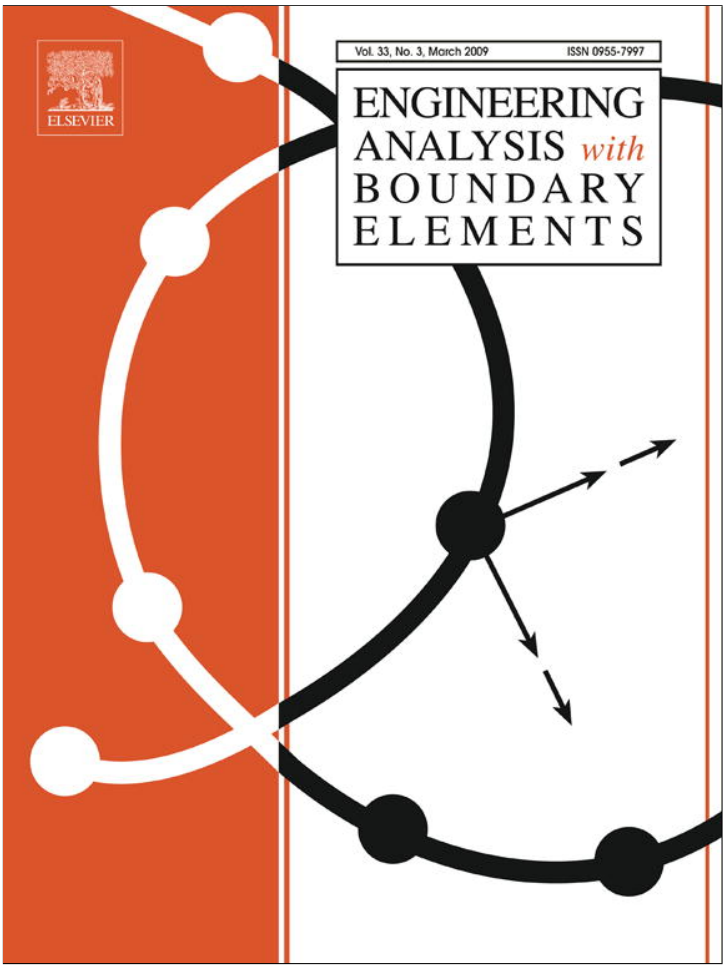

Engineering Analysis with Boundary Elements 33 (2009) 420–424

Contents lists available at ScienceDirect

Engineering Analysis with Boundary Elements

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

Combined single domain and subdomain BEM for 3D laminar viscous flow

J. Ravnik *, L. Škerget, Z. Žunič

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

ARTICLE INFO

Article history:
Received 10 July 2007
Accepted 16 June 2008
Available online 15 August 2008

PACS:
47.11.+j
44.25.+f

Keywords:
Subdomain boundary element method
Velocity–vorticity formulation
Laminar viscous fluid flow
Lid driven cavity

ABSTRACT

A subdomain boundary element method (BEM) using a continuous quadratic interpolation of function and discontinuous linear interpolation of flux is presented for the solution of the vorticity transport equation and the kinematics equation in 3D. By employing compatibility conditions between subdomains an over-determined system of linear equations is obtained, which is solved in a least squares manner. The method, combined with the single domain BEM, is used to solve laminar viscous flows using the velocity vorticity formulation of Navier–Stokes equations. The versatility and accuracy of the method are proven using the 3D lid driven cavity test case.

1. Introduction

The main advantage of the boundary element method (BEM) is the ability to solve partial differential equations by solving for boundary unknowns only, omitting the discretization of the domain. This advantage is lost when a suitable fundamental solution cannot be found and a domain contribution remains in the integral equation. This happens when solving the vorticity transfer equation or the kinematics equation. Several procedures have been proposed to avoid this difficulty, for instance methods based on the expansion of the integral kernel [1], dual reciprocity method [2]or compression of the resulting full matrices [3]. We employed the subdomain method [4], in which the domain is discretized into subdomains and BEM is applied on each subdomain. The resulting integral matrices are sparse and as such may be stored efficiently and enable fast algebraic operations.

In the last decade our research group developed several BEM based numerical algorithms for the solution of viscous incompressible and compressible, laminar and turbulent flows by solving the velocity–vorticity formulation of the Navier–Stokes equations. With the aim of increasing computational grid density and decreasing computational times the single domain BEM approach has been coupled by several numerical procedures, such as wavelet compression [5,6].

In this work, we are presenting a 3D viscous laminar flow solver, which is based on a combination of single domain BEM and subdomain BEM. 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. In the paper we present subdomain BEM for the solution of the vorticity transport and the kinematics equations. Ramšak and Škerget [9] employed a similar approach for the 3D Laplace equation, but using a lower order interpolation scheme.

The solution of viscous flow using a velocity–vorticity formulation of Navier–Stokes equations requires an iterative scheme for solution of both velocity and vorticity fields. The main challenge lies in the determination of boundary vorticity values, which are needed for the solution of the vorticity transport equation. Several different approaches have been proposed for the determination of vorticity on the boundary. Daube [10] used an

* Corresponding author.
E-mail addresses: jure.ravnik@uni-mb.si (J. Ravnik), leo@uni-mb.si (L. Škerget), zoran.zunic@uni-mb.si (Z. Žunič).

0955-7997/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.enganabound.2008.06.006

influence matrix technique to enforce both the continuity equation and the definition of the vorticity in the treatment of the 2D incompressible Navier–Stokes equations. Liu [11] recognized that the problem is even more severe when he extended it to three dimensions. Lo et al. [12] used a differential quadrature method to calculate vorticity from its definition to obtain a solution of a natural convection problem. We will use single domain BEM solution of the kinematics equation for determination of boundary vorticity. This approach was introduced by Škerget et al. [13] in 2D and used coupled by FEM in 3D by Žunič et al. [14].

2. BEM solution of the Navier–Stokes equations in velocity–vorticity formulation

In this paper we assume an incompressible viscous Newtonian fluid with constant material properties. Vorticity $\vec{\omega}$ is defined as the curl of the velocity $\vec{\omega} = \vec{\nabla} \times \vec{v}$. Both velocity and vorticity fields are divergence free. The viscous fluid flow is governed by the kinematics equation

$$\nabla^2 \vec{v} + \vec{\nabla} \times \vec{\omega} = 0, \tag{1}$$

which is a vector elliptic partial differential equation of Poisson type and links the velocity and vorticity fields for every point in space and time. The kinetic aspect of fluid movement is governed by the vorticity transport equation, written in non-dimensional form:

$$\frac{\partial \vec{\omega}}{\partial t} + (\vec{v} \cdot \vec{\nabla})\vec{\omega} = (\vec{\omega} \cdot \vec{\nabla})\vec{v} + \frac{1}{Re}\nabla^2 \vec{\omega}, \tag{2}$$

with the Reynolds number denoted by *Re*. Eq. (2) equates the advective vorticity transport on the left-hand side with the vortex twisting and stretching term and the diffusion term on the right-hand side. In this paper we are dealing with steady flows only, which makes $\partial\vec{\omega}/\partial t = 0$. The system of equations (1) and (2) is solved in a nonlinear loop of three steps. Firstly, calculate boundary vorticity values by solving the kinematics equation by single domain BEM. Secondly, calculate domain velocity values by solving the kinematics equation by subdomain BEM and finally solve vorticity transport equation for domain vorticity values using the boundary values from the solution of the kinematics equation by subdomain BEM.

The boundary conditions required to obtain the solution are prescribed velocity on the boundary. The unknown boundary conditions for the vorticity transport equation are calculated as a part of the algorithm using single domain BEM.

2.1. Solution of the steady vorticity transport equation by subdomain BEM

The integral form of the steady vorticity transport equation (2) may be written for the j th component of the vorticity vector as [13]

$$c(\vec{\vartheta})\omega_j(\vec{\vartheta}) + \int_\Gamma \omega_j \vec{\nabla} u^* \cdot \vec{n}\, \mathrm{d}\Gamma = \int_\Gamma u^* q_j \,\mathrm{d}\Gamma + Re \int_\Gamma \vec{n} \cdot \{u^*(\vec{v}\omega_j - \vec{\omega}v_j)\}\,\mathrm{d}\Gamma - Re \int_\Omega (\vec{v}\omega_j - \vec{\omega}v_j) \cdot \vec{\nabla} u^* \,\mathrm{d}\Omega, \tag{3}$$

where $\vec{r}$ is the location point, $\vec{\vartheta}$ is the collocation point, $u^\star = \frac{1}{4}\pi|\vec{\vartheta} - \vec{r}|$ and $\vec{q}$ is the vorticity flux vector $q_j = \vec{n} \cdot \vec{\nabla}\omega_j$.

The field functions as well as the products of velocity and vorticity field components are interpolated within elements using shape functions. The mesh elements used in this work are hexahedrons (Fig. 1). Quadratic interpolation of function within hexahedron is employed $\omega_j(\xi,\eta) = \sum_{i=1}^{27} \Phi_i \omega_j^i$, where Φ_i are the standard shape functions for a 27 node Lagrangian domain element. On each face of the hexahedron (Fig. 2) we used continuous quadratic interpolation for function $\omega_j(\xi,\eta) = \sum_{i=1}^{9} \varphi_i \omega_j^i$ and discontinuous linear interpolation for flux $q_j(\xi,\eta) = \sum_{i=1}^{4} \phi_i q_j^i$, where ω_j^i are function values in each function node, q_j^i are flux values in flux nodes and (ξ,η) are local coordinate system axes. The shape functions for function φ_i are the standard shape functions for a quadratic nine node Lagrangian surface

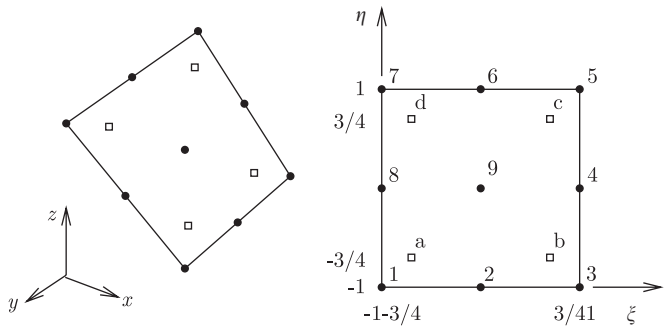

Fig. 2. A boundary element shown with nodes for discontinuous linear interpolation for flux (squares) and nodes for continuous quadratic interpolation for function (circles). Cartesian $\mathbb{R}^3$ space is shown on the left, local coordinate system on the right.

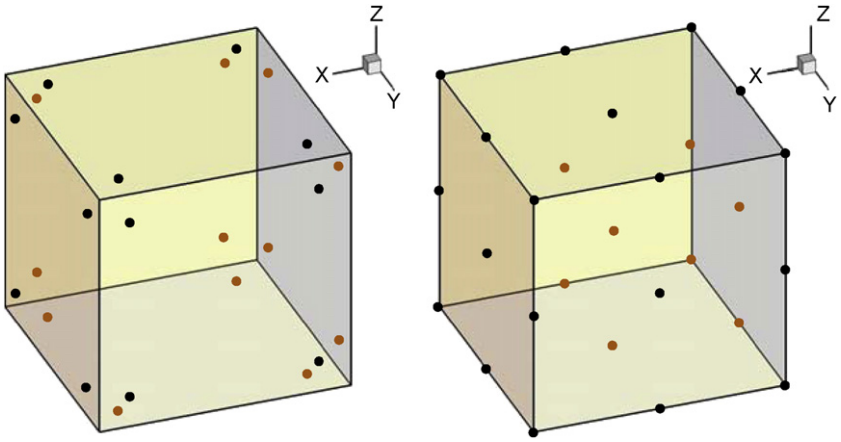

Fig. 1. A hexahedral mesh element with distribution of nodes: (left) nodes for interpolation of flux, (right) nodes for interpolation of function.

element, while the shape functions for flux are

$$\phi_1 = \tfrac{4}{9}(\xi - \tfrac{3}{4})(\eta - \tfrac{3}{4}), \quad \phi_2 = -\tfrac{4}{9}(\xi + \tfrac{3}{4})(\eta - \tfrac{3}{4}),$$
$$\phi_3 = \tfrac{4}{9}(\xi + \tfrac{3}{4})(\eta + \tfrac{3}{4}), \quad \phi_4 = -\tfrac{4}{9}(\xi - \tfrac{3}{4})(\eta + \tfrac{3}{4}). \tag{4}$$

The geometry of the hexahedron is defined by the eight corner nodes, thus each surface is defined by four nodes (numbers 1, 3, 5, 7 in Fig. 2). One may find the location of flux nodes (a, b, c ,d) by using the following transformation:

$$\begin{vmatrix} x_a & y_a & z_a \\ x_b & y_b & z_b \\ x_c & y_c & z_c \\ x_d & y_d & z_d \end{vmatrix} = \frac{1}{64} \begin{vmatrix} 49 & 7 & 1 & 7 \\ 7 & 49 & 7 & 1 \\ 1 & 7 & 49 & 7 \\ 7 & 1 & 7 & 49 \end{vmatrix} \cdot \begin{vmatrix} x_1 & y_1 & z_1 \\ x_3 & y_3 & z_3 \\ x_5 & y_5 & z_5 \\ x_7 & y_7 & z_7 \end{vmatrix}. \tag{5}$$

All flux nodes are located within boundary elements, none are located at corners and edges—thus the unit normal and the flux value are unambiguously defined in each flux node.

With this, the integrals required to solve the vorticity transport equation are

$$H^i_{\vartheta,\Gamma} = \int_\Gamma \varphi_i \vec{\nabla} u^\star \cdot \vec{n}\, \mathrm{d}\Gamma, \quad G^i_{\vartheta,\Gamma} = \int_\Gamma \phi_i u^\star \, \mathrm{d}\Gamma,$$

$$\vec{A}^i_{\vartheta,\Gamma} = \int_\Gamma \varphi_i \vec{n} u^\star \, \mathrm{d}\Gamma, \quad \vec{D}^i_{\vartheta,\Omega} = \int_\Omega \Phi_i \vec{\nabla} u^\star \, \mathrm{d}\Omega \tag{6}$$

and the discrete equation is

$$c(\vec{\vartheta})\omega_j(\vec{\vartheta}) + \sum_{i=1}^{26} \omega^i_j H^i_{\vartheta,\Gamma} = \sum_{i=1}^{24} q_i G^i_{\vartheta,\Gamma} + Re \sum_{i=1}^{26} (\vec{v}\omega_j - \vec{\omega}v_j)_i \cdot \vec{A}^i_{\vartheta,\Gamma} - Re \sum_{i=1}^{27} (\vec{v}\omega_j - \vec{\omega}v_j)_i \cdot \vec{D}^i_{\vartheta,\Omega}. \tag{7}$$

In order to calculate the integrals, a Gaussian quadrature algorithm is used. The integrals are calculated in local coordinate system via weighted summation of up to 48 integration points per coordinate axis. In the case of high aspect ratios of hexahedral elements, the boundary elements are divided into parts whose aspect ratio is approximately equal to one. Calculation of the free coefficient $c(\vec{\vartheta})$ is preformed indirectly, by considering a known solution of the rigid body movement problem.

In order to set up a system of equations the source point is set in all function and flux nodes of all mesh elements. Additionally, the source point is set into a node in the center of the hexahedron, where the function value may be obtained explicitly from known boundary values. The value in the center of the element is needed for the formation of the right-hand side of the single domain BEM solution of the kinematics equation.

Thus all in all we have 51 equations for each element. The corresponding integrals are stored in rectangular matrices, which have 51 times the number of elements rows. The $[H]$ integral matrix has 26 columns, while the $[G]$ matrix has 24 columns. Since neighboring elements share nodes and since boundary conditions on the outer boundaries of the domain are prescribed, we obtain an over-determined system of equations. The system is sparse. We store the system matrix in compressed row storage format. The system is solved in a least squares manner [15]. Using a combination of continuous and discontinuous interpolation schemes increases the required number of collocation points and results in an increase of the number of non-zero elements in the matrices and as a results increases computational time. However, the storage requirements of sub-domain BEM matrices are much smaller than the full matrices of the single domain part of the algorithm, thus the overall CPU time and storage requirements are not greatly increased.

2.2. Solution of the kinematics equation for domain velocity by subdomain BEM

The integral form of the kinematics equation without derivatives of the velocity and vorticity fields takes the following form (for derivation, see [6, Eqs. (19)–(24)]):

$$c(\vec{\vartheta})\vec{v}(\vec{\vartheta}) + \int_\Gamma \vec{v}\vec{\nabla}u^\star \cdot \vec{n}\, \mathrm{d}\Gamma = \int_\Gamma \vec{v} \times (\vec{n} \times \vec{\nabla})u^\star \, \mathrm{d}\Gamma + \int_\Omega (\vec{\omega} \times \vec{\nabla}u^\star)\, \mathrm{d}\Omega. \tag{8}$$

The boundary integrals on the left-hand side are stored in the $[H]$ matrix, the domain integrals on the right-hand side are the $[\vec{D}]$ matrices. We define the boundary integral on the right-hand side as $[\vec{H}^t]$ integrals in the following manner:

$$\vec{H}^t_{\vartheta,\Gamma,i} = \int_\Gamma \varphi_i (\vec{n} \times \vec{\nabla})u^\star \, \mathrm{d}\Gamma. \tag{9}$$

Since there are no fluxes in the equation, the source point is set to function nodes only. Let vectors of nodal values of field functions be denoted by curly brackets. The discrete kinematics equation written in component wise form is

$$[H]\{v_x\} = [H^t_z]\{v_y\} - [H^t_y]\{v_z\} + [D_z]\{\omega_y\} - [D_y]\{\omega_z\}, \tag{10}$$

$$[H]\{v_y\} = [H^t_x]\{v_z\} - [H^t_z]\{v_x\} - [D_z]\{\omega_x\} + [D_x]\{\omega_z\}, \tag{11}$$

$$[H]\{v_z\} = [H^t_y]\{v_x\} - [H^t_x]\{v_y\} + [D_y]\{\omega_x\} - [D_x]\{\omega_y\}. \tag{12}$$

Using $[H]$ as the system matrix the three linear systems of equations must be solved repeatedly, until convergence is achieved. This is due to the fact that the right-hand sides depend on velocity as well. No under-relaxation was needed in our simulations. We notice, that the $[H]$ and $[\vec{D}]$ integral matrices are needed for the vorticity transport equation as well and are thus used twice.

All in all the subdomain BEM solution of the kinematics and vorticity transport equations requires the calculation and storage of $[H]$, $[G]$, $[\vec{A}]$, $[\vec{H}^t]$ and $[\vec{D}]$ matrices. The total number of integrals that must be calculated and stored is 12 540 times the number of mesh elements. In comparison with the single domain BEM this is a very small number. The single domain BEM would require at least three matrices with the number of elements equal to the number of nodes squared. On a cubic mesh with $10 \times 10 \times 10$ elements with 9261 nodes, the ratio between subdomain BEM storage requirements and single domain BEM storage requirements would be approximately 0.04 and on a $20 \times 20 \times 20$ mesh it would be approximately 0.007.

2.3. Solution of the kinematics equation for boundary vorticity by single domain BEM

In order to use the kinematics equation to obtain boundary vorticity values, we rewrite the system of equations (10)–(12) in a tangential form by multiplying the system with a normal in the source point. This improves the conditioning of the system. This approach has been proposed by Škerget and used in 2D by Škerget et al. [13] and in 3D by Žunič et al. [14]. We employed the same procedure as Žunič et al. [14] with the difference of using the second order shape functions, while they used a first order interpolation scheme.

3. 3D lid driven cavity

The proposed numerical scheme was tested on the 3D lid driven cavity numerical example. Flow in a 3D lid driven cavity is

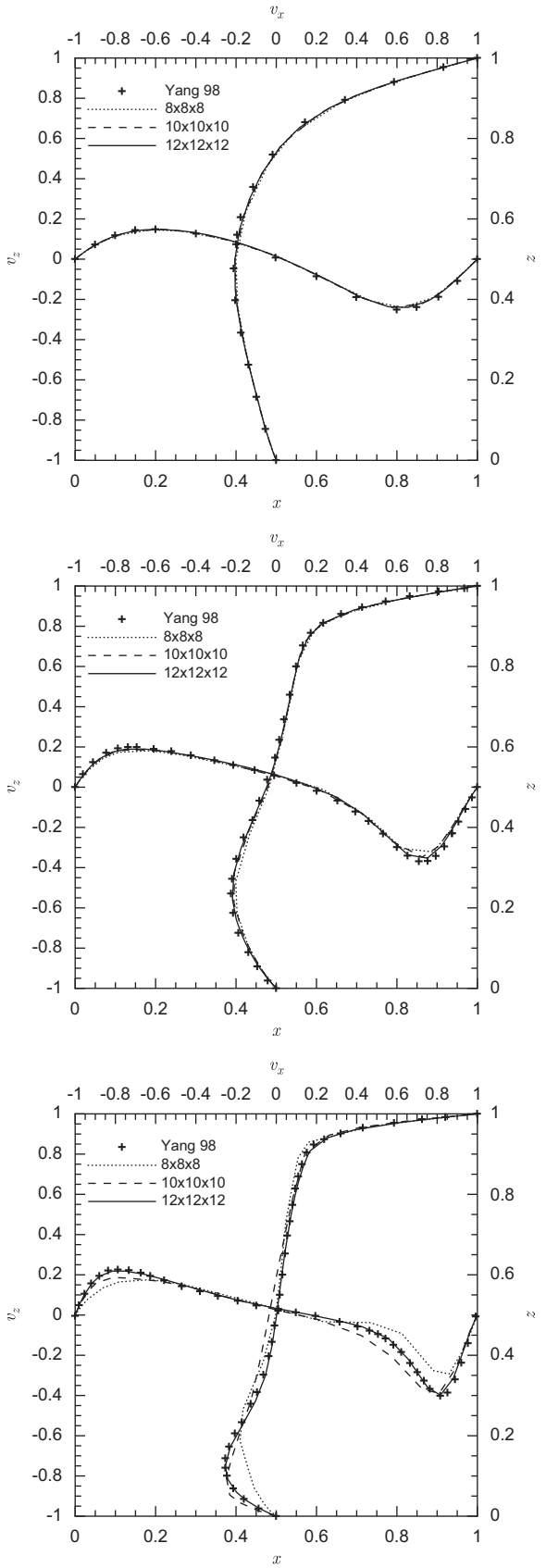

Fig. 3. Flow in a 3D lid driven cavity, comparison of present results on different meshes with Yang et al. [17] results; $Re = 100$ (top), $Re = 400$ (middle), $Re = 1000$ (bottom).

one of the standard benchmark test cases used in development of flow solvers. The domain as well as the boundary conditions is 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 [16].

The simulation was preformed 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 wall except the top wall, where a constant velocity in x direction is prescribed $\vec{v} = (1,0,0)$. 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 left and right walls, $\omega_y = 0$ on front and back walls and $\omega_z = 0$ on top and bottom walls.

Simulations were run on meshes of 8^3 elements with 4913 nodes, 10^3 elements with 9261 nodes and 12^3 elements with 15 625 nodes. The elements were concentrated towards the eight corners. 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 $Re = 100$, 400 and 1000.

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. We compared velocity profiles within the cavity in the $y = 0.5$ plane with the results of Yang et al. [17]. Fig. 3 shows good agreement for $Re = 100$, 400 on all meshes, while $Re = 1000$ profiles are in good agreement with the reference only on the dense mesh. The high Reynolds number induces high gradients, which can only be described correctly on a dense mesh. Comparison with the results of Žunič et al. [14] BEM–FEM algorithm, which used a linear interpolation scheme, we observe an increase of accuracy. The present velocity profiles calculated on a coarser grid are in much better accordance with the benchmark. The improvement is especially prominent in the $Re = 1000$ case.

4. Conclusions

We developed a 3D subdomain boundary element method based on the continuous quadratic interpolation of function and discontinuous linear interpolation of flux. Using discontinuous boundary elements for flux enabled us to avoid the undefined flux values in the corners and edges. The resulting over-determined system of linear equations was solved in a least squares manner.

By combining the subdomain BEM with the single domain BEM we are able to solve velocity–vorticity formulation of Navier–Stokes equations and simulate viscous laminar flow in 3D. The method was successfully tested by simulating the flow in a 3D lid driven cavity up to Reynolds number $Re = 1000$.

Acknowledgments

The first author gratefully acknowledges the support of the Wessex Institute of Technology during his stay at Ashurst Lodge in the UK.

References

[1] Bebendorf M. Approximation of boundary element matrices. Numer Math 2000;86:565–89.
[2] Jumarhon B, Amini S, Chen K. On the boundary element dual reciprocity method. Eng Anal Bound Elem 1997;20:205–11.

[3] Eppler K, Harbrecht H. Fast wavelet BEM for 3D electromagnetic shaping. Appl Numer Math 2005;54:537–54.
[4] Popov V, Power H, Škerget L, editors. Domain Decomposition Techniques for Boundary Elements: Applications to Fluid Flow. WIT press; 2007.
[5] Ravnik J, Škerget L, Hriberšek M. The wavelet transform for BEM computational fluid dynamics. Eng Anal Bound Elem 2004;28:1303–14.
[6] 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. Eng Anal Bound Elem 2006;30:671–86.
[7] Ong E, Lim K. Three-dimensional singular boundary element method for corner and edge singularities in potential problems. Eng Anal Bound Elem 2005;29:175–89.
[8] Gao XW, Davies TG. 3D multi-region BEM with corners and edges. Int J Solids Struct 2000;37:1549–60.
[9] Ramšak M, Škerget L. 3D multidomain BEM for solving the Laplace equation. Eng Anal Bound Elem 2007;31:528–38.
[10] Daube O. Resolution of the 2D Navier–Stokes equations in velocity–vorticity form by means of an influence matrix technique. J Comput Phys 1992;103:402–14.
[11] Liu CH. Numerical solution of three-dimensional Navier Stokes equations by a velocity–vorticity method. Int J Numer Meth Fl 2001;35:533–57.
[12] Lo D, Young D, Murugesan K, Tsai C, Gou M. Velocity–vorticity formulation for 3D natural convection in an inclined cavity by DQ method. Int J Heat Mass Transfer 2007;50:479–91.
[13] Škerget L, Hriberšek M, Žunič Z. Natural convection flows in complex cavities by BEM. Int J Numer Meth Heat Fluid Fl. 2003;13:720–35.
[14] Žunič Z, Hriberšek M, Škerget L, Ravnik J. 3-D boundary element-finite element method for velocity–vorticity formulation of the Navier–Stokes equations. Eng Anal Bound Elem 2007;31:259–66.
[15] Paige CC, Saunders MA. LSQR: an algorithm for sparse linear equations and sparse least squares. ACM Trans Math Software 1982;8:43–71.
[16] Shankar P, Deshpande MD. Fluid mechanics in the driven cavity. Annu Rev Fluid Mech 2000;32:93–136.
[17] Yang J, Yang YCSC, Hsu C. Implicit weighted ENO Schemes for the three-dimensional incompressible Navier–Stokes equations. J Comput Phys 1998;146(1):464–87.