int_\Gamma \mathbf{v} \times (\mathbf{n} \times \nabla) u^\star\, \mathrm{d}\Gamma + \int_\Omega (\boldsymbol{\omega} \times \nabla u^\star)\, \mathrm{d}\Omega \tag{6}$$

where $\boldsymbol{\xi}$ is the collocation point, $u^\star = 1/4\pi|\mathbf{r} - \boldsymbol{\xi}|$ is the fundamental solution of the Laplace equation and $\mathbf{n}$ is the unit normal. In order to use the kinematics equation to obtain boundary vorticity values, we must rewrite Equation (6) in a tangential form by multiplying it with a normal in the source point $\mathbf{n}(\boldsymbol{\xi})$:

$$c(\boldsymbol{\xi})\mathbf{n}(\boldsymbol{\xi}) \times \mathbf{v}(\boldsymbol{\xi}) + \mathbf{n}(\boldsymbol{\xi}) \times \int_\Gamma \mathbf{v} \nabla u^\star \cdot \mathbf{n}\, \mathrm{d}\Gamma = \mathbf{n}(\boldsymbol{\xi}) \times \int_\Gamma \mathbf{v} \times (\mathbf{n} \times \nabla) u^\star\, \mathrm{d}\Gamma + \mathbf{n}(\boldsymbol{\xi}) \times \int_\Omega (\boldsymbol{\omega} \times \nabla u^\star)\, \mathrm{d}\Omega \tag{7}$$

This approach has been proposed by Škerget and used in 3D by Žunič *et al.* [18] and Ravnik *et al.* [10]. In order to write a linear system of equations for the unknown boundary vorticity values, we set the source point into every boundary node of the whole computational domain.

In this paper we use meshes made up of hexahedral domain elements and parallelepipedial boundary elements. Shape functions are used to describe function variation in elements. Quadratic interpolation is achieved by using 27 nodes per hexahedra and domain shape functions Φ_i. Boundary shape functions φ_i having nine nodes are used within each boundary element.

To be able to write a discrete form of the kinematics equation, the following integrals must be calculated:

$$[H] = \int_\Gamma \varphi_i \nabla u^\star \cdot \mathbf{n}\, \mathrm{d}\Gamma, \quad [\mathbf{H}^t] = \int_\Gamma \varphi_i (\mathbf{n} \times \nabla) u^\star\, \mathrm{d}\Gamma \tag{8}$$

$$[\mathbf{D}] = \int_\Omega \Phi_i \nabla u^\star\, \mathrm{d}\Omega \tag{9}$$

The square brackets denote integral matrices. Each collocation point location yields one row in these matrices. The collocation point is placed into every boundary node, making the number of rows of matrices (8) and (9) equal to the number of boundary nodes. Since the integrals (8) are boundary integrals the number of columns of matrices $[H]$ and $[\mathbf{H}^t]$ are also equal to the number of boundary nodes. The domain matrices $[\mathbf{D}]$ have the number of columns equal to the number of all nodes.

A discrete form of Equation (7) is

$$[\mathbf{n}] \times [H]\{\mathbf{v}\} = [\mathbf{n}] \times (\{\mathbf{v}\} \times [\mathbf{H}^t]) + [\mathbf{n}] \times (\{\boldsymbol{\omega}\} \times [\mathbf{D}]) \tag{10}$$

DOI: 10.1002/nme

where the curly brackets denote vectors of nodal values, [**n**] is the matrix of nodal values of $\mathbf{n}(\boldsymbol{\xi})$ and the calculated $c(\boldsymbol{\xi})$ are added to the diagonal terms of the $[H]$ matrix. In order to use Equation (10) to calculate boundary vorticity values, it must be rearranged so that the boundary part of domain matrices $[\mathbf{D}_\Gamma]$ is moved to the left-hand side and all other matrices form the right-hand side:

$$[\mathbf{n}]\times(\{\boldsymbol{\omega}_\Gamma\}\times[\mathbf{D}_\Gamma])=-[\mathbf{n}]\times[H]\{\mathbf{v}\}+[\mathbf{n}]\times(\{\mathbf{v}\}\times[\mathbf{H}^t])+[\mathbf{n}]\times(\{\boldsymbol{\omega}_\Omega\}\times[\mathbf{D}_\Omega]) \tag{11}$$

The domain matrices $[\mathbf{D}_\Omega]$ on the right-hand side take up most storage space and their multiplication with vectors take up most CPU time. The number of elements in $[\mathbf{D}_\Omega]$ is equal to $n_b\cdot n_d$, where n_b is the number of boundary nodes and n_d is the number of domain nodes. Using the FMM we will set up an approximation of the domain matrices $[\mathbf{D}'_\Omega]\approx[\mathbf{D}_\Omega]$ whose complexity will scale as $\mathcal{O}(n_d)$.

Let us consider the domain integrals in Equation (9), which must be calculated in order to set up $[\mathbf{D}_\Omega]$. Since for each collocation point $\boldsymbol{\xi}$ integrals for all domain cells must be evaluated, we are obviously faced by a problem of quadratic complexity. The FMM, described below, is used to reduce this complexity.

3.1.1. Series expansion. The FMM is based on the fact that it is possible to separate the variables (i.e. the collocation point $\boldsymbol{\xi}$ and the domain integration point $\mathbf{r}$) of the integral kernel of Equation (9) by series expansion. The gradient of the Laplace fundamental solution of Equation (9) is expanded into a spherical harmonics series in the following manner:

$$\begin{aligned}\nabla u^\star &= \nabla\frac{1}{4\pi|\mathbf{r}-\boldsymbol{\xi}|}=\sum_{l=0}^{\infty}\sum_{m=-l}^{l}\frac{(-1)^m}{2l+1}\frac{1}{\xi^{l+1}}Y_l^{-m}(\theta_\xi,\varphi_\xi)\nabla[r^lY_l^m(\theta_r,\varphi_r)]\\ &=\sum_{l=0}^{\infty}\sum_{m=-l}^{l}\frac{(-1)^m}{2l+1}\frac{1}{\xi^{l+1}}Y_l^{-m}(\theta_\xi,\varphi_\xi)\{lY_l^m(\theta_r,\varphi_r)r^{l-2}\mathbf{r}+r^l\nabla Y_l^m(\theta_r,\varphi_r)\}\end{aligned} \tag{12}$$

where Y_l^m are spherical harmonics in polar coordinate system; $\mathbf{r}=(r,\varphi_r,\theta_r)$ and $\boldsymbol{\xi}=(\xi,\varphi_\xi,\theta_\xi)$. The gradient of spherical harmonics is expressed using associated Legendre polynomials P_l^m as

$$\begin{aligned}\nabla Y_l^m(\theta,\varphi)&=\sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}}\nabla\{P_l^m(\cos\theta)\mathrm{e}^{\mathrm{i}m\varphi}\}\\ &=\sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}}\nabla\left\{P_l^m(\cos\theta)\frac{\partial\mathrm{e}^{\mathrm{i}m\varphi}}{\partial\varphi}\nabla\varphi-\sin(\theta)\frac{\partial P_l^m(\cos\theta)}{\partial\cos\theta}\nabla\theta\right\}\end{aligned} \tag{13}$$

The associated Legendre polynomials are evaluated using recurrence relations as described in Press *et al.* [19]. The derivatives of associated Legendre polynomials are obtained using the following recurrence relation:

$$\frac{\partial P_l^m(x)}{\partial x}=\frac{lxP_l^m(x)-(l+m)P_{l-1}^m(x)}{x^2-1} \tag{14}$$

DOI: 10.1002/nme

The derivatives of the polar angles written in Cartesian coordinate system are

$$\begin{aligned}\nabla\theta &= \frac{\sqrt{x^2+y^2}}{x^2+y^2+z^2}\left(\frac{zx}{x^2+y^2}, \frac{zy}{x^2+y^2}, -1\right)\\ \nabla\varphi &= \frac{1}{x^2+y^2}(-y, x, 0)\end{aligned} \tag{15}$$

In order for the series (12) to converge we must have $r/\xi<1$. This condition is not satisfied for all $\boldsymbol{\xi}$ and $\mathbf{r}$ combinations in an arbitrary domain. However, since the integral kernel depends only on the distance between the collocation and domain points, we may swap $\boldsymbol{\xi}$ and $\mathbf{r}$ in order to meet the convergence criteria. Furthermore, it is also possible to move the origin of the coordinate system so that the series convergence is improved.

Using the above expansion, the domain integrals of Equation (9) may now be written with separate variables as

$$[\mathbf{D}_\Omega] \approx [\mathbf{D}'_\Omega] = \sum_{l=0}^{L} \sum_{m=-l}^{l} F_l^m(\boldsymbol{\xi}) \int_\Omega G_l^m(\mathbf{r})\, \mathrm{d}\Omega \tag{16}$$

where F and G represent the above-derived relationships. We are able to approximately calculate each entry in the domain matrices with the above sum. The number of expansion terms $n_{\exp} = (L+1)^2$ in the series controls the accuracy of the approximation.

Using the series (16) instead of the direct evaluation of the integral kernel does not by itself bring a reduction in memory. Only when (16) is used on a cluster of collocation points and domain cells it is possible to form a data sparse approximation of a part of the domain matrix. The clusters are formed and organized in a hierarchical tree-like structure, which is described below.

3.1.2. Cluster trees. Let us consider a cluster of n_r nearby collocation points and a cluster of n_c nearby domain cells, as illustrated in Figure 1. These correspond to an $n_r \times n_c$ matrix block, which is a part of the domain matrix. Since the variables in Equation (16) are separated, it is possible to evaluate two lower-order matrix blocks ($n_r \times n_{\exp}$) and ($n_{\exp} \times n_c$) instead of the full matrix block ($n_r \times n_c$). In the first lower-order matrix block expansion terms F are evaluated for all collocation points. In the second one integrals of expansion terms G are evaluated for all domain cells. Multiplication of the two lower-order matrix blocks gives the full $n_r \times n_c$ matrix block up to an expansion error, which is defined by the number of terms in the expansion. But this is never done; namely we store the two lower-order matrices instead of the full matrix. This technique saves memory if the amount of data that must be stored in the two lower-order matrices is smaller than the amount of data in the full matrix block, i.e.

$$2(n_r n_{\exp} + n_c n_{\exp}) < n_r n_c \tag{17}$$

the factor 2 on the left-hand side is due to the fact that spherical harmonics are complex and must be stored as such, while real values are stored in the full matrix. As long as the collocation node cluster and the domain cells cluster are far apart from each other the integral kernels are slowly varying functions; thus, we can expect a low number of expansion terms to yield a suitable approximation. When the clusters are nearby, they should be smaller and a larger number of expansion terms must be used. When the clusters coincide, i.e. the collocation nodes are a part

DOI: 10.1002/nme

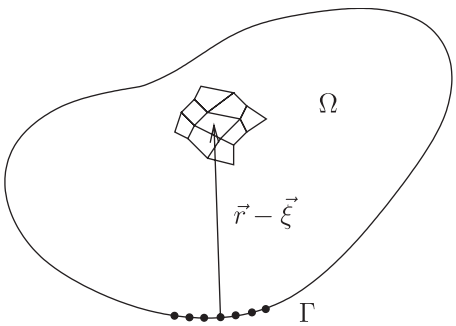

Figure 1. A problem domain shown with a cluster of collocation points $\boldsymbol{\xi}$ and a cluster of domain cells.

of the integration cells, the kernels are singular. Such cluster pairs are called inadmissible and the corresponding matrix block is evaluated in full, not approximated with two lower-order matrices.

In order to be able to build a sparse approximation of the whole domain matrix, we must divide the collocation points and domain cells into clusters. We constructed a tree of collocation point clusters and a tree of clusters of domain cells. The trees were constructed in a recursive hierarchical manner. The problem domain was enclosed by a parallelepiped. All collocation points and all of the domain cells are within this root parallelepiped. They make up root clusters of both trees. The parallelepiped is cut in half by a plane, breaking the root clusters into two. The cutting process is repeated recursively, thus, the clusters on each level have less and less collocation points and domain cells. Each branch in the tree of clusters has two child branches corresponding to the cluster's domain being cut in half. The cutting planes are parallel to the coordinate system axes, a sequence of $x-y$, $x-z$ and $y-z$ is used. Thus, three cuts are needed to cut a cube into eight equal parts. The cutting sequence is stopped, when the number of collocation nodes and domain cells in the cluster is so small that condition (17) can no longer be satisfied.

With both cluster trees in place, the next step is to pair them; thus, a tree of pairs of clusters can be formed. Each branch of the collocation tree is paired with each branch of the domain cells tree on the same level and with each branch of the domain cells tree on the next level, thus forming branches on the tree of pairs of clusters. For each pair a decision is taken based on the admissibility criterion whether a sparse approximation for this cluster pair is possible or not. If the pair is admissible, the branch on the tree becomes an admissible leaf, where the two low-order matrices will be calculated. If the admissibility criterion is not reached until the last level of the tree, such cluster pairs are inadmissible and will be calculated in full and not with the sparse approximation.

3.1.3. Admissibility criterion. The admissibility criterion is devised as follows. Let us consider one branch of the tree of pairs of clusters, which has a cluster of collocation points and a cluster of domain cells. First, we try to find an origin of the coordinate system in nodes within the domain cells of the cluster. We choose such origin that the ratio r/ξ is minimal for all pairs of collocation nodes and domain cells so that the series will converge as fast as possible. If the minimal ratio is greater than one, series expansion for this pair of clusters is not possible. Thus this pair is not admissible. Second, if the r/ξ ratio is less than one, we calculate the number of expansion terms needed to have the accuracy of calculation of the integral kernel less than the user's prescribed criteria ε. If the number of expansion terms is low enough, so that condition (17) is fulfilled,

DOI: 10.1002/nme

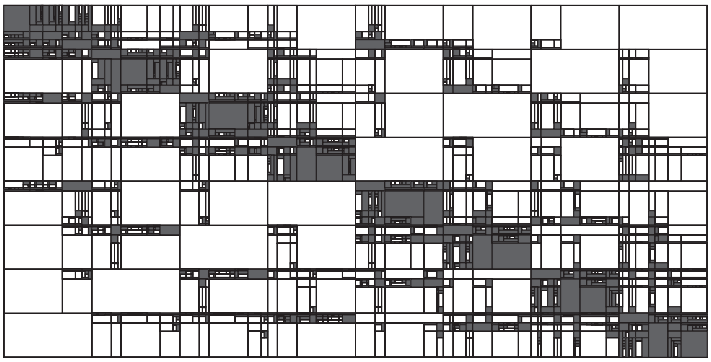

Figure 2. Matrix structure of a cubic mesh (12^3 cells, 25^3 nodes). Filled areas show inadmissible matrix blocks; white areas are admissible matrix blocks obtained using an admissibility criterion of $\varepsilon = 10^{-4}$.

this cluster pair is admissible. At this point the tree of pairs of clusters gets a leaf—no further branching is necessary.

To illustrate the algorithm, a cubic domain is considered meshed by 12^3 domain cells having in total 25^3 nodes. Admissible and inadmissible blocks are shown in Figure 2.

3.1.4. Implementation. A set of routines was written to form a hierarchical tree structure with evaluated matrices on each of the admissible or inadmissible leaves. They are capable of constructing an approximation of the domain matrices $[\mathbf{D}'_\Omega]$, which are used to evaluate the right-hand side of the discrete kinematics system of equations (11). The advantages of using the approximation $[\mathbf{D}'_\Omega]$ instead of the fully populated $[\mathbf{D}_\Omega]$ are summarized in the following points.

- Since we have taken careful care of the fact that the amount of data required to store matrices in all admissible leaves is smaller than data storage of their fully populated counterparts, we know that the memory required to store the FMM sparse approximation of the domain matrices $[\mathbf{D}'_\Omega]$ will be less than the memory required to store $[\mathbf{D}_\Omega]$.
- Evaluation of the two lower-order matrices for each admissible leaf is computationally less expensive than the computation of their full matrix block counterpart. First, there are less matrix elements to evaluate; second, one of the low-order matrices holds values only (see Equation (16)) and not integrals. The elements of the second low-order matrix are integrals of slowly varying functions; thus, less effort is needed to evaluate them.
- We are approximating three domain matrices, each holding integrals of one component of the gradient of the fundamental solution. Since one of the low-order matrices in admissible leaves holds values independent of $\mathbf{r}$, they are also independent of the gradient direction. Thus, only one set of these matrices, which is common for all three directions, needs to be stored. This saves additional data storage space.

3.2. Subdomain BEM solution of the vorticity transport equation

The subdomain BEM is used to solve the vorticity transport equation for domain vorticity values. Boundary vorticity values obtained by the FMM single domain BEM solution of the kinematics equation are used as Dirichlet boundary conditions. The flows considered in this paper are steady;

DOI: 10.1002/nme

thus, we may write $\partial\boldsymbol{\omega}/\partial t = 0$. The boundary-domain integral form of the steady vorticity transport equation (3) is (Wrobel [1]):

$$c(\boldsymbol{\xi})\boldsymbol{\omega}(\boldsymbol{\xi}) + \int_\Gamma \boldsymbol{\omega}\nabla u^\star \cdot \mathbf{n}\,\mathrm{d}\Gamma = \int_\Gamma u^\star \mathbf{q}\,\mathrm{d}\Gamma + Re\int_\Omega u^\star\{(\mathbf{v}\cdot\nabla)\boldsymbol{\omega} - (\boldsymbol{\omega}\cdot\nabla)\mathbf{v}\}\,\mathrm{d}\Omega + \frac{Ra}{PrRe}\int_\Omega u^\star \nabla\times T\mathbf{g}\,\mathrm{d}\Omega \tag{18}$$

where $u^\star$ is the fundamental solution for the diffusion operator. Equation (18) is rewritten in a way that omits the derivatives of field functions (full derivation is given in Ravnik *et al.* [11]). The integral vorticity transport equation in Cartezian tensor notation is

$$\begin{aligned} c(\boldsymbol{\xi})\omega_j(\boldsymbol{\xi}) &+ \int_\Gamma \omega_j \nabla u^* \cdot \mathbf{n}\,\mathrm{d}\Gamma \\ &= \int_\Gamma u^* q_j\,\mathrm{d}\Gamma + Re\int_\Gamma \mathbf{n}\cdot\{u^*(\mathbf{v}\omega_j - \boldsymbol{\omega}v_j)\}\,\mathrm{d}\Gamma - Re\int_\Omega (\mathbf{v}\omega_j - \boldsymbol{\omega}v_j)\cdot\nabla u^*\,\mathrm{d}\Omega \\ &\quad - \frac{Ra}{PrRe}\int_\Gamma (u^* T\mathbf{g}\times\mathbf{n})_j\,\mathrm{d}\Gamma - \frac{Ra}{PrRe}\int_\Omega (T\nabla\times u^*\mathbf{g})_j\,\mathrm{d}\Omega \end{aligned} \tag{19}$$

where q_j is the jth component of vorticity flux. In the subdomain BEM method we make a mesh of the entire domain Ω and name each mesh element a subdomain. Equation (19) is written for each of the subdomains. In order to obtain a discrete version of (19) we use shape functions to interpolate field functions and flux across the boundary and inside of the subdomain. In this paper we used hexahedral subdomains with 27 nodes, which enable continuous quadratic interpolation of field functions. The boundary of each hexahedron consists of six boundary elements. On each boundary element we interpolate the flux using discontinuous linear interpolation scheme with four nodes. By using discontinuous interpolation we avoid flux definition problems in corners and edges.

The collocation point is placed into all function and flux nodes of each subdomain. Since neighbouring subdomains share nodes, compatibility conditions between subdomains are used and the resulting systems of linear equations are over-determined. After taking into account the boundary conditions, we solve them using a least-squares solver (Paige and Saunders [20]). All integrals depend only on the shape of subdomains and as such may be calculated only once, prior to the start of the non-linear iterative process.

3.3. Subdomain BEM solution of the energy equation

A subdomain BEM is used to solve the energy equation for domain values of temperature. The energy equation (5) is a diffusion convection partial differential equation exactly like the vorticity transport equation without the vortex twisting and stretching and buoyancy terms. The boundary conditions are Dirichlet or Neumann type and are prescribed by the user. The solution of Equation (5) is thus obtained in the same manner as the solution of the vorticity transport equation.

DOI: 10.1002/nme

The boundary-integral equation is

$$c(\boldsymbol{\xi})T(\boldsymbol{\xi})+\int_{\Gamma} T\nabla u^{\star}\cdot\mathbf{n}\,\mathrm{d}\Gamma=\int_{\Gamma} u^{\star}\mathbf{q}_T\,\mathrm{d}\Gamma+RePr\int_{\Gamma}\mathbf{n}\cdot u^{\star}\mathbf{v}T\,\mathrm{d}\Gamma$$
$$-RePr\int_{\Omega} T\mathbf{v}\cdot\nabla u^{\star}\,\mathrm{d}\Omega \tag{20}$$

where $\mathbf{q}_T$ is the temperature flux. After discretization the integrals are stored in sparse matrices and the system of linear equations is solved in a least-squares manner [20].

3.4. Subdomain BEM solution of the kinematics equation

The kinematics equation is solved for domain velocity values by subdomain BEM. The integral form (6) is used. Dirichlet-type boundary conditions are prescribed on the boundary of the domain. A discrete system of linear equations is written for each subdomain. After taking into account the compatibility conditions, an over-determined system is obtained for each velocity component. They are solved in a least-squares manner [20] for the unknown velocity field in the domain.

4. NUMERICAL EXAMPLES

4.1. Lid-driven cavity

Flow in a 3D lid-driven cavity is one of the standard benchmark test cases used in the development of flow solvers. The domain and the boundary conditions are unambiguously defined and do not change with the Reynolds number. The flow exhibits a wide variety of phenomena, such as eddies, complex 3D patterns and instabilities (Shankar and Deshpande [21]). The research of lid-driven cavity flow started with the observations of Koseff and Street [22], who were able to observe the inherent 3D nature of flow phenomena in the cavity.

The simulation was performed on a unit cube $(0,0,0)\times(1,1,1)$. We named the walls in the following manner: left wall $x=0$, right $x=1$, top $z=1$, bottom $z=0$, front $y=0$ and back $y=1$. No-slip velocity boundary conditions are employed on all walls except for the top wall, where a constant velocity in x direction is prescribed $\mathbf{v}=(1,0,0)$. Dirichlet-type boundary conditions are used for the vorticity transport equation. Vorticity on the boundary is obtained by the solution of the kinematics equation for all directions and walls, except for $\omega_x=0$ on the left and right walls, $\omega_y=0$ on the front and back walls and $\omega_z=0$ on the top and bottom walls. The boundary conditions are sketched in Figure 3.

Simulations were run on a mesh of 12^3 elements with 15 625 nodes. The elements were concentrated towards the eight corners. The mesh is coarse enough that it enables simulations with and without approximation of the domain matrices.

The Reynolds number for this test case is defined with the length of cavity's edge and the top wall velocity. We ran simulations at $Re=100$, $Re=400$ and $Re=1000$.

We set up the cluster trees with an admissibility condition of $\varepsilon=10^{-4}$ and ran simulations with 4, 9, 16 and 25 expansion terms in FMM series. A simulation without using the FMM approximation of domain matrices was run as well. We compared velocity profiles within the cavity in the $y=0.5$ plane with the results of Yang *et al.* [23] on Figure 4. We observe good agreement between benchmark results and present results obtained without FMM approximation

DOI: 10.1002/nme

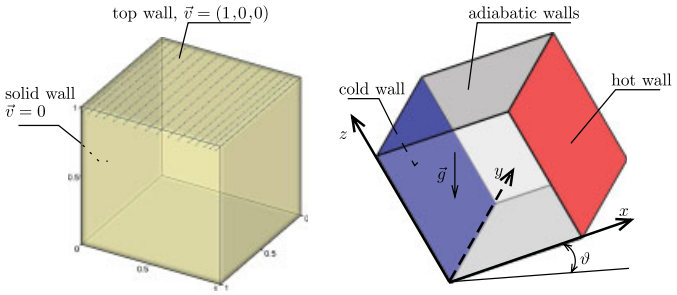

Figure 3. Geometry and boundary conditions of the lid-driven cavity (left) and natural convection (right) test cases.

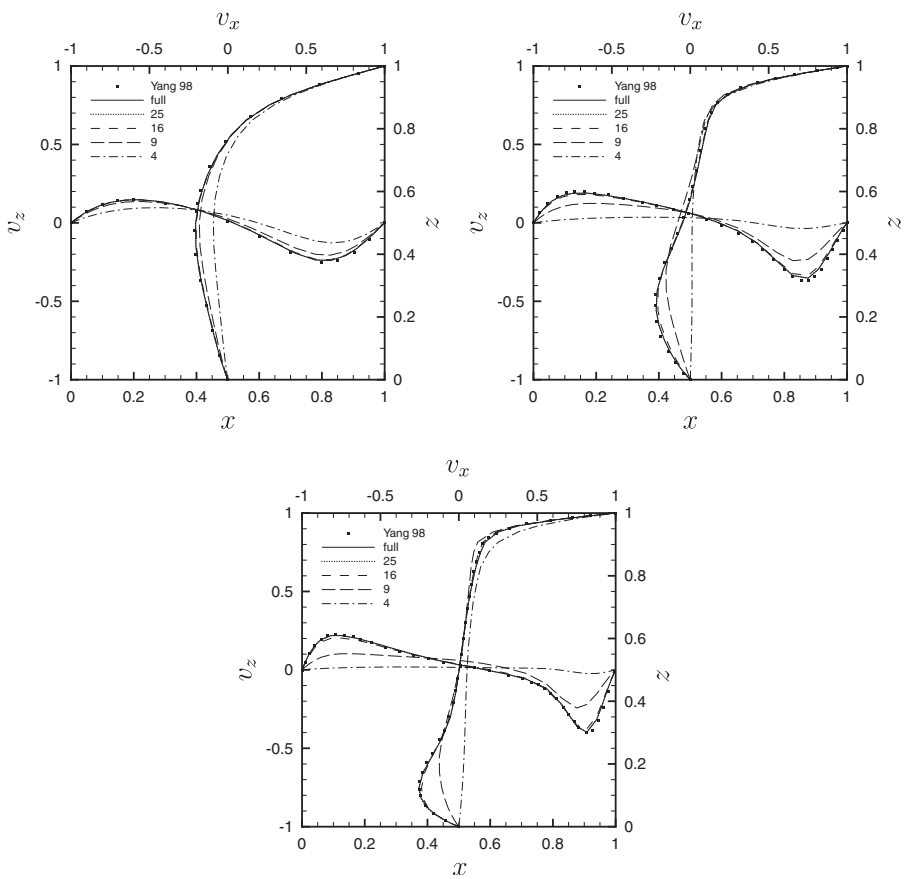

Figure 4. Velocity profiles through the $y=0.5$ plane in the lid-driven cavity test case. Solution obtained using the full domain matrices (full) are compared with solutions obtained with FMM approximations with different numbers of expansion terms and the benchmark results of Yang *et al.* [23]. The admissibility criterion was $\varepsilon=10^{-4}$. $Re=100$ (top left), $Re=400$ (top right) and $Re=1000$ (bottom).

DOI: 10.1002/nme

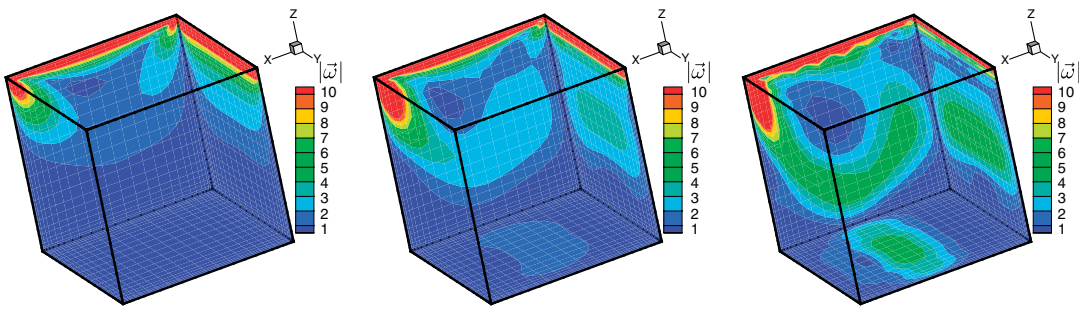

Figure 5. Vorticity magnitude $|\boldsymbol{\omega}|$ contours along the walls of the cavity. Left $Re=100$, middle $Re=400$ and right $Re=1000$.

for all Reynolds number values considered. When FMM approximation is used, the accuracy of results increases with the increasing number of expansion terms. Increasing non-linearity of the problem severely affects the accuracy of the simulations. We observe that good agreement with the benchmark is obtained with less expansion terms in the case of low Reynolds number and more terms are needed for the same accuracy at higher Reynolds number values. Thus, the non-linearity of flow phenomena requires careful setting of the number of expansion terms.

The moving lid induces a primary vortex inside of the cavity. The size of the vortex increases with Reynolds number. Secondary vortices appear in the corners of the cavity, their position and strength changing with Reynolds number. Figure 5 shows the development of vorticity on the walls of the cavity with increasing Reynolds number. The vorticity is generated at the wall and transported into the flow by diffusion and advection. The highest vorticity values are found along the moving lid. When the Reynolds number is increased, the high vorticity area spreads along the centres of other walls as well.

4.2. Natural convection in an inclined enclosure

Consider a cubic enclosure filled with air ($Pr=0.71$) subjected to a temperature difference on two opposite walls, while the rest of the walls are adiabatic. The fluid next to the hot wall will be heated and due to its decrease in density buoyancy will carry it upwards. An inverse phenomenon will occur along the cold wall; the fluid there will be colder and denser and thus it will travel downwards. This phenomenon is natural convection and occurs frequently in nature and is present in many industrial applications. The fluid movement is steady up to a critical Rayleigh number, above which vortices are formed along the hot and the cold walls and transported through the flow field by convection and diffusion. Further increase in temperature difference leads through oscillatory behaviour to turbulent natural convection [17]. The phenomenon is defined by the type of fluid, temperature difference and the layout of the enclosure.

The layout of the problem is shown in Figure 3. The temperature $T=-0.5$ will be prescribed at $x=0$ and $T=0.5$ at $x=1$. The other four walls are adiabatic, i.e. there is no temperature flux through them. Since we consider a closed enclosure, no-slip velocity boundary conditions $\mathbf{v}=0$ are applied on all walls.

We set $RePr=1$ and simulated natural convection for $Ra=10^3\ldots10^6$ and for inclination angles $\vartheta=0°$, $\vartheta=30°$ and $\vartheta=60°$. Cluster trees were formed using the admissibility criteria $\varepsilon=10^{-4}$ and $\varepsilon=10^{-5}$ for a mesh with 12^3 elements and 15 625 nodes.

DOI: 10.1002/nme

The heat flux through the side walls is used to compare present results with the benchmark. The flux is presented using the Nusselt number value. The Nusselt number value, defined in our geometry by

$$Nu = \int_0^1 \int_0^1 \left(\frac{\partial T}{\partial x} \right)_{x=0} \mathrm{d}z\, \mathrm{d}y \tag{21}$$

is a nondimensional measure of wall heat flux. Additionally, we define the data ratio, $\mathscr{D}$, as the ratio of the amount of data required to store the FMM approximations of domain matrices to the amount of data required to store the full domain matrices. Naturally, the data ratio increases with the increasing number of FMM expansion terms.

Table I presents Nu values and numbers of non-linear iterations for $\varepsilon = 10^{-4}$, $\vartheta = 0°$ case. The results of simulation using FMM approximation are compared with simulations using full matrices and benchmark results of Lo *et al.* [24] and Tric *et al.* [25]. When a large number of expansion terms is used in the FMM approximation of the domain matrices, we obtain almost identical results as in the full domain matrix case. When less expansion terms are used, the error is larger. The error also increases when a problem with greater non-linearity is considered, i.e. the temperature difference between the walls is increased. The influence of different numbers of expansion terms on the number of iterations required to reach convergence is not very large. Especially at low Rayleigh number values the number of iterations remains constant. On the other hand, in the worst case scenario, when the largest Rayleigh number was simulated using only four expansion terms, the code diverged. The conclusion drawn from this analysis is that the number of terms required for a successful simulation depends on the non-linearity of the problem. More terms should be used for strongly non-linear problems. These conclusions are confirmed by examining the results of the inclined enclosure test cases: $\vartheta = 30°$ in Table II and $\vartheta = 60°$ in Table III.

Comparing the Nusselt number value, which is an integral measure, does not reveal all differences in the flow fields. For the case of $\varepsilon = 10^{-4}$, $\vartheta = 0°$ we plotted velocity and temperature profiles through the $y = 0.5$ plane in Figure 6 for $Ra = 10^3$ and $Ra = 10^4$ and in Figure 7 for $Ra = 10^5$ and $Ra = 10^6$. The figures reveal very good agreement between profiles of 9 or more expansion terms

Table I. Natural convection, $\varepsilon = 10^{-4}$, $\vartheta = 0°$. Nusselt number values and number of non-linear iterations are shown for Rayleigh number values $Ra = 10^3, \ldots, 10^6$. Present results with and without FMM approximation are compared with benchmark results of Lo *et al.* [24] and Tric *et al.* [25].

		Nusselt number				Number of iterations			
		Ra				*Ra*			
No. of terms	$\mathscr{D}$	10^3	10^4	10^5	10^6	10^3	10^4	10^5	10^6
4	0.365	1.0734	2.1254	4.8944	/	109	104	827	/
9	0.434	1.0713	2.0606	4.3924	8.9973	108	104	820	1332
16	0.530	1.0710	2.0547	4.3405	8.6926	108	104	819	1250
25	0.654	1.0711	2.0560	4.3455	8.7076	108	104	819	1262
Full		1.0711	2.0564	4.3462	8.7118	108	104	819	1270
[24]		1.0710	2.0537	4.3329	8.6678				
[25]		1.0700	2.0542	4.3370	8.6407				

DOI: 10.1002/nme

Table II. Natural convection, $\varepsilon = 10^{-4}$, $\vartheta = 30°$.

		Nusselt number				Number of iterations			
		Ra				*Ra*			
No. of terms	$\mathscr{D}$	10^3	10^4	10^5	10^6	10^3	10^4	10^5	10^6
4	0.365	1.0448	1.6262	3.2699	/	104	101	791	/
9	0.434	1.0434	1.5928	2.9350	5.4770	103	101	789	1034
16	0.530	1.0431	1.5895	2.9008	5.2215	103	101	790	963
25	0.654	1.0432	1.5902	2.9034	5.2294	103	101	790	962
Full		1.0432	1.5904	2.9042	5.2342	103	101	790	962
[24]		1.0432	1.5894	2.9014	5.2133				

Table III. Natural convection, $\varepsilon = 10^{-4}$, $\vartheta = 60°$.

		Nusselt number				Number of iterations			
		Ra				*Ra*			
No. of terms	$\mathscr{D}$	10^3	10^4	10^5	10^6	10^3	10^4	10^5	10^6
4	0.365	1.0133	1.1611	1.4212	/	100	98	776	/
9	0.434	1.0127	1.1534	1.3676	1.5891	99	98	770	922
16	0.530	1.0126	1.1523	1.3610	1.5547	99	98	770	888
25	0.654	1.0126	1.1524	1.3610	1.5541	99	98	770	888
Full		1.0126	1.1524	1.3612	1.5545	99	98	770	887
[24]		1.0127	1.1524	1.3623	1.5585				

for the case of $Ra \leqslant 10^5$. In the case of $Ra = 10^6$ only profiles of 16 and 25 expansion terms are in good agreement with the full domain matrix solution. As the mesh is more dense at the walls of the enclosure, the profiles are more accurately close to the walls yielding a good Nusselt number value.

In order to examine the influence of the admissibility criterion, we repeated the simulations of the $\vartheta = 0°$ case with cluster trees obtained with $\varepsilon = 10^{-5}$. The number of leaves in the boundary-domain cluster tree was 2745 as compared with 2378 in the case of $\varepsilon = 10^{-4}$. The increase in the number of leaves is due to the fact that smaller clusters are required to achieve the prescribed higher accuracy. The results presented in terms of the Nusselt number value are given in Table IV. We observe that setting a stricter admissibility criterion resulted in converged results in the $Ra = 10^6$ case with four terms in the FMM series. In general a stricter admissibly criterion on a larger number of expansion terms can be used to improve the accuracy of the results. However, comparing the data ratios and accuracy of results in Tables I and IV, we observe that a worse data ratio is achieved for the same accuracy when using $\varepsilon = 10^{-5}$. We recommend increasing the number of terms in the series according to the non-linearity of the problem and leaving the admissibility criterion at $\varepsilon = 10^{-4}$.

Influence of the inclination angle on the flow field is examined in Figure 8. Here we plot temperature contours on the $y = 0.5$ plane for $Ra = 10^6$. We observe that temperature stratification

DOI: 10.1002/nme

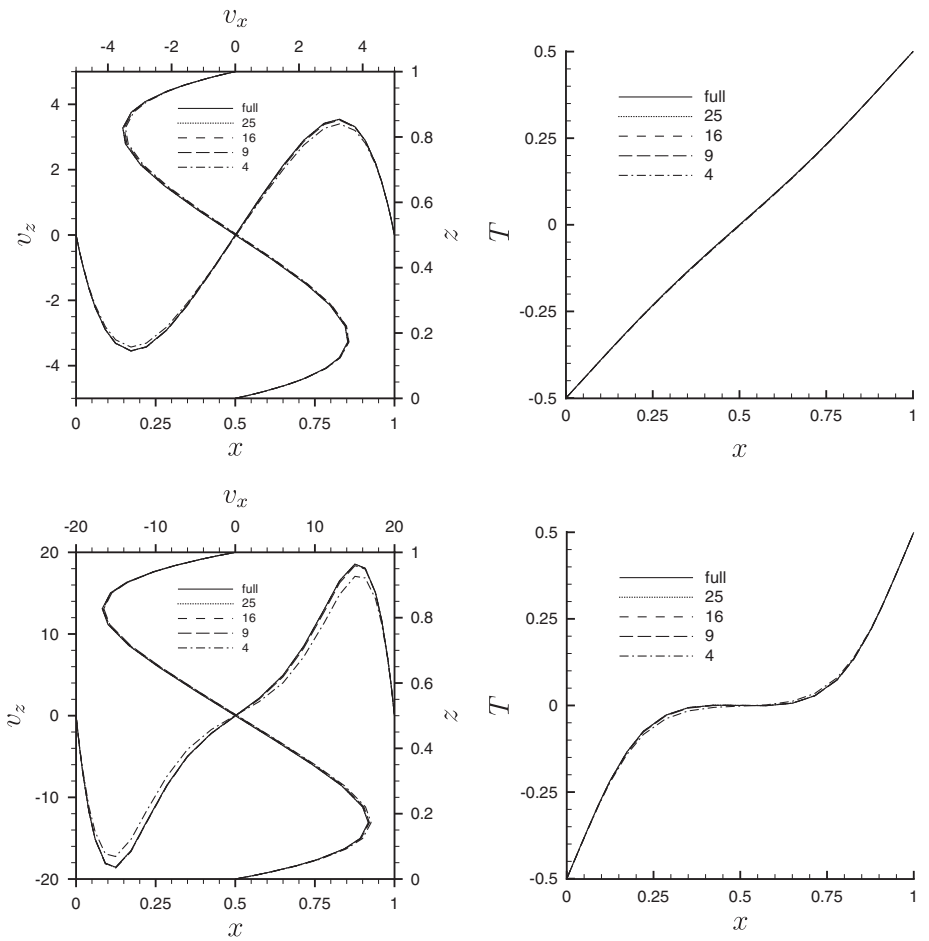

Figure 6. Velocity and temperature profiles through the centre of the enclosure (on the $y=0.5$ plane) in the natural convection test case, $\vartheta=0°$. Solution obtained using the full domain matrices (full) are compared with solutions obtained with FMM approximations with different numbers of expansion terms. The admissibility criterion was $\varepsilon=10^{-4}$. Top row $Ra=10^3$ and bottom row $Ra=10^4$.

occurs in the centre of the enclosure regardless of the inclination angle. The temperature gradient at the wall is highest at $\vartheta=0°$ and lower at higher inclination angles. This is the cause of the highest wall heat flux in the $\vartheta=0°$ case and lower wall heat flux in inclined cases.

5. CONCLUSIONS

We developed a BEM-based numerical algorithm, which was used to simulate incompressible fluid flow and heat transfer by solving the velocity–vorticity formulation of Navier–Stokes equations. Two techniques were combined: the subdomain BEM and the FMM-based single domain BEM.

DOI: 10.1002/nme

Both techniques ensure sparse discrete systems of linear equations making BEM applicable to non-linear and nonhomogenous problems. Spherical harmonics series and a hierarchical cluster tree structure were used for the construction of the FMM domain matrix approximations.

The algorithm was tested by solving two benchmark examples: lid-driven cavity and natural convection in an inclined enclosure. Comparison of results between benchmark solution, solution obtained without FMM approximation and solution with different number of terms in FMM series was preformed. As the number of terms in the FMM is increased, the results converge to the benchmark. We showed that by using nine terms in the expansion results are virtually identical to the results without FMM approximation for $Ra \leqslant 400$ in the lid-driven cavity case and $Ra \leqslant 10^5$ in the natural convection case. For higher Reynolds and Rayleigh numbers ($Re = 1000$, $Ra = 10^6$)

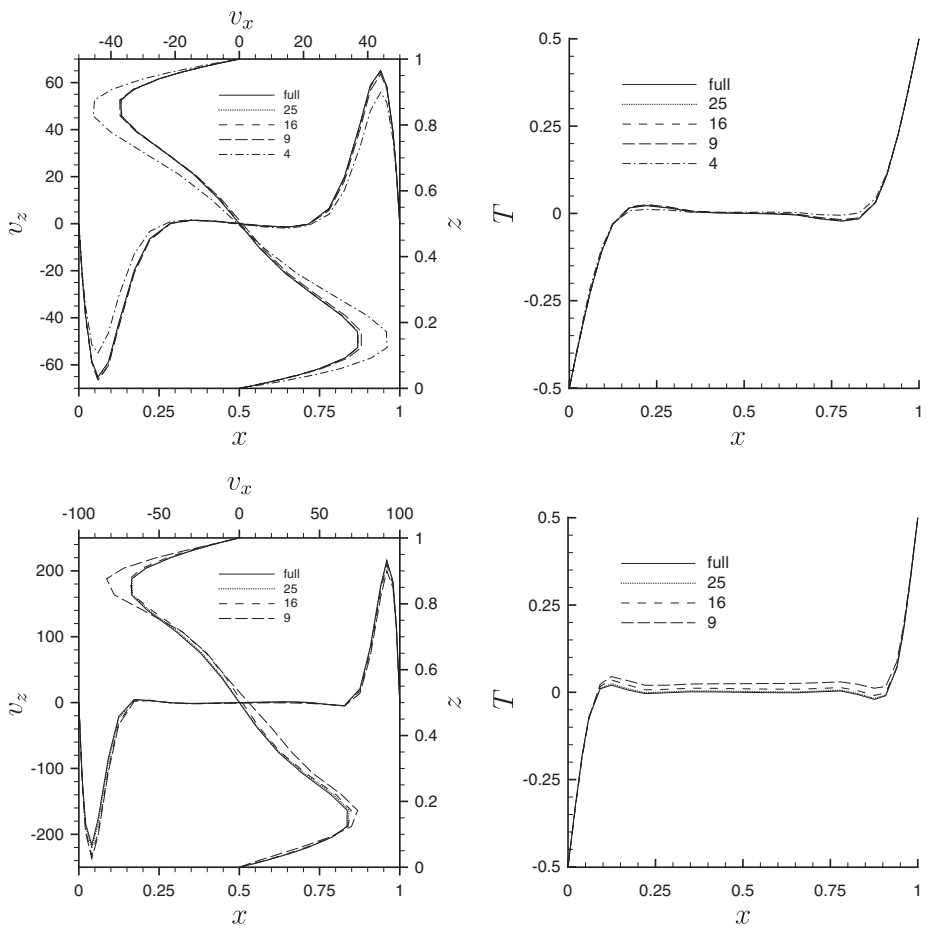

Figure 7. Velocity and temperature profiles through the centre of the enclosure (on the $y = 0.5$ plane) in the natural convection test case, $\vartheta = 0°$. Solution obtained using the full domain matrices (full) are compared with solutions obtained with FMM approximations with different numbers of expansion terms. The admissibility criterion was $\varepsilon = 10^{-4}$. Top row $Ra = 10^5$ and bottom row $Ra = 10^6$.

DOI: 10.1002/nme

Table IV. Natural convection, $\varepsilon = 10^{-5}$, $\vartheta = 0°$.

		Nusselt number				Number of iterations			
		Ra				*Ra*			
No. of terms	$\mathscr{D}$	10^3	10^4	10^5	10^6	10^3	10^4	10^5	10^6
4	0.542	1.0721	2.1003	4.7317	10.5246	108	104	811	3438
9	0.602	1.0710	2.0540	4.3321	8.6512	141	148	808	863
16	0.686	1.0710	2.0550	4.3382	8.6750	108	104	808	769
25	0.794	1.0711	2.0564	4.3473	8.7179	108	104	808	651

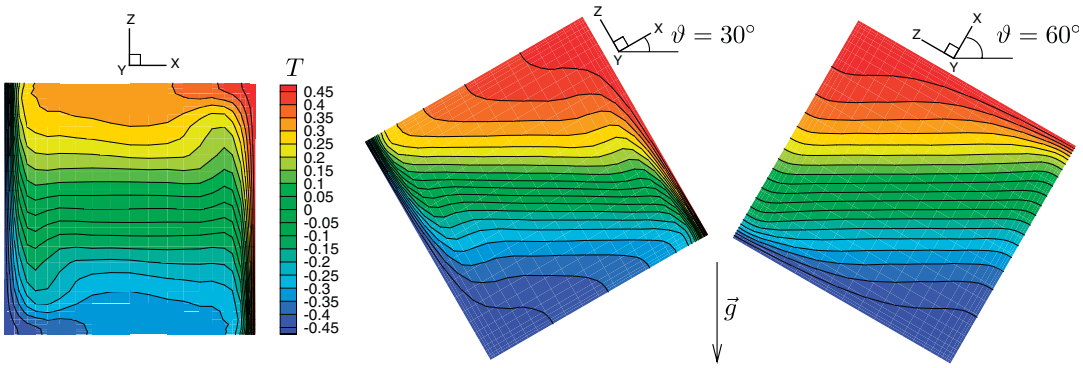

Figure 8. Temperature contours in the centre of the enclosure (on the $y = 0.5$ plane) for $Ra = 10^6$ and three inclination angles.

16 terms must be used. For a general problem, we conclude that the number of terms used in the FMM series must be increased when a problem's non-linearity is increased.

REFERENCES

1. Wrobel LC. *The Boundary Element Method*. Wiley: New York, 2002.
2. Partridge PW, Brebbia CA, Wrobel LC. *The Dual Reciprocity Boundary Element Method*. Computational Mechanics Publications/Elsevier Applied Science: Boston, Southampton, U.K., London, New York, 1992.
3. Hackbusch W, Nowak ZP. On the fast multiplication in the boundary element method by panel clustering. *Numerische Mathematik* 1989; **54**:463–491.
4. Hackbusch W. A sparse matrix arithmetic based on $\mathscr{H}$-matrices. Part I: introduction to $\mathscr{H}$-matrices. *Computing* 1999; **62**:89–108.
5. Bebendorf M, Rjasanow S. Adaptive low rank approximation of collocation matrices. *Computing* 2003; **70**:1–24.
6. Ravnik J, Škerget L, Hriberšek M. The wavelet transform for BEM computational fluid dynamics. *Engineering Analysis with Boundary Elements* 2004; **28**:1303–1314.
7. Ong ET, Lim KM. Three-dimensional singular boundary element method for corner and edge singularities in potential problems. *Engineering Analysis with Boundary Elements* 2005; **29**:175–189.
8. Gao XW, Davies TG. 3D multi-region BEM with corners and edges. *International Journal of Solids and Structures* 2000; **37**:1549–1560.
9. Ramšak M, Škerget L. 3D multidomain BEM for solving the Laplace equation. *Engineering Analysis with Boundary Elements* 2007; **31**:528–538.

DOI: 10.1002/nme

10. Ravnik J, Škerget L, Žunič Z. Combined single domain and subdomain BEM for 3D laminar viscous flow. *Engineering Analysis with Boundary Elements* 2008; DOI: 10.1016/j.enganabound.2008.06.006.
11. 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* 2008; **51**:4517–4527.
12. Greengard L, Rokhlin V. A fast algorithm for particle simulations. *Journal of Computational Physics* 1987; **73**:325–348.
13. Bui TT, Ong ET, Khoo BC, Klaseboer E, Hung KC. A fast algorithm for modeling multiple bubbles dynamics. *Journal of Computational Physics* 2006; **216**:430–453.
14. Gumerov NA, Duraiswami R. Fast multipole method for the biharmonic equation in three dimensions. *Journal of Computational Physics* 2006; **215**:363–383.
15. Popov V, Power H, Walker SP. Numerical comparison between two possible multipole alternatives for the BEM solution of 3D elasticity problems based upon Taylor series expansions. *Engineering Analysis with Boundary Elements* 2003; **27**:521–531.
16. Ravnik J, Škerget L, Žunič Z. Comparison of wavelet and fast multipole data sparse approximations for boundary-domain integral equations of Poisson type. *Computer Methods in Applied Mechanics and Engineering* 2009; submitted.
17. Ravnik J, Škerget L, Hriberšek M. 2D velocity vorticity based LES for the solution of natural convection in a differentially heated enclosure by wavelet transform based BEM and FEM. *Engineering Analysis with Boundary Elements* 2006; **30**:671–686.
18. Žunič Z, Hriberšek M, Škerget L, Ravnik J. 3-D boundary element–finite element method for velocity–vorticity formulation of the Navier–Stokes equations. *Engineering Analysis with Boundary Elements* 2007; **31**:259–266.
19. Press WH, Teukolsky SA, Vetterling WT, Flannery BP. *Numerical Recipes—The Art of Scientific Computing* (2nd edn). Cambridge University Press: Cambridge, 1997.
20. Paige CC, Saunders MA. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software* 1982; **8**:43–71.
21. Shankar PN, Deshpande MD. Fluid mechanics in the driven cavity. *Annual Review of Fluid Mechanics* 2000; **32**:93–136.
22. Koseff JR, Street RL. The lid-driven cavity flow: a synthesis of qualitative and quantitative observations. *Journal of Fluids Engineering* 1984; **106**:385–389.
23. Yang JY, Chen YN, Yang SC, Hsu CA. Implicit weighted ENO schemes for the three-dimensional incompressible Navier–Stokes equations. *Journal of Computational Physics* 1998; **146**(1):464–487.
24. Lo DC, Young DL, Murugesan K, Tsai CC, Gou MH. Velocity–vorticity formulation for 3D natural convection in an inclined cavity by DQ method. *International Journal of Heat and Mass Transfer* 2007; **50**:479–491.
25. Tric E, Labrosse G, Betrouni M. A first incursion into the 3D structure of natural convection of air in a differentially heated cubic cavity, from accurate numerical simulations. *International Journal of Heat and Mass Transfer* 2000; **43**:4034–4056.

DOI: 10.1002/nme