INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2004; **46**:815–847
Published online 14 September 2004 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.776

A subdomain boundary element method for high-Reynolds laminar flow using stream function-vorticity formulation

Matjaž Ramšak*,† and Leopold Škerget

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

SUMMARY

The paper presents a new formulation of the integral boundary element method (BEM) using subdomain technique. A continuous approximation of the function and the function derivative in the direction normal to the boundary element (further 'normal flux') is introduced for solving the general form of a parabolic diffusion-convective equation. Double nodes for normal flux approximation are used. The gradient continuity is required at the interior subdomain corners where compatibility and equilibrium interface conditions are prescribed. The obtained system matrix with more equations than unknowns is solved using the fast iterative linear least squares based solver. The robustness and stability of the developed formulation is shown on the cases of a backward-facing step flow and a square-driven cavity flow up to the Reynolds number value 50 000. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: boundary element method; incompressible viscous flow; stream function-vorticity formulation; backward-facing step flow; driven cavity flow

1. INTRODUCTION

The *stability* of boundary element method (BEM) at high Reynolds number value flows is common problem to all BEM formulations. In the available literature, the highest Reynolds number computed using BEM is found to be 15 000 for the well-known test case of the driven cavity flow by authors Rek and Škerget [1]. In their work and also in earlier papers dealing with the BEM, the solution domain is treated as a single entity with the system matrix full and non-symmetric demanding a great amount of computer memory and CPU. This matrix is expensive to solve for large problems. The economics of BEM computation is considerably improved using the subdomain technique (also known as a method of domain decomposition or multidomain method) in which the original domain is divided into subdomains and the full integral formula is applied on each of them, see References [2, 3]. Let us review some of the highest Reynolds number value computed for the case of driven cavity flow using BEM

*Correspondence to: M. Ramšak, Faculty of Mechanical Engineering, University of Maribor, Smetanova ulica 17, SI-2000 Maribor, Slovenia.
†E-mail: matjaz.ramsak@uni-mb.si

Received 5 May 2003
Revised 6 April 2004

and subdomain technique below. $Re = 3200$ by Hriberšek and Škerget [4] who used velocity vorticity formulation. Practically the same formulation is used by Young *et al.* [5], but only up to the $Re = 2000$. The poly-region BEM using penalty formulation of the Navier–Stokes equations by Grigoriev and Dargush [6] reached $Re = 5000$, which is the highest value found when using subdomain technique. The dual reciprocity BEM was used by Florez and Power [7] for computations up to $Re = 1000$. In this paper, we are going to show that the stability of BEM is not decreased by using the subdomain technique.

By using the *subdomain technique*, the compatibility and equilibrium interface conditions are enforced at the common interface between adjacent subdomains. The implementation of the subdomain technique in the limiting case, i.e. the subdomain becomes similar to the element in the FEM, the resulting coefficient matrix is as sparse as that of the FEM, and thus suitable for fast iterative solvers, see Reference [4].

Overdetermined system matrix: While using continuous element interpolation of high orders, the application of matching conditions at the common interface leads to an overdetermined system of algebraic equations. If the overdetermined matrix is reduced to a squared one by ignoring some of the boundary integral equations, the reduced system matrix is not consistent with the initial one, resulting in an unstable numerical scheme for Reynolds number higher than 1000 in the case of driven cavity flow, see Grigoriev and Fafurin [8]. In their next work [6], the overdeterminancy of the system matrix is relieved using the grid with hexagonal subdomains. Thus, the number of linear independent equations written for the internal vertex node becomes equal to the degrees of freedom. The stability of the algorithm is demonstrated in the case of driven cavity flow at Reynolds number value 5000 using primitive variables and pressure penalty function. In the previous work of Ramšak and Škerget [9] the initial overdetermined matrix is not reduced but solved in linear least-squares sense using fast iterative solver of Paige and Saunders [10]. The same solver by Paige and Saunders is also applied in the subdomain dual reciprocity method by authors Florez and Power [7], but only up to $Re = 1000$ for the case of driven cavity flow.

A stream function-vorticity formulation of Navier–Stokes equations for 2D incompressible flows provides no direct boundary condition for vorticity at the zero-slip walls. In practice, the vorticity boundary condition is computed locally from near wall distribution of stream function values together with the vorticity definition in general not satisfying vorticity solenoidity condition, see Ghia *et al.* [11]. The significant advantage of the present BEM numerical scheme is compute boundary vorticities implicitly from the stream function transport equation. Thus, no differencing is needed from near boundary distribution of stream function values, which may decrease accuracy.

The main aim of the present work is to develop a stable subdomain BEM formulation for solving a general form of a parabolic diffusion-convective equation with convection dominating over diffusion. The developed numerical technique is general enough and well prepared for solving any problems governed by this type of differential equations and could be extended on 3D.

The structure of the paper follows. In Section 2, the governing equations of stream function-vorticity formulation are stated. In Section 3, the general form of a differential parabolic diffusion-convective equation is written in order to make the starting point for the following sections clearer. The integral representation is explained briefly in Section 4 while the extended discrete representation is discussed in Section 5. Some details regarding the solution of the overdetermined system matrix are shown in Section 6. The developed BEM formulation is

applied to a stream function-vorticity formulation of Navier–Stokes equations. Computational algorithm is described in Section 7. In Section 8 the numerical examples are represented with a detailed discussion, to be followed by conclusions.

2. GOVERNING EQUATIONS

In this paper, the incompressible Navier–Stokes equations for stream function-vorticity formulation are solved. In two dimensions, the stream function Ψ equation can be written as the elliptic PDE

$$\frac{\partial^2 \Psi}{\partial x_i \partial x_i} = -\omega \tag{1}$$

and the vorticity ω transport equation as the parabolic PDE

$$\frac{\partial \omega}{\partial t} + v_j \frac{\partial \omega}{\partial x_j} = \nu \frac{\partial^2 \omega}{\partial x_i \partial x_i} \tag{2}$$

where the velocity vector v_j is

$$v_j = \left(\frac{\partial \Psi}{\partial y}, -\frac{\partial \Psi}{\partial x} \right) \tag{3}$$

2.1. *Boundary conditions*

The boundary condition at the walls yields the known stream function value

$$\Psi = \bar{\Psi} \tag{4}$$

and its derivative in the normal direction to the wall

$$\frac{\partial \Psi}{\partial n} = \overline{v_t} \tag{5}$$

where $\overline{v_t}$ is the known tangential velocity. As it is well known, this provides no direct condition for the wall vorticity. If the numerical techniques like FEM and FDM are used, the wall vorticity ω_J is computed locally from a near wall stream function distribution. In the work of Ghia *et al.* [11] the second-order accurate formula is used

$$\omega_J = -\frac{\partial^2 \Psi}{\partial y^2} = -\frac{\Psi_{J+1} - 2\Psi_J + \Psi_{J-1}}{\Delta y^2}$$

generally not satisfying the vorticity solenoidity condition. In the present BEM numerical algorithm, wall vorticities ω_Γ are computed implicitly from the stream function transport equation (1), because Ψ and its normal derivatives are both known as boundary conditions. Thus, the BEM integral equations are available at the solution domain boundary Γ for the unknown vorticities ω_Γ. This is the significant advantage of BEM against other mentioned numerical techniques.

3. DIFFERENTIAL REPRESENTATION OF PARABOLIC DIFFUSION-CONVECTIVE EQUATION

The transport equation for stream function and vorticity can be written in general form as

$$a\frac{\partial^2 u}{\partial x_j \partial x_j} - \frac{\partial u}{\partial t} - \frac{\partial v_j u}{\partial x_j} + S(x_j, u) = 0 \tag{6}$$

where u is the scalar field function, a is a constant diffusivity and S is source term.

By using finite difference approximation the field function time derivative at time level l and time increment $\Delta t = t^l - t^{l-1}$ is written as

$$\frac{\partial u}{\partial t} \approx \frac{u^l - u^{l-1}}{\Delta t}$$

The contribution of the previous time step value is added to the source term S as

$$S = \frac{u^{l-1}}{\Delta t}$$

A semi-implicit approach based on the backward Euler scheme as the linearization technique for the convective term, the differential representation of the parabolic diffusion-convective equation can be written in the final form as

$$a\frac{\partial^2 u^l}{\partial x_j^2} - \frac{u^l}{\Delta t} - \frac{\partial v_j^{l-1} u^l}{\partial x_j} + S^{l-1} = 0 \tag{7}$$

For the sake of clarity the superscript l for the time level shall be omitted in the future text.

Boundary conditions on $\Gamma = \Gamma_1 + \Gamma_2$ and initial conditions in domain Ω must be known

$$u = \bar{u}, \text{ in } \Gamma_1 \text{ for } t \geqslant t_o \tag{8}$$

$$\frac{\partial u}{\partial n} = \overline{\frac{\partial u}{\partial n}} \text{ in } \Gamma_2 \text{ for } t \geqslant t_o \tag{9}$$

$$u = \bar{u}_o \text{ in } \Omega \text{ for } t = t_o \tag{10}$$

In order to make system matrices sparse and block banded the subdomain technique is used. Therefore, at the interface boundaries between subdomains I and II the compatibility interface condition for u has to be applied

$$u\,|_I = u\,|_{II} \tag{11}$$

and the equilibrium interface condition

$$\lambda \left.\frac{\partial u}{\partial n}\right|_I = -\lambda \left.\frac{\partial u}{\partial n}\right|_{II} \tag{12}$$

where λ is the diffusion factor for the sake of generality. In the case of diffusion factor continuity ($\lambda_I = \lambda_{II}$) the equilibrium interface condition is reduced so that it becomes equivalent to the normal fluxes.

4. INTEGRAL REPRESENTATION OF PARABOLIC DIFFUSION-CONVECTIVE EQUATION

The general form of the differential parabolic diffusion-convective equation (7) can be transformed into an equivalent integral statement, see References [12, 13],

$$\begin{aligned} 0 = &-c(\xi)u(\xi) - a\int_\Gamma u\frac{\partial u^\star}{\partial n}\,\mathrm{d}\Gamma + a\int_\Gamma \frac{\partial u}{\partial n}u^\star\,\mathrm{d}\Gamma \qquad \text{diffusion} \\ &-\int_\Gamma uv_j n_j u^\star\,\mathrm{d}\Gamma + \int_\Omega uv_j\frac{\partial u^\star}{\partial x_j}\,\mathrm{d}\Omega \qquad \text{convection} \\ &+\int_\Omega Su^\star\,\mathrm{d}\Omega, \qquad \text{source } S \end{aligned} \tag{13}$$

where n_j is the unit normal vector to the boundary element at the nodal point. The first Green identity for arbitrary field function f_j has been applied to the convection term

$$\int_\Omega \frac{\partial f_j}{\partial x_j}u^\star\,\mathrm{d}\Omega = \int_\Gamma f_j n_j u^\star\,\mathrm{d}\Gamma - \int_\Omega f_j\frac{\partial u^\star}{\partial x_j}\,\mathrm{d}\Omega$$

The variable $u^\star$ is the modified Helmholtz fundamental solution, i.e. the solution of the equation

$$\frac{\partial^2 u^\star}{\partial x_j \partial x_j} - \mu^2 u^\star + \delta(\xi, s) = 0 \tag{14}$$

and given for the plane case as

$$\begin{aligned} u^\star &= \frac{1}{2\pi}K_0(\mu r) \\ \frac{\partial u^\star}{\partial x_j}n_j &= \frac{r_j n_j}{2\pi r^2}\mu r K_1(\mu r) \end{aligned} \tag{15}$$

where the parameter μ is defined as

$$\mu^2 = \frac{1}{a\Delta t} \tag{16}$$

K_0 and K_1 are the modified Bessel functions of the second kind and $r_j(\xi, s)$ is the vector from the source point ξ to the reference field point s, e.g. $r_j = x_j(\xi) - x_j(s)$ for $j = 1, 2$, while r is its magnitude $r = |\, r_j \,|$.

Other fundamental solutions can be applied, e.g. elliptic diffusion-convective, see Reference [14], but in all presented numerical examples, the modified Helmholtz fundamental solution is used.

5. DISCRETE REPRESENTATION OF PARABOLIC DIFFUSION-CONVECTIVE EQUATION

5.1. Introduction

Notation: The integral equation (13) is valid for any arbitrary geometry. In order to distinguish the solution domain Ω and its exterior boundary Γ from the subdomain geometry, from now on the subdomain domain is denoted by Ω_s and the subdomain boundary by Γ_s consistently.

Discontinuous elements: The development of the present numerical scheme originated from the work of authors Škerget *et al.* [14]. The field function u and its normal flux $\partial u/\partial n$ are computed explicitly from discretized BEM integral equations in the implicit matrix system. For this reason the unit normal vector in the normal flux interpolation nodal point has to be known. In the vertex of the rectangular subdomain, the normal direction is undefined. In Reference [14] the problem is solved by using the discontinuous approximation of function and normal flux, see left-hand side of Figure 1. By definition, the first and major disadvantage is the discontinuous field function approximation and therefore conservation of field function is not preserved. The second disadvantage is a significant increase in the number of nodal points, which slows down the rate of solution convergence and increases the computer memory demands. The third disadvantage is non-unique approximation of the field function over the subdomain using discontinuous boundary nodal points. Either a non-symmetrical interpolation has to be used or new additional nodal points have to be introduced for domain approximation. Thus, the discretization of the boundary and domain approximation of the convective term over subdomain

$$-\int_{\Gamma_s} uv_j n_j u^\star \, \mathrm{d}\Gamma_s + \int_{\Omega_s} uv_j \frac{\partial u^\star}{\partial x_j} \, \mathrm{d}\Omega_s$$

do not use the same function interpolation nodal points, causing unstability at the convection dominated flows. Using the discontinuous numerical scheme developed in Reference [14] the highest computed Reynolds number value is 3200 in the case of the flow in a closed square driven cavity.

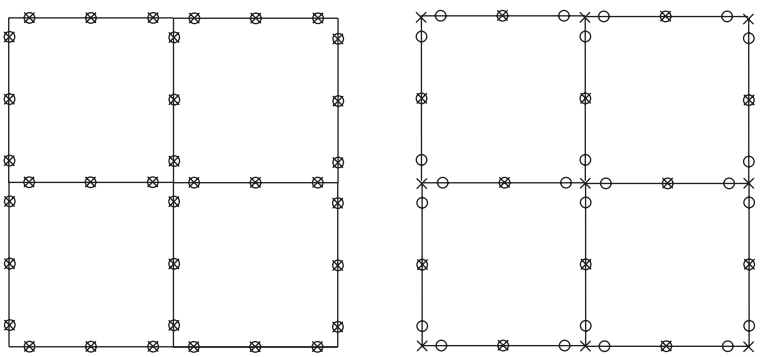

Figure 1. Discontinuous function and normal flux approximation in the classic subdomain BEM formulation (left), by authors Škerget *et al.* [14], and mixed boundary elements (right), Ramšak and Škerget [9]. The $\times$ represents field function u nodal points while the $\circ$ represents a normal flux $\partial u/\partial n$ nodal points.

Mixed elements: The idea of mixed elements represented in Ramšak and Škerget [9] is an alternative approach in BEM. The basic idea of mixed elements is to split the field function and normal flux nodal points to keep the advantages of function continuous approximation and to avoid not uniquely defined unit normal vectors of normal flux interpolation nodal points. Therefore, the function is approximated using continuous interpolation polynomials, while normal flux is interpolated using discontinuous interpolation polynomials. As a consequence, the advantages of continuous field function approximations are retained and its conservation is preserved while the normal flux values are modelled in a proper way. While using continuous elements of high orders, the application of the matching conditions of common interfaces, i.e. the matrix assembly, leads to an over-determined system of algebraic equations. Instead of using one of the several schemes that reduce the over-determined system to a closed system, see Reference [15], the over-determined system matrix is solved in a least squares sense. Using the same function interpolation nodal points for boundary and domain discretization of the convective term improves the stability of the mixed elements. The highest computed Reynolds number value has increased from 3200 to 10 000 in the case of the driven cavity flow.

The present formulation represents the continuity of the mixed elements. Both numerical schemes used the identical field function discretization, while the normal flux approximation is different.

5.2. Function approximation

In the general case, the function approximation over the boundary element could be a linear interpolation or even higher. The present work deals with the quadratic interpolation polynomials Φ^n, see left-hand side of Figure 2,

$$\Phi^n = \frac{1}{2}\begin{Bmatrix} -\xi + \xi^2 \\ 1 - \xi^2 \\ \xi + \xi^2 \end{Bmatrix} \tag{17}$$

where $\xi \in [-1, 1]$ is local co-ordinate frame and n the degree of freedom ($n = 3$).

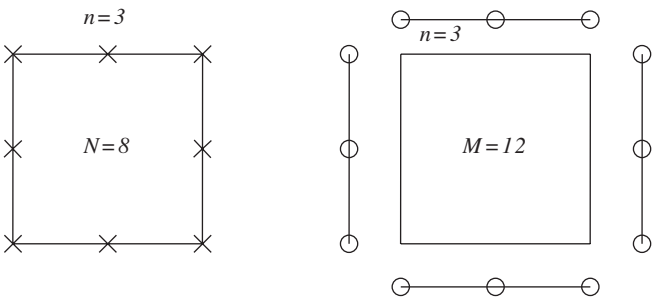

Figure 2. Continuous quadratic approximation over boundary element $n = 3$ and subdomain approximation $N = 8$ (left). Continuous normal flux approximation over boundary element ($n = 3$) using double boundary nodes (figure). The $\times$ represents function u boundary nodes and the $\circ$ represents normal flux $\partial u/\partial n$ boundary nodes.

The subdomain is surrounded by four boundary elements, see right-hand side of Figure 2. The two neighbouring boundary elements share the same boundary nodal point, let us call it the subdomain vertex, with only one (un)known value. Therefore, the function is approximated using continuous and single boundary nodal points, without need for double points.

Next, the function domain approximation has to be dealt. While using different nodal points for the function approximation over the boundary elements and domain, see Reference [14], the discretization of the boundary and domain integral of the convective term in the boundary integral equation becomes the source of numerical instabilities in convection-dominated flows. Thus the same interpolation nodal points for boundary and domain approximation are preferable in a BEM discretization. The function domain interpolation ϕ^N is biquadratic using $N=8$ boundary nodal points

$$\phi^N = \frac{1}{4}\begin{Bmatrix} -1+\xi\eta+\xi^2+\eta^2-\xi^2\eta-\xi\eta^2 \\ 2-2\eta-2\xi^2+2\xi^2\eta \\ -1-\xi\eta+\xi^2+\eta^2-\xi^2\eta+\xi\eta^2 \\ 2+2\xi-2\eta^2-2\xi\eta^2 \\ -1+\xi\eta+\xi^2+\eta^2+\xi^2\eta+\xi\eta^2 \\ 2+2\eta-2\xi^2-2\xi^2\eta \\ -1-\xi\eta+\xi^2+\eta^2+\xi^2\eta-\xi\eta^2 \\ 2-2\xi-2\eta^2+2\xi\eta^2 \end{Bmatrix} \tag{18}$$

where $\xi,\eta\in[-1,1]$ are again local co-ordinate frames.

With the interpolation polynomials defined, the boundary integrals of the fundamental solution $u^\star$ over the individual boundary element Γ_e are written as

$$h^n = \int_{\Gamma_e} \Phi^n \frac{\partial u^\star}{\partial n}\,\mathrm{d}\Gamma_e, \qquad g^n = \int_{\Gamma_e} \Phi^n u^\star\,\mathrm{d}\Gamma_e \tag{19}$$

and domain integrals over individual subdomain Ω_s

$$d_j^N = \int_{\Omega_s} \phi^N \frac{\partial u^\star}{\partial x_j}\,\mathrm{d}\Omega_s, \qquad d^N = \int_{\Omega_s} \phi^N u^\star\,\mathrm{d}\Omega_s \tag{20}$$

The integrals are dependent on geometry and material properties. In the case of the Helmholtz fundamental solution, the remaining variables are only the time increment value and diffusivity. The integrals have to be computed only once at the beginning of the computation. Details on the numerical integration are given in Reference [16].

The complete boundary integral equation over the subdomain boundary Γ_s will be written as the sum of all individual boundary integrals Γ_e surrounding the subdomain. In the vertex nodal point the contribution $\{h\}$ of both neighbouring boundary integrals Γ_e could be summed up as $\{h'\}$. The obtained boundary integral over the subdomain boundary Γ_s has $N=4(n-1)=8$

degrees of freedom. The boundary integral discretization has the following form:

$$\int_{\Gamma_s} \frac{\partial u^\star}{\partial n}\, u \,\mathrm{d}\Gamma_s = \sum_{e=1}^{4} \left\{ \int_{\Gamma_e} \Phi^n \frac{\partial u^\star}{\partial n} \,\mathrm{d}\Gamma_e \right\}^{\mathrm{T}} \{u\}^n = \sum_{e=1}^{4} \{h^n\}^{\mathrm{T}} \{u\}^n = \sum_{i=1}^{N=8} \{h'\}^{\mathrm{T}} \{u\}^N \tag{21}$$

where the last sum represents the loop over $N = 8$ interpolation boundary nodal points. The boundary integrals $\{h'\}$ and domain integrals $\{d_j\}$ and $\{d\}$ over the subdomain (Equation (20)) share the same nodal points and could be added together at each nodal point.

5.3. Normal flux approximation

The normal flux interpolation is more complicated than function interpolation in the previous section. The exterior boundary nodal points are described separately from the interior boundary nodal points in the next section where the implementation of equilibrium interface conditions are dealt with.

As mentioned before, the continuous quadratic approximation of normal flux is prescribed to the boundary element, Equation (17). Owing to the lack of uniquely defined normal vector to the boundary in the vertex nodal point, double nodal points are used. Two nodal points at the vertex of subdomain have the same geometry co-ordinates but different unit normal vector on each neighbouring boundary element and thus two (un)known values of normal flux $\partial u/\partial n$. The boundary integral of normal flux over complete boundary of subdomain Γ is computed as a sum of four individual boundary integrals Γ_e; thus, four elements with three nodal points each, are written as

$$\int_{\Gamma_s} \frac{\partial u}{\partial n} u^\star \,\mathrm{d}\Gamma_s = \sum_{e=1}^{4} \left\{ \int_{\Gamma_e} \Phi^n u^\star \,\mathrm{d}\Gamma_e \right\}^{\mathrm{T}} \left\{ \frac{\partial u}{\partial n} \right\}^n = \sum_{e=1}^{4} \{g^n\}^{\mathrm{T}} \left\{ \frac{\partial u}{\partial n} \right\}^n = \sum_{i=1}^{M=12} \{g\}^{\mathrm{T}} \left\{ \frac{\partial u}{\partial n} \right\}^M \tag{22}$$

where the last sum represents the loop over all $M = 12$ boundary normal flux nodal points.

5.4. Discrete formulation of boundary integral equation

Let us write the boundary integral equation (13) for an individual subdomain in discrete form considering discrete equations for function (21) and normal flux (22)

$$\begin{aligned} 0 = {} & -c(\xi)u(\xi) - a \sum_{i=1}^{N} \{h'\}^{\mathrm{T}} \{u\}^N + a \sum_{i=1}^{M} \{g\}^{\mathrm{T}} \left\{ \frac{\partial u}{\partial n} \right\}^M \\ & - \sum_{i=1}^{N} \{g_j\}^{\mathrm{T}} \{uv_j\}^N + \sum_{i=1}^{N} \{d_j\}^{\mathrm{T}} \{uv_j\}^N \\ & + \sum_{i=1}^{N} \{d\}^{\mathrm{T}} \{S\}^N \end{aligned} \tag{23}$$

where index i means the sum over $N = 8$ function nodal points and $M = 12$ normal flux nodal points of individual subdomain, see Figure 2. The terms containing (un)known function u could be added together as

$$0 = \sum_{i=1}^{N} \{-(c(\xi) + ah') + (-g_j + d_j)v_j\}^{\mathrm{T}} \{u\}^N + a \sum_{i=1}^{M} \{g\}^{\mathrm{T}} \left\{ \frac{\partial u}{\partial n} \right\}^M + \sum_{i=1}^{N} \{d\}^{\mathrm{T}} \{S\}^N$$

The new variable e is introduced as

$$e = -(c(\xi) + ah') + (-g_j + d_j)v_j$$

and scalar f of known value

$$f = \sum_{i=1}^{N} \{d\}^{\mathrm{T}} \{S\}^N$$

Using new variables the discretized integral equation for the subdomain is written as

$$\sum_{i=1}^{N} \{e\}^{\mathrm{T}} \{u\}^N - \sum_{i=1}^{M} \{g\}^{\mathrm{T}} \left\{ \frac{\partial u}{\partial n} \right\}^M = f \tag{24}$$

The last equation (24) represents the discrete form of the integral boundary equation (13) at the source point ξ. The complete system matrix for one subdomain is obtained by writing Equation (24) for all subdomain boundary nodal points $\xi = 1, M$

$$[E]\{u\} - [G] \left\{ \frac{\partial u}{\partial n} \right\} = \{f\} \tag{25}$$

5.5. Implementation of boundary conditions on external boundary

The matrix form of discrete boundary integral equations (25) is transformed to the system of algebraic equations by applying the boundary conditions (8), the known function value $\bar{u}$ on the boundary Γ_1 and the known normal flux value $\overline{\partial u/\partial n}$ on Γ_2 (9)

$$\begin{bmatrix} [-G]^{\Gamma_1} \\ [E]^{\Gamma_2} \end{bmatrix} \begin{Bmatrix} \left\{ \dfrac{\partial u}{\partial n} \right\}^{\Gamma_1} \\ \{u\}^{\Gamma_2} \end{Bmatrix} = \begin{Bmatrix} -[E]\{\bar{u}\}^{\Gamma_1} \\ [G] \left\{ \overline{\dfrac{\partial u}{\partial n}} \right\}^{\Gamma_2} \end{Bmatrix} + \{f\}$$

and further to the

$$[A]\{x\} = \{B\} \tag{26}$$

5.6. Implementation of interface conditions between subdomains

The middle and vertex nodal points of the boundary element are treated differently and we will consider them separately, see Figure 3.

5.6.1. Middle boundary element nodal points. Let us consider two middle points in contact on the interface between two subdomains I and II, see left-hand side of Figure 3. At subdomain I the unknown function value at nodal point I is denoted as u_I and normal flux as $\partial u/\partial n_I$. Following the same notation the u_{II} and $\partial u/\partial n_{II}$ are unknown values at subdomain II. The discretized integral boundary equation (24) can be written for subdomain I as

$$\left(\sum_{i=1}^{N-1} \{e\}^{\mathrm{T}} \{u\}^{N-1} + e_I u_I \right)_I - \left(\sum_{i=1}^{M-1} \{g\}^{\mathrm{T}} \left\{ \frac{\partial u}{\partial n} \right\}^{M-1} + g_I \frac{\partial u}{\partial n_I} \right)_I = f_I$$

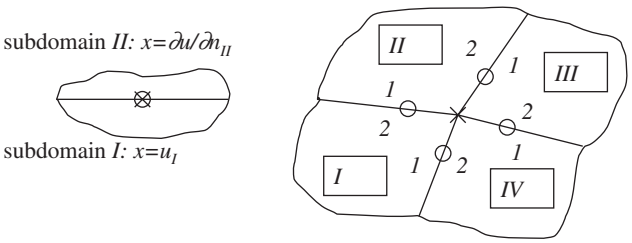

Figure 3. The middle (left) and the vertex (right) nodal point of the boundary element and implementation of the interface boundary condition. Subdomain indexes are I, II, II and IV, while the nodal points indexes are $I_1, I_2, \ldots, IV_2$. For the sake of clarity, the double vertex nodal points $\circ$ are written out of the corner, although they have the same geometrical position as the $\times$.

and for the subdomain II as

$$\left(\sum_{i=1}^{N-1}\{e\}^{\mathrm{T}}\{u\}^{N-1} + e_{II}u_{II}\right)_{II} - \left(\sum_{i=1}^{M-1}\{g\}^{\mathrm{T}}\left\{\frac{\partial u}{\partial n}\right\}^{M-1} + g_{II}\frac{\partial u}{\partial n}_{II}\right)_{II} = f_{II}$$

where the integrals with index I are computed for subdomain I and the integrals II for subdomain II. For the sake of simplicity, the sum terms on the left-hand side of equations will be omitted in the future. From the interface conditions (11) and (12), two additional equations are obtained to reduce the number of unknowns from four to two. From the compatibility interface condition $u_I = u_{II}$ let us choose the function value u_I at subdomain I as an unknown value. From the equilibrium interface condition (12) it follows that the second unknown has to be the normal flux value $\partial u/\partial n_{II}$ at subdomain II. With this unknown chosen, the above equations are rewritten as

$$e_I u_I + g_I \frac{\lambda_{II}}{\lambda_I}\frac{\partial u}{\partial n}_{II} = f_I$$

$$e_{II}u_I - g_{II}\frac{\partial u}{\partial n}_{II} = f_{II}$$

to form a closed system of two equations with two unknowns.

5.6.2. Vertex boundary element nodal points. Owing to the topological aspects of the vertex points, see right-hand side of Figure 3, an implementation of the interface boundary conditions to the vertex nodal points are not so straightforward as to the middle points. Let us write the discrete form of the integral boundary equation (24) for subdomain I at vertex nodal point $I,1$ as

$$e_I u_I - g_{I,1}\frac{\partial u}{\partial n}_{I,1} = f_I$$

and in a similar way at the double vertex point $I,2$

$$e_I u_I - g_{I,2}\frac{\partial u}{\partial n}_{I,2} = f_I \tag{27}$$

Additional pairs of equations could be written for subdomains II, III and IV, altogether eight linear independent equations. The implementation of the function compatibility interface condition is straightforward because of no doubled nodal points in the vertex. Let us set the unknown function value at the vertex nodal point as u_v and rewrite the compatibility interface condition as

$$u_I = u_{II} = u_{III} = u_{IV} = u_v$$

The equivalence of the normal fluxes has to be written at double points separately with respect to the topology, see right-hand side of Figure 3,

$$\begin{aligned}
\lambda_I \frac{\partial u}{\partial n_{I,2}} &= -\lambda_{II} \frac{\partial u}{\partial n_{II,1}} \\
\lambda_{II} \frac{\partial u}{\partial n_{II,2}} &= -\lambda_{III} \frac{\partial u}{\partial n_{III,1}} \\
\lambda_{III} \frac{\partial u}{\partial n_{III,2}} &= -\lambda_{IV} \frac{\partial u}{\partial n_{IV,1}} \\
\lambda_{IV} \frac{\partial u}{\partial n_{IV,2}} &= -\lambda_I \frac{\partial u}{\partial n_{I,1}}
\end{aligned} \tag{28}$$

which in the general case could not be further equalized. To summarize, the number of unknowns is five, one function value u_v and four normal fluxes values, and eight available equations, which leads to the overdetermined system matrix.

The number of unknowns can be reduced further using the next approach. At arbitrary vertex nodal point i with uniquely defined normal unit vector (n_x, n_y) the normal flux is defined with

$$\left(\frac{\partial u}{\partial n}\right)_i = \left(n_x \frac{\partial u}{\partial x} + n_y \frac{\partial u}{\partial y}\right)_i$$

where $(\partial u/\partial x, \partial u/\partial y)$ is gradient. The set of interface equations (28) accounting for the linear independence of the co-ordinate frame could be rewritten for both axes as

$$\begin{aligned}
\left(\lambda n_x \frac{\partial u}{\partial x}\right)_i &= -\left(\lambda n_x \frac{\partial u}{\partial x}\right)_j \\
\left(\lambda n_y \frac{\partial u}{\partial y}\right)_i &= -\left(\lambda n_y \frac{\partial u}{\partial y}\right)_j
\end{aligned}$$

where indexes i, j means arbitrary opposite vertex nodal points from Equations (28). Using this approach gradient continuity is required at the subdomain vertex points instead of normal flux equalization Equation (28).

Let us assign the index v for the unknown vertex gradient value and rewrite the discrete form of integral equation (27) at arbitrary vertex nodal point $i = 1, 2, \ldots, 8$ with respect to the

opposite nodal point using index j

$$e_i u_v - \frac{\lambda_j}{\lambda_i} g_i \left(n_{x,i} \frac{\partial u}{\partial x_v} + n_{y,i} \frac{\partial u}{\partial y_v} \right) = f_i \tag{29}$$

The number of unknowns in the interior subdomain vertex has been reduced from five to three, i.e. vertex function u_v and gradient $(\partial u/\partial x)_v, (\partial u/\partial y)_v$, while the number of equations remains eight. As already mentioned the resulting system of equations has more equations than unknowns, hence it should be solved in the linear least squares sense, which will be explained in detail in the next section.

The discrete form of integral boundary equation (24) for the internal subdomain accounting for the interface boundary conditions could be written as

$$\sum_{i=1}^{N} \{e\}^{\mathrm{T}} \{u\}^N - \sum_{i=1}^{M_m} \{g\}^{\mathrm{T}} \left\{ \frac{\lambda_j}{\lambda_i} \frac{\partial u}{\partial n} \right\}^{M_m} - \sum_{i=1}^{M_v} \{g\}^{\mathrm{T}} \left\{ \frac{\lambda_j}{\lambda_i} n_x \frac{\partial u}{\partial x} + \frac{\lambda_j}{\lambda_i} n_y \frac{\partial u}{\partial y} \right\}^{M_v} = f \tag{30}$$

where the sum $i = 1, M$ over flux nodal points is splitted to the middle boundary element nodal points $M_m = 4$ and vertex nodal points $M_v = 8$.

6. SOLVING THE OVERDETERMINED MATRIX

One of the simplest approaches for solving an over-determined set of algebraic equations

$$A_{M \times N} \{x\}_N = \{b\}_M$$

where $M > N$, is finding a least squares solution using QR factorization of A for dense matrices from the lapack library (LLS) [17]. Since in our case the system matrix A is sparse and block banded, the iterative linear least squares solver of Paige and Saunders [10] (PAIGE) is much faster at same solution accuracy. The method is based on the bidiagonalization procedure of Golub and Kahan, for details see Reference [10]. It is analytically equivalent to the standard method of conjugate gradients.

6.1. Preconditioning

To accelerate the convergence a diagonal preconditioning is applied (PRECOND). For square systems $Ax = b$, it means finding non-singular matrices C_1 and C_2 that make

$$B = C_1 A C_2$$

closer to identity matrix or nearer to a matrix with clustered singular values. $Ax = b$ is equivalent to

$$C_1 A C_2 y = C_1 b$$

so

$$By = C_1 b$$

is solved by the iterative method and recovers

$$x = C_2 y$$

Table I. Comparison of LLS QR factorization for dense matrices and iterative LLS with or without preconditioning. N_{sub} means number of subdomains, M and N number of equations and unknowns, respectively, *sparse* coefficient is calculated as number of non-zero elements in the matrix divided with $M * N$, N_{it} is the number of local solver iterations.

Matrix parameters				LLS	PAIGE		PRECOND	
N_{sub}	M	N	Sparse	CPU	N_{it}	CPU	N_{it}	CPU
$16 = 4 \times 4$	216	111	0.124	0.17	129	0.26	190	0.38
$64 = 8 \times 8$	944	447	0.033	23.0	191	1.6	153	1.2
$256 = 16 \times 16$	2184	1007	0.015	1435.0	313	10.0	293	9.6
$1024 = 32 \times 32$	16128	9215	0.002	—	275	37.0	254	34.0

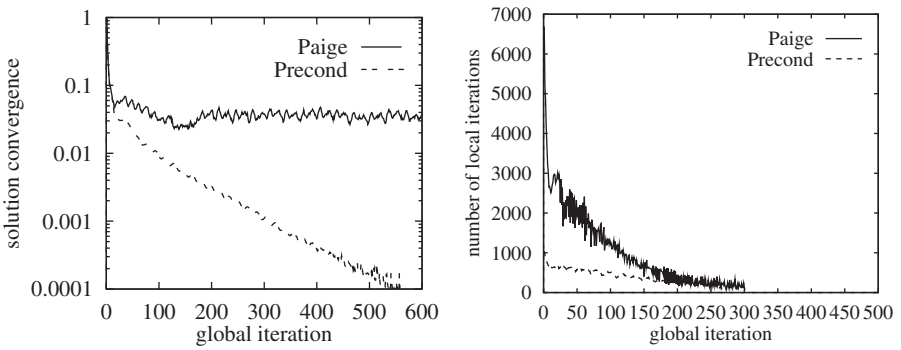

Figure 4. Non-convergence of the problem without using the preconditioner (left). Decreasing number of local solver iterations using the preconditioner.

For overdetermined systems the 'left preconditioner' $C_1 = I$ is an identity orthogonal matrix. Ideally C_2 would approximate R^{-1}, where R comes from the QR factorization, which of course is not affordable to compute. In general an approximate Cholesky factorization is needed to set the right preconditioner as

$$C_2 = PR^{-1}$$

where P is some permutation and R is an upper triangular matrix. One of the simplest approaches is set $P = I$ defining R to be the diagonal as the square-root of the diagonals of $A^{\mathrm{T}}A$

$$\mathrm{diag}(R) = \sqrt{\mathrm{diag}(A^{\mathrm{T}}A)}$$

The comparison of the presented solvers are made for solving a diffusion-convective problem known as entry flow, see Table I. From the table it is evident that the iterative solver is approximately 100 times faster at higher grid densities. Application of preconditioner results on approximately 10% speedup in this case. In the case of a driven flow in the square closed cavity the number of solver local iterations significantly decreases, see right-hand side of Figure 4, the resulting speedup is approximately 2. In some strong non-linear cases where

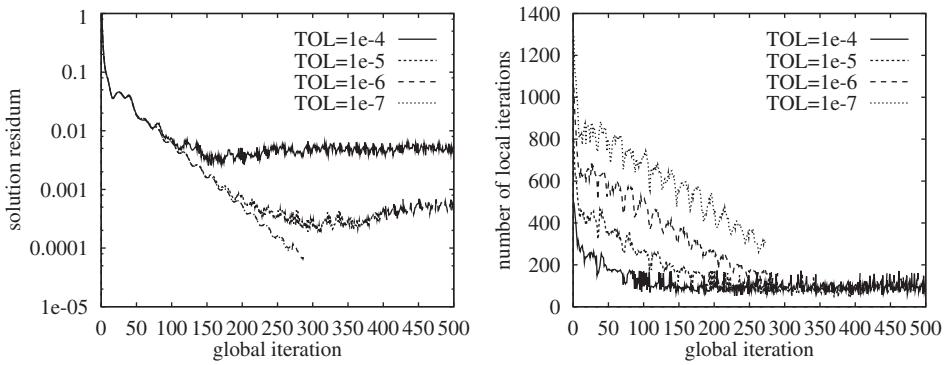

Figure 5. The convergence and local number of solver iterations depending on solver stopping criteria TOL.

Table II. The necessary CPU, the number of global iterations NIT and the solution residue ε dependency on the solver stopping criteria TOL.

TOL	CPU [s]	NIT	ε
10^{-4}	—	—	5.E-2
10^{-5}	—	—	5.E-3
10^{-6}	220	290	1.E-4
10^{-7}	280	273	1.E-4

the convection is dominant over diffusion processes, the use of the preconditioner can be of crucial matter to achieve the solution convergence, see left-hand side of Figure 4.

The solver stopping criteria TOL is another significant parameter crucial for convergence. The TOL is a required estimation of the relative error for the solver solution vector. In Figure 5, the convergence (left) and the number of solver local iterations (right) is plotted against the global iterations. The TOL values of 10^{-4} and 10^{-5} are too high, causing the solution convergence only to certain values, linearly dependent on TOL. With the tightening of the TOL, i.e. smaller values, the necessary number of solver local iterations is increasing, thus increasing solution accuracy and the CPU, see Table II. The optimal selection of TOL may be a variable during the convergence, i.e. at the final stage of global iterations TOL should be tighter.

The solution convergence criteria ε is defined as the normalized quadratic norm of the current solution vector u^l and the solution from the previous iteration u^{l-1}

$$\varepsilon = \frac{\|u^l - u^{l-1}\|}{\|u^l\|}$$

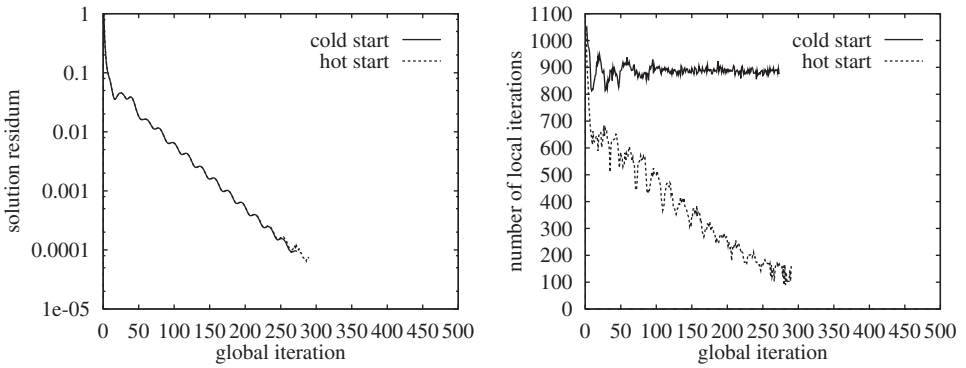

Figure 6. The solution residum (left) and number of solver local iterations (right) applying the hot start.

6.2. *Hot start*

In order to speed up the solution another idea named *hot start* connected is applied. The basic idea of the hot start is using the solution vector x (Equation (26)) from previous global iteration as an initial guess x_0 in present iteration. The hot start is applied at three stages:

1. computing the new right-hand side vector of known values b_0 at Equation (26) from the initial solution vector x_0 as

$$b_0 = b - A \cdot x_0$$

2. solving the *change* of the solution dx from the system of algebraic equations

$$A \cdot \mathrm{d}x = b_0$$

3. recovering the true solution x from

$$x = x0 + \mathrm{d}x$$

The test case is driven cavity flow. From the left-hand side of Figure 6, no significant difference of vorticity convergence is evident between the hot and cold start. But the right-hand side of Figure shows a significantly decreased number of local solver iterations resulting in speedup factor up to 2. At the final stages of convergence the solution changes very little, thus the initial solution vector $x0$ is already a good estimate to the final solution x. That makes the dx go to zero and reduce the necessary local solver iterations during the convergence, while at the cold start the necessary number of local iterations remains constant. Note, the results in Figures 4 and 5 already included the hot start, thus explaining the decreasing number of local iterations during global convergence.

7. COMPUTATIONAL ALGORITHM

For the sake of clarity let us assume known velocities on the complete boundary Γ. The outlet boundary conditions are explained in the numerical example and require only slight

modification. Before we proceed, we should note once more that boundary values mean nodes on the external boundary and domain values are all the rest nodes on the interfaces between the subdomains. The solution algorithm for stream function-vorticity formulation can be written as:

1. Choose initial values of domain vorticity ω_Ω^{l-1}
2. Begin new time step
3. *Solve the stream function equation* with known boundary conditions for the boundary stream function $\bar{\psi}_\Gamma$, its normal flux $\overline{\partial\psi/\partial n}_\Gamma$ and known domain vorticities ω_Ω^{l-1} from previous time step

$$\frac{\partial^2}{\partial x_i \partial x_i}(\bar{\psi}_\Gamma + \psi_\Omega^l) = -(\omega_\Gamma^l + \omega_\Omega^{l-1}) \tag{31}$$

resulting in new domain values of stream function ψ_Ω^l, its derivatives $\partial\psi/\partial x_{i\Omega}^l$ and boundary vorticities ω_Γ^l.

3.1 If necessary, under relaxation of ψ_Ω^l and ω_Γ^l use URFψ_Ω and URFω_Γ, respectively.

3.2 Set new domain velocities

$$v_j^l = \left(\frac{\partial\Psi}{\partial y}, -\frac{\partial\Psi}{\partial x}\right)^l \tag{32}$$

implicitly as the result of Equation (31).

4. *Solve the vorticity equation* using the new vorticity boundary conditions $\bar{\omega}_\Gamma^l$ and new known velocities v_j^l

$$\frac{\omega^l - \omega^{l-1}}{\Delta t} + v_j^l \frac{\partial\omega^l}{\partial x_j} = \nu \frac{\partial^2}{\partial x_i \partial x_i}(\bar{\omega}_\Gamma^l + \omega_\Omega^l) \tag{33}$$

resulting in the new domain values of vorticity ω_Ω^l, its derivatives $\partial\omega/\partial x_{i\Omega}^l$ and boundary vorticity normal fluxes $\partial\omega/\partial n_\Gamma^l$.

4.1 If necessary, under relaxation of ω_Ω^l use URFω_Ω.

5. If convergence is not achieved go to 2.

8. NUMERICAL EXAMPLES

8.1. Scalar diffusion in 'L' shape

The laplace equation without sources is solved, physically representing a heat conduction in a solid or potential flow. The geometry of the problem and the boundary conditions are presented in Figure 7.

The minimal possible grid using three subdomains is applied, see right-hand side of Figure 7. The results are compared with the hypersingular boundary element method (HBEM) by Rek [16] using the same grid. The potential function u is compared at the vertical profile $x = 5.0$ in Table III. It can be clearly seen that our result fits the best to the analytical solution. The main problem represents the concave corner at $x = 5.0$ and $y = 5.0$ where the flux solution is singular. The present study has the excellent agreement of six decimal places with the analytical solution in this corner.

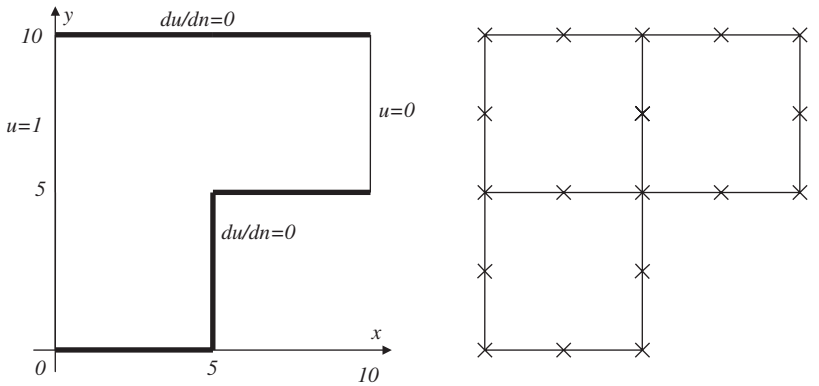

Figure 7. The geometry and boundary conditions for the scalar diffusion in 'L' shape (left). The minimal possible grid for this case (right).

Table III. Comparison of potential value u at the vertical profile $x = 5.0$ for the scalar diffusion in the 'L' shape.

y	Anal.	HBEM	$\Delta\%$	Present	$\Delta\%$
10.0	0.547	0.553	1.1	0.553	1.1
7.5	0.566	0.523	−7.6	0.572	1.1
5.0	0.667	0.731	9.5	0.667	0.0
2.5	0.868	0.881	1.5	0.857	−1.3
0.0	0.906	0.910	0.4	0.893	−1.4

8.2. *Driven cavity flow*

The square-driven cavity flow has been one of the most attractive test problems for the CFD community in spite of velocity discontinuity at the cavity corners adjacent to the moving wall. The results are compared using Ghia *et al.* [11] work as a benchmark solution to this problem.

The problem considered here is that of a square cavity $x, y = [0, 1]$ totally filled with an incompressible viscous fluid and a moving top wall at a constant velocity.

The test case was solved using two commercial codes: the Fluid Dynamics International (FIDAP) using the Finite Element Method and the TASCflow (Advanced Scientific Computing) [18] using the Control Volume Method. The comparison of the three different methods on an equal grid using the same number of degrees of freedom is performed. The FEM results at $Re = 100$ are computed without upwinding, while at $Re = 1000$ an upwinding factor of 1.25 was necessary to obtain a good solution. The FVM results are computed as a transient case using $\Delta t = 10^{-3}$. The convergence criteria being 10^{-6} (see Figure 8).

Convergence: The results at $Re = 100$ are computed with time step value $\Delta t = 10^{16}$ and approximately 150 iterations to the default convergence criteria of $\varepsilon = 10^{-4}$, see Figure 9. The boundary vorticity under relaxation value URFω_Γ and domain vorticity URFω_Ω have to be applied to obtain convergence. The optimal values in this case were found to be 0.8

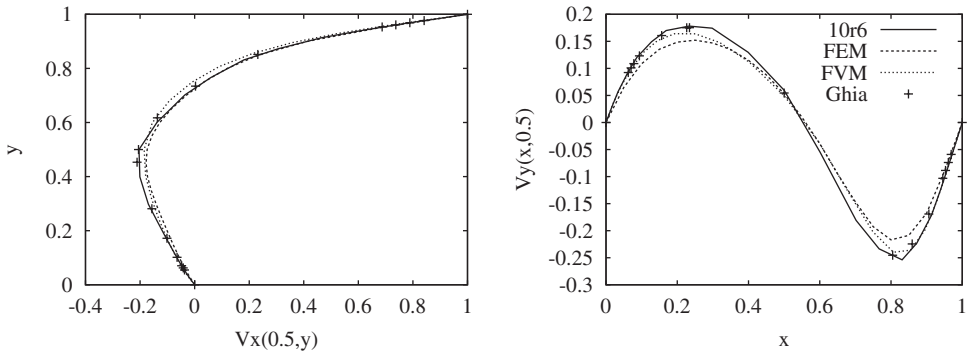

Figure 8. Driven cavity flow at $Re = 100$. Velocity profiles comparison on grid density 10×6.

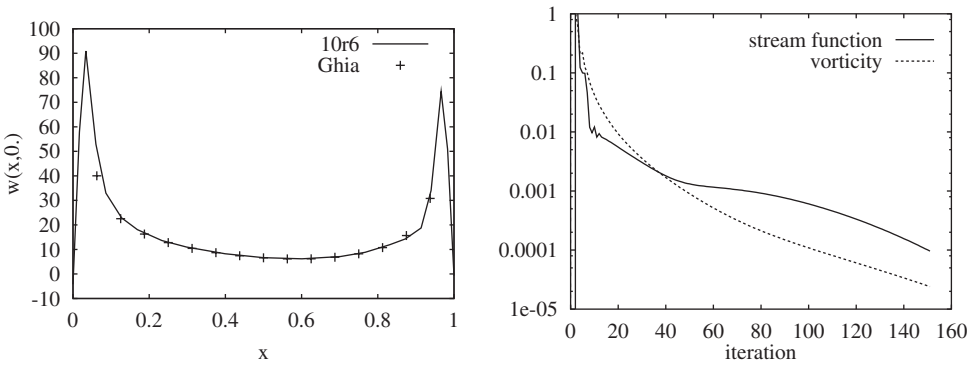

Figure 9. Driven cavity flow at $Re = 100$. Vorticity distribution on driven wall (left) and convergence (right).

and 0.1, respectively. The Poisson equation for stream function is numerically easier to solve and results do not have to be under relaxed URF$\omega_\Omega = 1$. With higher Reynolds number values convection dominates diffusivity and the vorticity transport equation becomes more non-linear, lower URF values have to be applied or shorter time step values. Using short time step values decreases fundamental solution integrals accuracy, Equations (19) and (20). We found the optimal value of $\Delta t = 1.0$ at the beginning stage of computation and then restart with higher value of $\Delta t = 10^3$ to the accurate steady solution, if necessary. The URF values remain the same as mentioned, up to $Re = 10\,000$. At the extreme value of $Re = 25\,000$ the time step value has been $\Delta t = 0.1$, URF$\psi = 0.1$, URF$\omega_\Gamma = 0.1$ and URF$\omega_\Omega = 0.01$. The number of iterations was approximately 1000, see Figure 10.

A stability of high Reynolds flow for $Re > 10\,000$: Grigoriev and Dargush [6] stated the lack of a steady-state solution for two-dimensional flow in a cavity at $Re > 5000$. Using BEM the solution divergate for $Re > 7500$ in their case. Ghia *et al.* [11] published results up to the $Re = 10000$ (see Figures 11–18). Shen [19] found that the flow converges to a stationary state for Re up to 10 000 using spectral methods. The flow loses periodicity for $Re > 16\,000$.

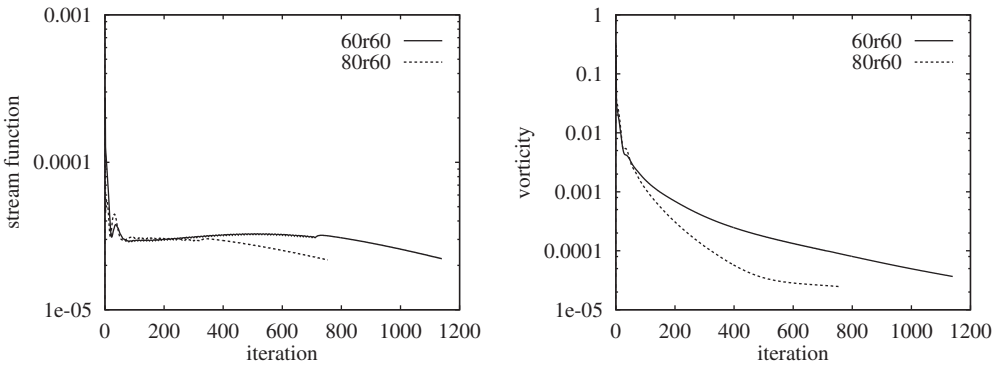

Figure 10. Driven cavity flow at $Re = 25\,000$. Stream function (left) and vorticity (right) convergence computed as restart from results at $Re = 10\,000$.

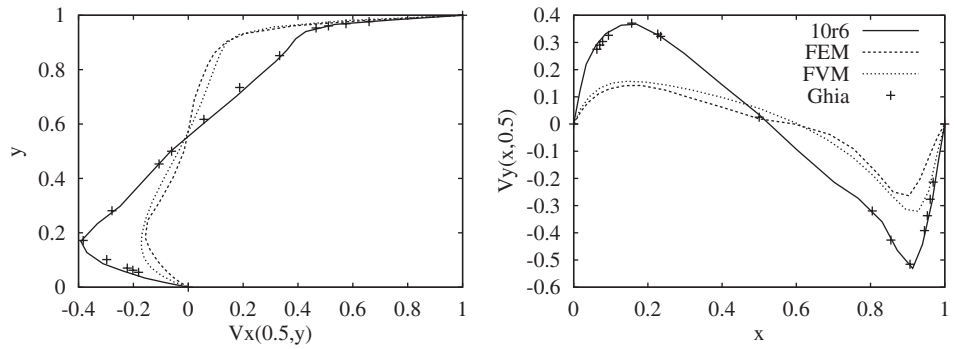

Figure 11. Driven cavity flow at $Re = 1000$. Velocity profiles comparison on grid density 10×6.

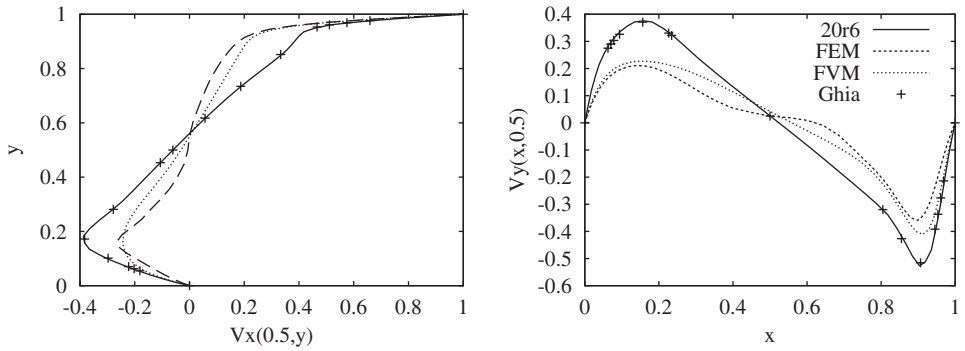

Figure 12. Driven cavity flow at $Re = 1000$. Velocity profiles comparison on grid density 20×6.

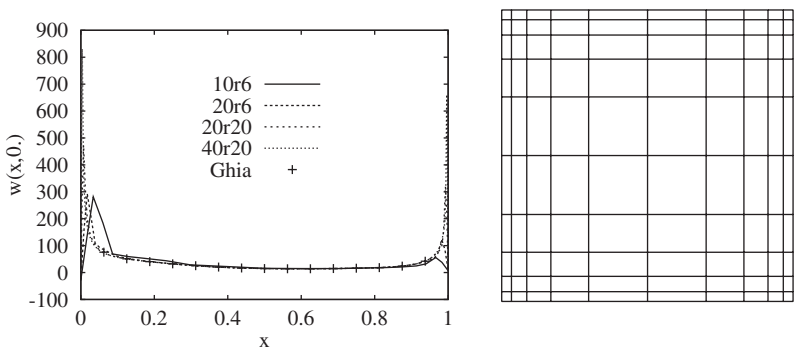

Figure 13. Driven cavity flow at $Re = 1000$. Vorticity distribution at driven wall (left) and grid density 10×6.

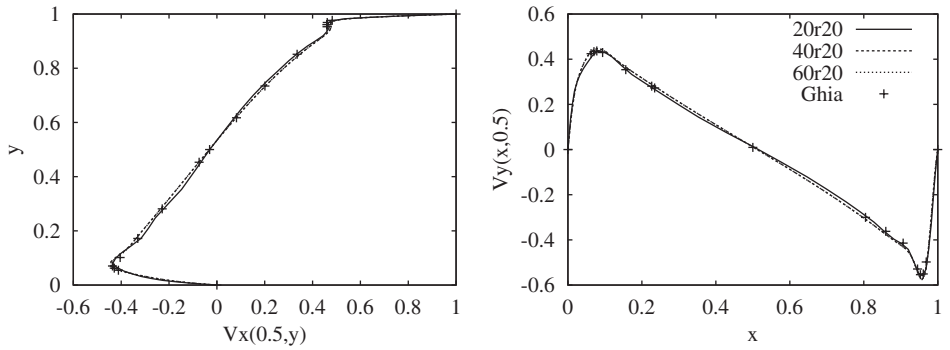

Figure 14. Driven cavity flow at $Re = 5000$. Velocity profiles comparison.

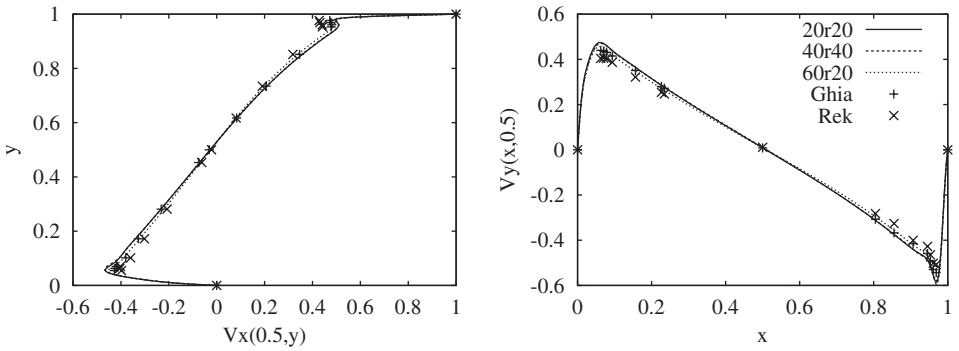

Figure 15. Driven cavity flow at $Re = 10\,000$. Velocity profiles comparison.

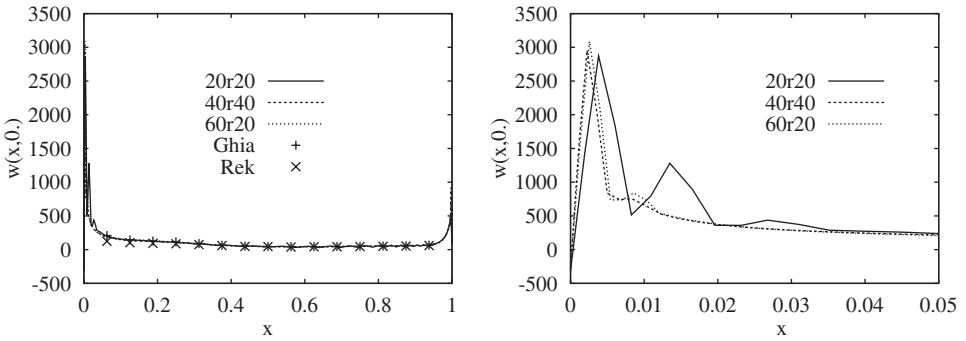

Figure 16. Driven cavity flow at $Re = 10\,000$. Vorticity distribution at driven wall.

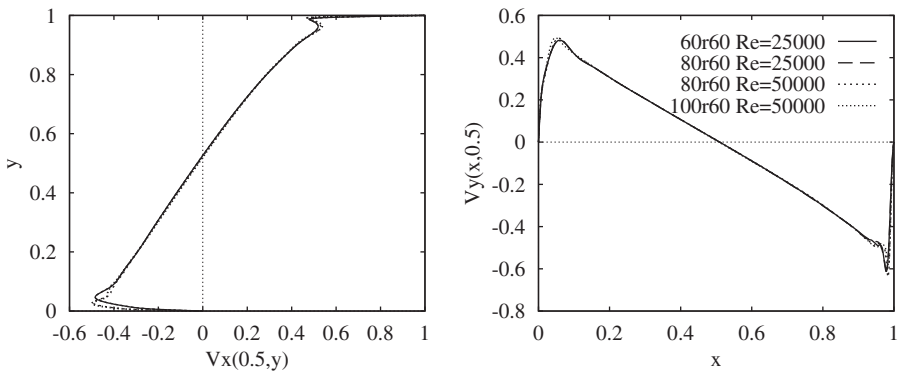

Figure 17. Driven cavity flow at $Re = 25\,000$ and $Re = 50\,000$. Velocity profiles comparison.

A steady state solution at $Re = 25\,000$ has been provided by Bhardwaj and Verma [24] using DNS. Discussion about stability is beyond our scope. The presented BEM numerical algorithm is stable even at $Re = 50\,000$ producing a steady-state solution using transient computation. One should note the existence of new vortex at the upper left corner at $Re = 25\,000$, see Figure 19. It becomes stronger at $Re = 50\,000$, see Figure 20. At the lower right corner the secondary vortex has the tendency to break on two new vortex at slightly higher Re value, which supposed to be investigated in future work.

Fundamental solution: It is well known that using velocity dependent fundamental solution increases the stability of BEM. Our experience does not shows significant improvement using it, certainly not worth the increasing CPU demands computing new integrals for each iteration. We cannot prove that, but the reason is probably the linear least squares solver averaging the influence of eight BEM equations at internal subdomain vertex nodal point, thus neglecting the non-symmetrical velocity contribution by fundamental solution in the mean flow direction.

An *accuracy* of developed BEM is most evident on the extreme coarse grid densities, starting with the grid of 10×10 subdomains with the symmetrically non-equidistant, the two-way bias grid seed with ratio longest/shortest element of value 6 (further assigned as grid

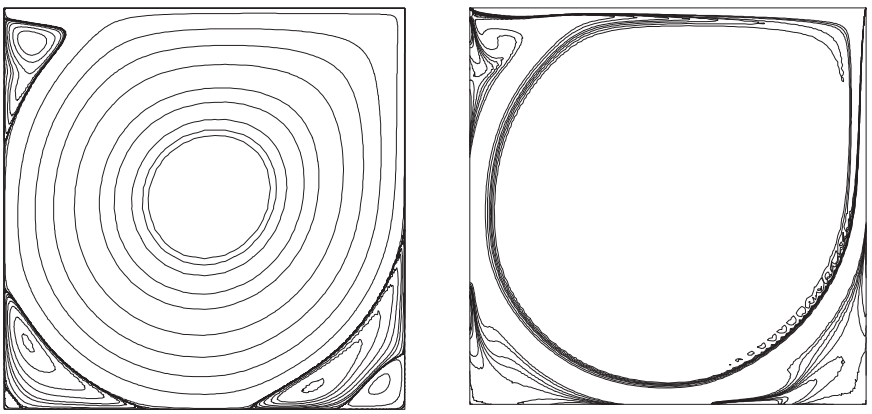

Figure 18. Driven cavity flow at $Re = 10\,000$. Stream line pattern (left) and vorticity contours (right) at finest grid 60×20. Stream function level values are: −1e-10, −1e-7, −1e-5, −1e-4, −0.01, −0.03, −0.05, −0.07, −0.09, −0.1, −0.11, −0.115, −0.1175, 1e-8, 1e-7, 1e-6, 1e-5, 5e-5, 1e-4, 2.5e-4, 5e-4, 1e-3, 1.5e-3, 3e-3 (as in Reference [4]). Vorticity levels are: −3, −2, −1, −0.5, 0, 0.5, 1, 2, 3, 4, 5 (as in Reference [11]).

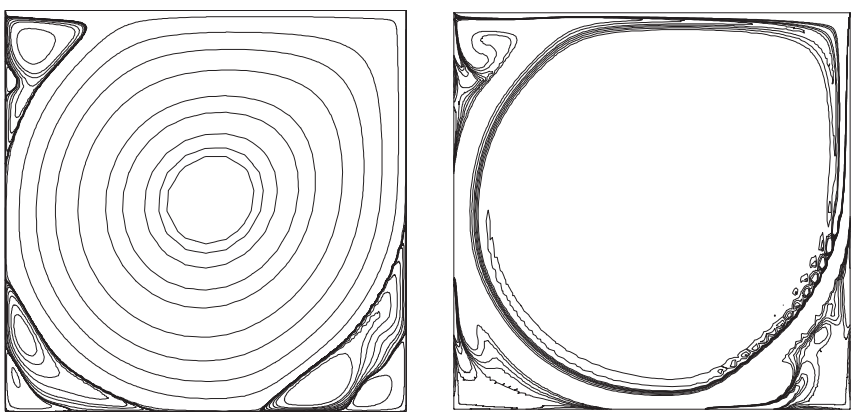

Figure 19. Driven cavity flow at $Re = 25\,000$. Stream line pattern (left) and vorticity contours (right) at finest grid 80×60. Contour level values are as in Figure 18.

density '10×6', see Figure 13) where the velocity profiles agree excellently with the reference solution for $Re = 100$ and 1000, see Figures 8–12. FEM and FVM results fail to account for viscous forces at such low grid densities with approximately equal degrees of freedom resulting in much lower velocity peaks. At higher grid densities the difference between results is not so obvious. The comparison with BEM results provided by Rek [1] shows slightly better agreement of our results on the same grid density 40×10, see Figures 15 and 16.

Nodalization analysis or grid sensitivity on centreline velocity profiles shows relative independence and good convergence to reference solutions using higher grid densities, see Figure 11 with grid 10×6 versus Figure 12 with 20×6. At higher *Re* values, the velocity profiles

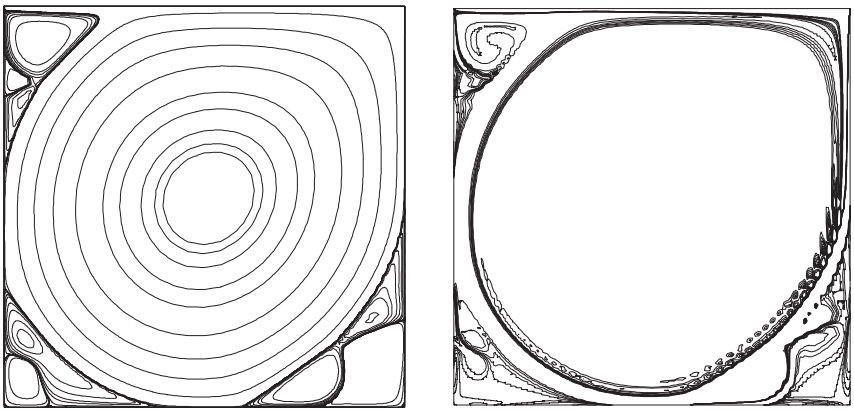

Figure 20. Driven cavity flow at $Re = 50\,000$. Stream line pattern (left) and vorticity contours (right) at finest grid 100×60. Contour level values are as in Figure 18.

agrees very well with reference solution at all computed grid densities. Velocities representing the stream function derivative equation (32) and computed implicitly from the BEM equations are also free of numerical dispersion. Boundary vorticity computation (see Equation (31)) is much more sensitive to the grid density and numerical wiggles, especially near the singular corners, due to large gradient of nearly step change of tangential velocity boundary condition, see Figure 13. In the case of driven cavity this seems the most critical part of computation. While the vorticity peaks near the singular corners are higher with higher grid density, the velocity profiles along centrelines are much less sensitive. They are far away from singular corners. At $Re = 10\,000$ the grid 20×20 is obvious to coarse producing weak numerical wiggles, see Figure 16. We should note, the maximal vorticity value is an approximately equal value at this figure. It appears that the developed BEM numerical algorithm converges despite weak numerical dispersion. It is natural that if wiggles became too strong, the scheme does not converge (see Tables IV–VI).

Unfortunately *computer demands* should be discussed. The grid densities up to 1600 subdomains required neglectful amount of memory, while computing a driven cavity case represent an overnight job using 400 MHz workstation. Higher grid densities using 3600 or 10 000 eight node quadratic subdomains require 125 or 300 MB of memory, while CPU is 2 or 10 min per one time step, respectively.

8.3. Backward-facing step flow

There are two main aims of this standard numerical example. The first is to show the regularity of the developed numerical scheme and stream function-vorticity formulation using the outflow boundary conditions and thus fulfilment of mass conservation. The second is to present accuracy on a wide range of grid densities.

The geometry and boundary conditions are presented in Figure 21. The Reynolds number based on the inlet height 0.5 and average inlet velocity 1.0 is 100, 800 and 1000.

Grids: The computations were carried out on three non-equidistant grids. The first extremely coarse grid is 30×2–6×1 using 30 subdomains using one way bias value of 2 in x-direction

Table IV. Results for v_x velocity along the vertical line through the geometric centre of cavity.

Re	100	1000		5000	10 000		25 000	50 000
grid	10×6	10×6	20×6	40×20	20×20	60×20	80×20	100×60
$N_{\text{d.o.f.}}$	341	341	1281	4961	1281	11041	19521	30401
0.0000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.0547	−0.03770	−0.22359	−0.18586	−0.40642	−0.46567	−0.45332	−0.47638	−0.42910
0.0625	−0.04257	−0.24641	−0.20746	−0.42788	−0.46192	−0.44596	−0.46031	−0.42789
0.0703	−0.04738	−0.26835	−0.22787	−0.44178	−0.45036	−0.43243	−0.44175	−0.42354
0.1016	−0.06556	−0.33684	−0.30423	−0.42346	−0.40592	−0.38763	−0.38774	−0.39167
0.1719	−0.10408	−0.38932	−0.38717	−0.33820	−0.33958	−0.32167	−0.32205	−0.32284
0.2813	−0.16098	−0.27189	−0.27942	−0.23481	−0.23263	−0.21960	−0.22348	−0.21924
0.4531	−0.20599	−0.10704	−0.10613	−0.07521	−0.07152	−0.06714	−0.06879	−0.06271
0.5000	−0.20402	−0.06397	−0.06015	−0.03171	−0.02780	−0.02594	−0.02515	−0.01917
0.6172	−0.12363	0.07390	0.05790	0.08019	0.08533	0.08007	0.08773	0.09271
0.7344	0.00943	0.21086	0.18885	0.20268	0.20936	0.19720	0.21036	0.21275
0.8516	0.23827	0.34108	0.33664	0.34723	0.35803	0.33869	0.35428	0.35193
0.9531	0.69024	0.47269	0.47679	0.47753	0.50667	0.48839	0.51378	0.52762
0.9609	0.73854	0.50737	0.51620	0.47681	0.50873	0.49284	0.51895	0.53691
0.9688	0.78960	0.56379	0.58574	0.47776	0.50370	0.49100	0.51662	0.53523
0.9766	0.84221	0.65981	0.66866	0.49785	0.49438	0.48502	0.50468	0.52019
1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

Table V. Results for v_y velocity along the horizontal line through the geometric centre of cavity.

Re	100	1000		5000	10 000		25 000	50 000
grid	10×6	10×6	20×6	40×20	20×20	60×20	80×20	100×60
$N_{\text{d.o.f.}}$	341	341	1281	4961	1281	11041	19521	30401
0.0000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.0625	0.09376	0.29472	0.28158	0.43155	0.47447	0.45656	0.48259	0.47509
0.1250	0.10279	0.30938	0.29674	0.44222	0.47044	0.45153	0.47583	0.46038
0.1875	0.11142	0.32162	0.31005	0.44590	0.46113	0.44244	0.46308	0.44786
0.2500	0.12597	0.33954	0.33388	0.44103	0.43934	0.42036	0.43056	0.42715
0.3125	0.16457	0.36891	0.37567	0.36520	0.36502	0.34710	0.35310	0.35492
0.3750	0.17791	0.32826	0.33357	0.28798	0.28531	0.27134	0.27840	0.27932
0.4375	0.17824	0.32187	0.32529	0.27964	0.27681	0.26323	0.27032	0.27129
0.5000	0.05965	0.03410	0.02578	0.01149	0.00851	0.00891	0.00941	0.00759
0.5625	−0.24950	−0.32076	−0.32039	−0.30880	−0.31214	−0.29594	−0.30715	−0.30537
0.6250	−0.23502	−0.43579	−0.42728	−0.37560	−0.38049	−0.36152	−0.37174	−0.37116
0.6875	−0.18208	−0.52176	−0.52629	−0.43089	−0.44042	−0.42061	−0.43174	−0.43350
0.7500	−0.11229	−0.41600	−0.40896	−0.54154	−0.48660	−0.47241	−0.47170	−0.48683
0.8125	−0.09681	−0.37191	−0.35591	−0.56982	−0.51406	−0.50332	−0.47145	−0.49470
0.8750	−0.08098	−0.32155	−0.29764	−0.56851	−0.55569	−0.54797	−0.48489	−0.48545
0.9375	−0.06459	−0.25985	−0.23128	−0.51699	−0.58331	−0.57817	−0.52559	−0.48270
1.0000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

and only 6 equidistant subdomains in y-direction. The second grid is 60×2–10×1 and the finest is 120×2–20×1.

Convergence: Results at all *Re* number values are obtained using $\Delta t = 1.0$. The stream function under relaxation factor is $\text{URF}\psi = 1.0$, since the ψ governing equation is of Poisson type

Table VI. Results for vorticity along moving boundary.

Re	100	1000		5000	10 000		25 000	50 000
grid	10×6	10×6	20×6	40×20	20×20	60×20	80×20	100×60
$N_{\text{d.o.f.}}$	341	341	1281	4961	1281	11041	19521	30401
0.0625	50.3105	174.436	84.2831	140.503	201.538	184.542	290.400	389.968
0.1250	23.3507	60.0627	52.6517	102.828	145.649	142.261	214.590	298.949
0.1875	16.5955	51.1416	40.9195	91.2248	126.221	134.218	186.676	249.321
0.2500	12.7632	40.3458	32.3196	76.4351	104.896	109.940	181.668	251.847
0.3125	10.5052	27.7116	25.3117	58.3082	85.1456	83.2074	147.548	219.550
0.3750	8.78134	23.7769	20.1473	44.4863	67.5477	62.9283	108.388	160.270
0.4375	7.49459	19.7529	16.6858	36.1961	54.4392	51.2952	85.1226	123.464
0.5000	6.64493	15.6397	14.8310	32.0036	38.1508	45.5952	73.9458	108.304
0.5625	6.20547	15.0865	13.9692	30.5201	43.1701	43.3595	69.9852	100.784
0.6250	6.28753	14.7902	14.0934	30.9304	45.9512	43.7597	70.8356	102.258
0.6875	6.89110	14.7507	14.8087	32.6810	44.1457	46.0866	75.1534	108.791
0.7500	8.24257	16.2630	16.0602	35.3294	46.3567	49.7756	81.9522	122.041
0.8125	10.8698	18.0160	18.3451	38.3662	54.4209	54.1875	89.6403	129.883
0.8750	14.5068	20.9193	23.8935	41.9363	46.3055	58.2146	94.4127	144.097
0.9375	31.5660	30.9000	41.3518	58.9405	75.4742	69.4561	95.9171	116.848

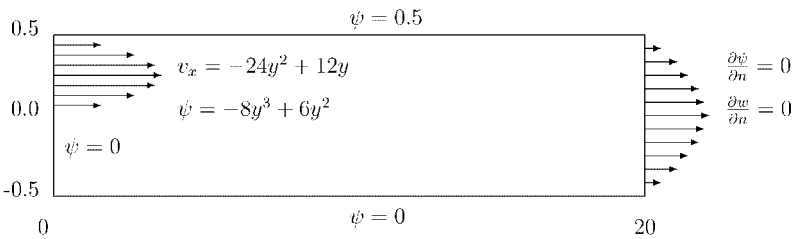

Figure 21. Geometry and boundary conditions for backward-facing step flow.

and numerically easier to solve than convection–diffusion type vorticity kinetics where domain vorticity under relaxation is necessarily URF$w_\Omega = 0.1$ and boundary vorticity URF$w_\Gamma = 0.8$ at this Reynolds number value.

Results at $Re = 100$: Stream function and vorticity convergence at $Re = 100$ during the iterative procedure is shown in Figure 22 for all grid densities (see Table VII). As expected the stream function converges faster than under relaxed vorticity. Practically the same number of necessary iterations for all grid densities is encouraging. Figure 23 shows the comparison of vorticity distribution along lower and upper wall between computational grids. Profiles are practically the same for all grid densities. The coarser grid 30×2–6×1 produced weak numerical oscillations at the outlet boundary, which vanished using the highest grid density. The mass balance is improved from 0.5% at the coarser grid to 0.01% at the finest grid density. Using the higher *Re* number value the mass balance is practically unchanged.

Results at $Re = 800$ are obtained using the restart from $Re = 100$. The necessary number of additional iterations is approximately 400 for all grid densities. From the wall vorticity distribution in Figure 24 it is obvious that the grid 30×2–6×1 is too coarse, while the results

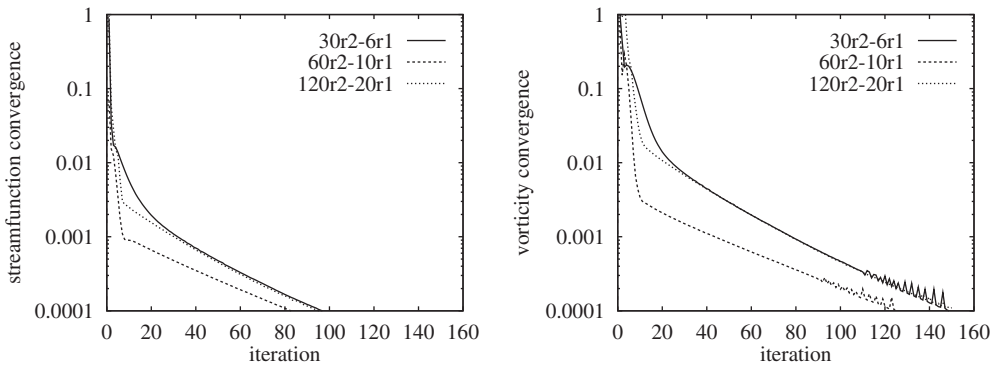

Figure 22. Backward-facing step flow at $Re = 100$. Stream function (left) and vorticity (right) convergence for all grid densities.

Table VII. Backward-facing step flow $Re = 100$. $X1$ length of lower wall eddy, m mass balance and N_{it} number of iterations.

Grid	N_{sub}	$X1$	m (%)	N_{it}
30×2–6×1	180	1.75	0.45	149
60×2–10×1	600	1.64	0.10	125
120×2–20×1	2400	1.61	0.01	152

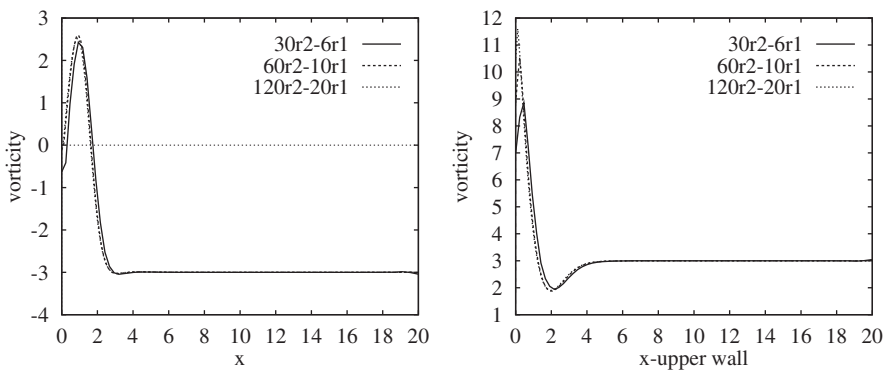

Figure 23. Backward-facing step flow at $Re = 100$. Distribution of vorticity along lower (left) and upper (right) wall for all grid densities.

used grids 60×2–10×1 and 120×2–20×1 produced approximately equal results. The comparison with FEM results by Gartling [20] shows excellent agreement of eddies properties, see Table VIII. Comparing approximately equal grid density, his A-grid and our 60×2–10×1, the BEM results are significantly better. Comparison of BEM results by Grigoriev [6] with our

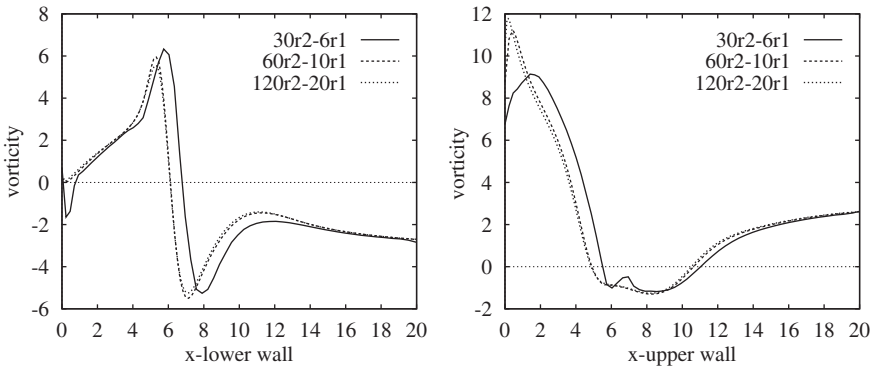

Figure 24. Backward-facing step flow at $Re = 800$. Distribution of vorticity along lower (left) and upper (right) wall for all grid densities.

Table VIII. Backward-facing step flow $Re = 800$. $X1$ length of lower wall eddy, $X2$ separation $X3$ reattachment and $L2$ length of upper wall eddy, m mass balance.

Grid	N_{sub}	N_{nodes}	$X1$	$X2$	$X3$	$L2$	m (%)
BEM 30×2–6×1	180	1440	6.78	5.51	11.04	5.53	0.50
BEM 60×2–10×1	600	4800	6.12	4.90	10.63	5.73	0.11
BEM 120×2–20×1	2400	19 200	6.10	4.86	10.51	5.65	0.01
Gartling FEM A-grid	720	6480	5.81	4.79	10.48	5.69	
Gartling FEM E-grid	32 000	28 8000	6.10	4.85	10.48	5.63	
Grigoriev BEM	558	8920	6.10	4.85	10.47	5.62	

Table IX. Backward-facing step flow $Re = 1000$. $X1$ length of lower wall eddy, $X2$ separation $X3$ reattachment and $L2$ length of upper wall eddy, m mass balance.

Grid	N_{sub}	$X1$	$X2$	$X3$	$L2$	m (%)
30×2–6×1	180	7.70	6.31	13.23	6.92	0.57
60×2–10×1	600	6.71	5.37	12.89	7.52	0.13
120×2–20×1	2400	6.55	5.17	12.67	7.50	0.01

results shows a minimal advantage of recirculation length order 0.02 by Grigoriev. We should note that Grigoriev has included the inlet region, therefore the comparison is not exactly in place.

Is $Re = 800$ steady-state? We should mention that there has been some controversy about whether there is a steady-state for 2-D flow over backward-facing step at $Re = 800$, see References [21, 22]. Based upon our work, it appears that a steady-state solution does exist. The presented BEM numerical algorithm is stable even at $Re = 1000$ producing a steady-state solution at all grid densities (see Table IX). The upper wall vorticity wiggles at coarse grid 30×2–6×1 increased with higher Re number value, see Figures 24 and 25, while results between finer grids agree well (see Figures 26 and 27).

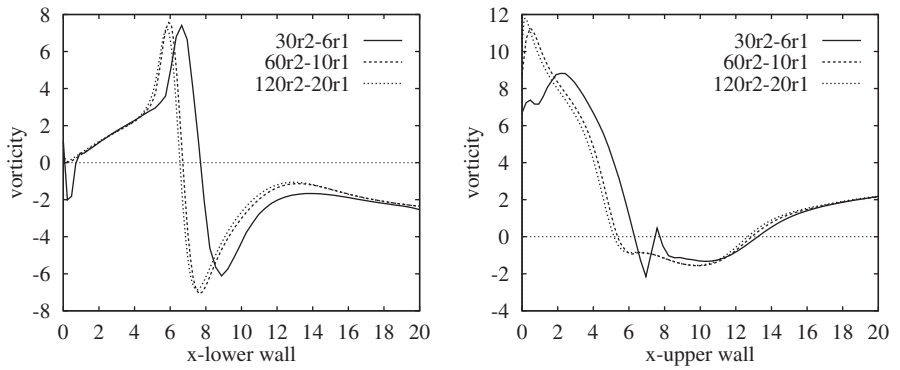

Figure 25. Backward-facing step flow at $Re = 1000$. Distribution of vorticity along lower (left) and upper (right) wall for all grid densities.

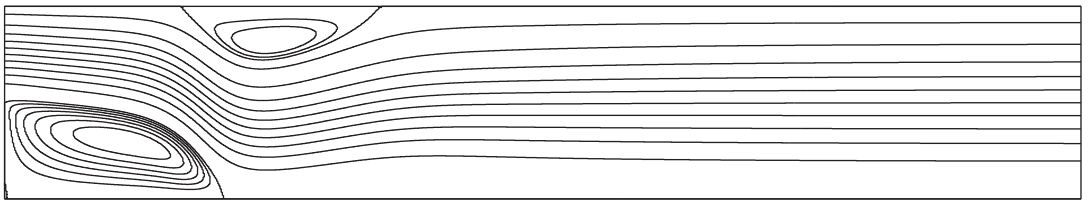

Figure 26. Backward-facing step flow at $Re = 800$. Stream function contours on finest grid. Level values are -0.030, -0.025, -0.020, -0.015, -0.010, -0.005, 0, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.49, 0.50, 0.502, 0.504.

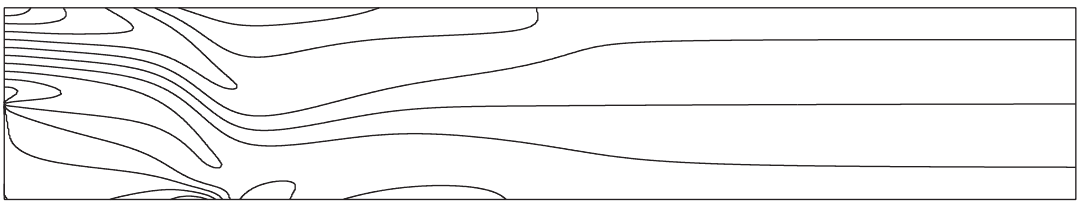

Figure 27. Backward-facing step flow at $Re = 800$. Vorticity contours on finest grid. Level values are -8.0, -6.0, -4.0, -2.0, 0.0, 2.0, 4.0, 6.0, 8.0, 10.0.

8.4. Flow in a Z shaped channel

Since the BEM are sensitive to singularities in stresses at concave corners $\alpha > 180°$, see References [9, 23], the aim of this test case is to demonstrate the effectiveness of presented BEM numerical algorithm and stream function vorticity formulation of Navier–Stokes equations on the test example of the flow in the Z shaped channel with two such concave corners. The geometry and boundary conditions are shown in Figure 28.

Two equidistant grids are used. Coarser grid A using 900 rectangular subdomains with $\Delta x = \Delta y = 0.1$ and finer grid B using 3600 subdomains with $\Delta x = \Delta y = 0.05$, see Table X (see Figure 29). Time step $\Delta t = 1.0$ is used. Under relaxation factors are URF$\psi = 1.0$ for

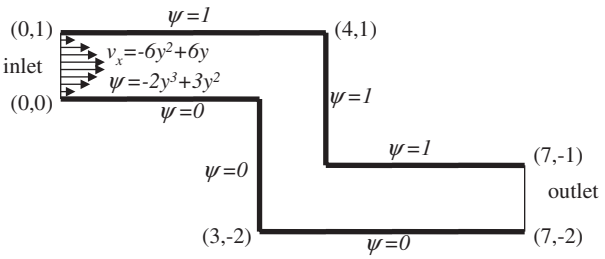

Figure 28. The geometry and boundary conditions for the flow in a 'Z' channel.

Table X. Flow in a Z shaped channel, grid properties, N_{it} number of iterations and m mass balance.

	Grid properties		$Re = 100$		$Re = 200$	
Grid	N_{sub}	$\Delta x = \Delta y$	N_{it}	m (%)	N_{it}	m (%)
A	900	0.1	219	0.039	216	0.067
B	3600	0.05	207	0.005	220	0.007

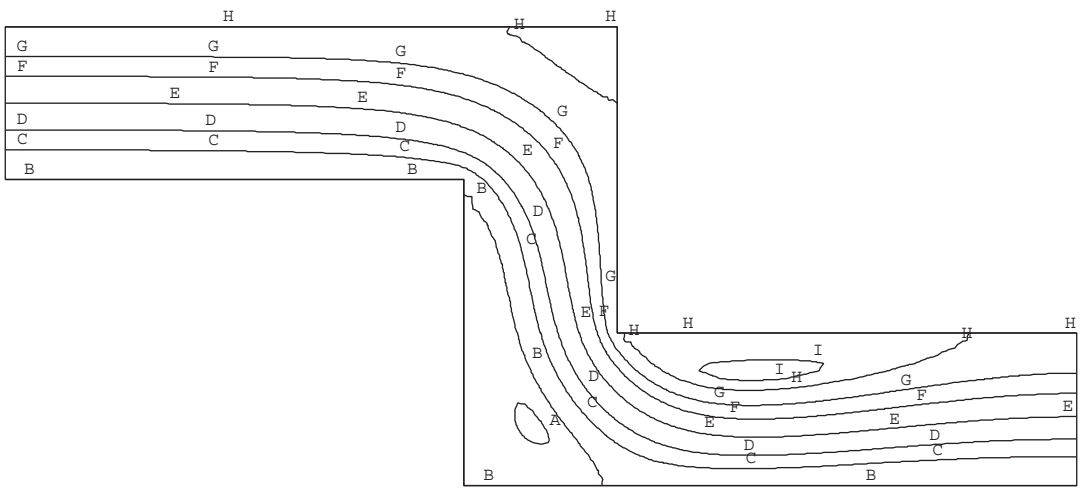

Figure 29. Flow in a Z shaped channel $Re = 100$ finest grid, stream function contours. Contour levels: $A = -0.03$, $B = 0.0$, $C = 0.1$, $D = 0.25$, $E = 0.5$, $F = 0.75$, $G = 0.9$, $H = 1.0$ and $I = 1.03$.

stream function, URF$w_\Omega = 0.10$ for domain vorticity and URF$w_\Gamma = 0.40$ for boundary vorticity at both Reynolds numbers value. The convergence criterion is 10^{-4}. The results agreement between both grids is very good.

From the vorticity contour plots it is not obvious that near both concave corners the vorticity local extreme is obtained, see Figures 30–32. This should be explained in more detail. Exactly at the concave corner the obtained wall vorticity is approximately zero. After that, vorticity rapidly increases in both wall directions. The local maximum appears immediately after the

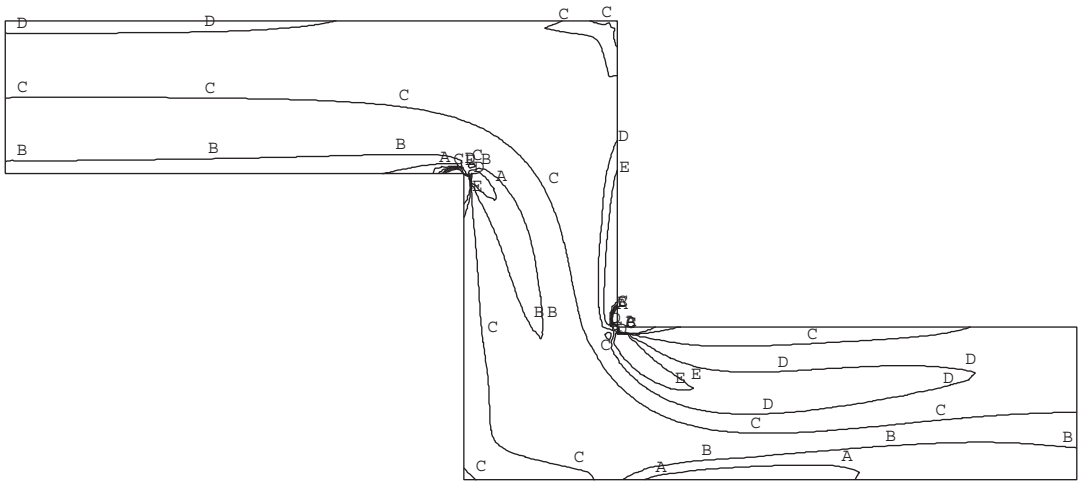

Figure 30. Flow in a Z shaped channel $Re = 100$ finest grid, vorticity contours. Contour levels: $A = -10$, $B = -5$, $C = 0$, $D = 5$, $E = 10$.

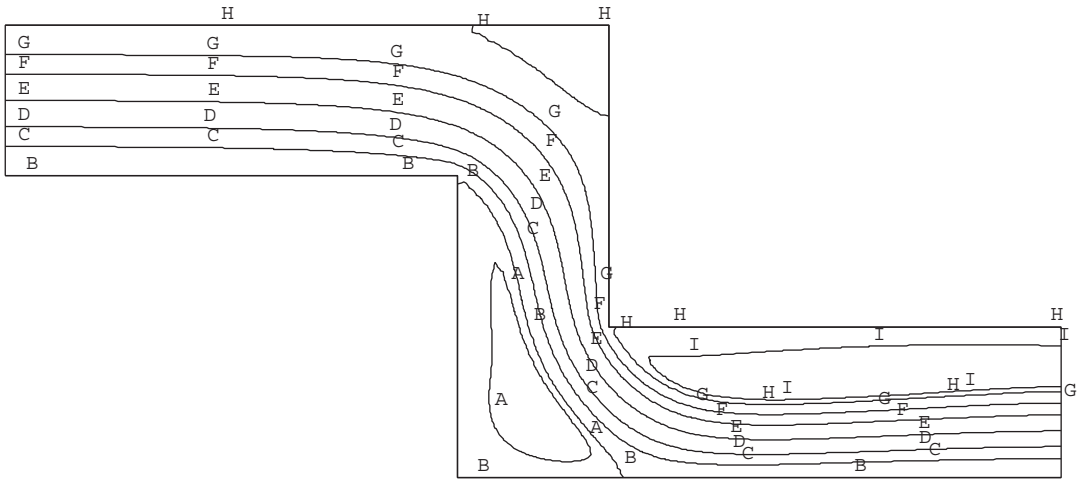

Figure 31. Flow in a Z shaped channel $Re = 200$ finest grid, stream function contours. Contour levels: $A = -0.03$, $B = 0.0$, $C = 0.1$, $D = 0.25$, $E = 0.5$, $F = 0.75$, $G = 0.9$, $H = 1.0$ and $I = 1.03$.

corner in the main flow direction. For the case $Re = 200$ the maximum vorticity at the upper concave corner is approximately 120 and at lower concave corner -180 at grid B. At lower grid density A these values are approximately 80 and -120, respectively. Obviously, these values are strongly grid dependent. In this case these values are also global maximums.

9. CONCLUSIONS

An accurate and stable BEM numerical scheme has been proposed and established to solve a general form of parabolic diffusion–convection type of governing equation.

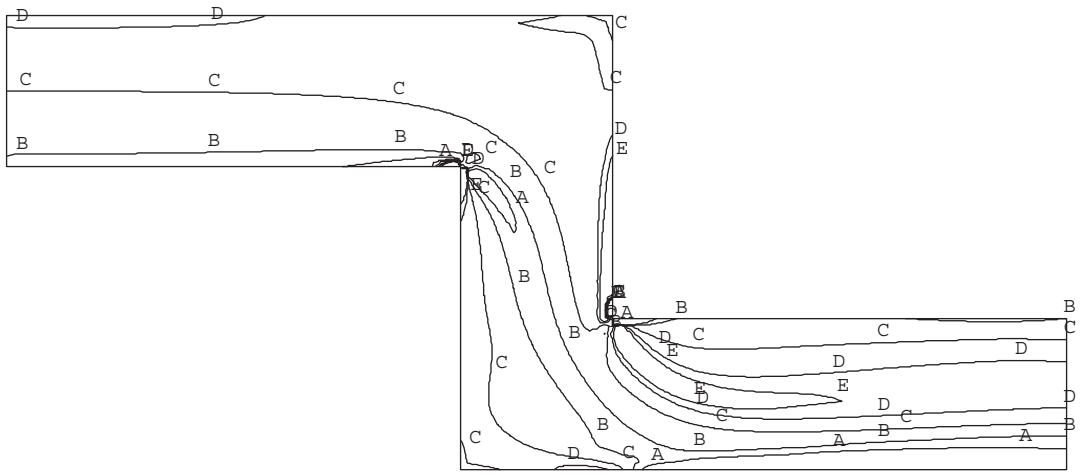

Figure 32. Flow in a Z shaped channel $Re = 200$ finest grid, vorticity contours. Contour levels: $A = -10$, $B = -5$, $C = 0$, $D = 5$ and $E = 10$.

One of the *original features* of the proposed method lies in the continuous approximation of function and normal flux using the subdomain technique. In the domain gradient continuity is required. Next, the obtained overdetermined and sparse system matrix is solved using the fast iterative linear least squares solver. The developed numerical algorithm for solving the general diffusion–convection equation is applied to solve a stream function-vorticity formulation of Navier–Stokes equations. Vorticity boundary conditions at walls are computed implicitly as a result of stream function transport equation.

Accuracy of proposed numerical scheme was demonstrated on the scalar diffusion in 'L' shape domain, see Table III, where corner value is practically equal to analytical solution. The solutions of a driven cavity flow show an excellent agreement with the reference solution at the extremely coarse grid densities, while both FEM and FVM fail to account viscous effects at this grid density, see Figures 8–12. The same conclusions can be reached in the case of a flow over backward-facing step, see Table VIII. The accuracy of BEM is evident, but the CPU consumption is much higher in comparison with FEM and FVM.

Stability of the presented BEM formulation is shown for the driven cavity flow at the Reynolds number of 50 000. The poly-regioned BEM by authors Grigoriev and Dargush [6] accomplished results only to the Reynolds number 5000, while at 7500 diverged. They used a velocity dependent fundamental solution, while the Helmholtz fundamental solution, used in this paper, has the velocity free kernel, thus saving CPU consumption.

Complex geometry test case of the Z shaped channel shows regularity of developed numerical technique for concave corners using the stream function-vorticity formulation.

REFERENCES

1. Rek Z, Škerget L. Boundary element method for steady 2D high-Reynolds-number flow. *International Journal for Numerical Methods in Fluids* 1994; **19**:334–361.
2. Telles JC. A self adaptive co-ordinate transformation for efficient numerical evaluation of general boundary element integrals. *International Journal for Numerical Methods in Fluids* 1987; **24**:959–973.

3. Žagar I, Škerget L. The numerical simulation of non-linear separation columns by boundary-domain integral formulation. *Computers and Chemical Engineering* 1995; **19**:785–790.
4. Hriberšek M, Škerget L. Iterative methods in solving Navier–Stokes equations by the boundary element method. *International Journal for Numerical Methods in Fluids* 1996; **39**:115–139.
5. Young DL, Yang SK, Eldho TI. Solution of the Navier–Stokes equations in velocity vorticity form using a Eulerian-Lagrangian boundary element method. *International Journal for Numerical Methods in Fluids* 2000; **34**:627–650.
6. Grigoriev MM, Dargush GF. A poly-region boundary element method for incompressible viscous fluid flows. *International Journal for Numerical Methods in Fluids* 1999; **46**:1127–1158.
7. Florez WF, Power H. Comparison between continuous and discontinuous boundary elements in the multidomain dual reciprocity method for the solution of the two-dimensional Navier–Stokes equations. *Engineering Analysis with Boundary Elements* 2001; **25**:57–69.
8. Grigoriev MM, Fafurin AV. A boundary element method for steady viscous flow using penalty function formulation. *International Journal for Numerical Methods in Fluids* 1997; **25**:907–929.
9. Ramšak M, Škerget L. Mixed boundary elements for laminar flows. *International Journal for Numerical Methods in Fluids* 1999; **31**:861–877.
10. Paige CC, Saunders MA. LSQR: Sparse linear equations and least-squares problems. *ACM Transactions on Mathematical Software* 1982; **8**(2):195–209.
11. Ghia U, Ghia KN, Shin CT. High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method. *Journal of Computational Physics* 1982; **48**:387–411.
12. Brebbia CA, Telles JCF, Wrobel LC. *Boundary Element Methods, Theory and Applications*. Springer: NY, 1984.
13. Škerget L. Mixed convection cavity flows. *BEM-19*, Rim, 1997; 505–514.
14. Škerget L, Hriberšek M, Kuhn G. Computational fluid dynamics by boundary-domain integral method. *International Journal for Numerical Methods in Fluids* 1999; **46**:1291–1311.
15. Banerjee PK, Butterfield R. *Boundary Element Methods in Engineering Science*. McGraw-Hill Book Company: U.K., 1981.
16. Rek Z. Boundary-domain integral method for dynamics of turbulent fluid flow. *PhD thesis*, University of Maribor, 1993.
17. LAPACK: Linear Algebra Package, http://www.netlib.org/lapack/
18. TASCflow, Version 2.4.1-3, Advanced Scientific Computing Ltd, 1996.
19. Shen J. Hopf bifurcation of the unsteady regularized driven cavity flow. *Journal of Computational Physics* 1991; **95**:228–245.
20. Gartling DK. A test problem for outflow boundary conditions—flow over a backward-facing step. *International Journal for Numerical Methods in Fluids* 1990; **11**:953–967.
21. Gresho PM, *et al.* Is the steady viscous incompressible two-dimensional flow over a backward-facing step at $Re = 800$ stable? *International Journal for Numerical Methods in Fluids* 1993; **17**:501–541.
22. Kaiktis L, Karniadakis GE, Orszag S. Onset on three-dimensionality, equilibria, and early transition on flow over a backward-facing step. *International Journal of fluids mechanics* 1991; **231**:501–528.
23. Williams ML. Stress singularities resulting from various boundary conditions in angular corners of plates in extension. *Journal of Applied Mechanics* 1952; **19**:526–528.
24. Bhardwaj D, Verma CS. *Direct Numerical Simulation of Incompressible Flows on Clusters.* HPC ASIA: Bangalore, December 17–20, India, 2002.

