Contents lists available at ScienceDirect

Journal of Computational Physics

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

A lagged implicit segregated data reconstruction procedure to treat open boundaries

M. Sani, M.S. Saidi *

Center of Excellence in Energy Conversion, School of Mechanical Engineering, Sharif University of Technology, P.O. Box 11155-9567, Tehran, Iran

ARTICLE INFO

Article history:
Received 20 August 2009
Received in revised form 3 April 2010
Accepted 6 April 2010
Available online 10 April 2010

Keywords:
Open boundary conditions
Segregated procedure
Implicitly lagged data reconstruction

ABSTRACT

The problem of treating open boundaries is still a challenging one. Applying fully developed condition is constrained to long enough domains. Without having enough physical evidence about what happens on boundaries, the domain extent could not be shortened and computational costs could not be reduced. From the advent of free (open) boundary conditions, they were confined to mixed finite element procedures. Recent works have extended their application to coupled finite volume solvers based on the shape function data reconstruction. A wider class of flow solvers available, however, rely on the segregated procedure where the velocity components and pressure are solved in succession. Moreover, many finite volume algorithms do not use the shape function reconstruction. In this work, by proposing a lagged implicit procedure, we have extended the application of the open boundary condition to these wider classes of flow solvers. The proposed extension is a combination of lagged implicit data reconstruction and overall mass conservation enforcement, which is easily applicable to any segregated and coupled flow solver. To validate the compatibility of this extension, benchmark problem of backward facing step is solved on successively truncated domains, where open boundary may pass through recirculation zones. Results show that the proposed extension works fine. For that problem, it reduced the computational domain length (and hence memory) by a factor of 4.6 and the required computational time by a factor of 21. Flow passing a cylinder is also solved which proves that the method could be applied to external flow problems as well.

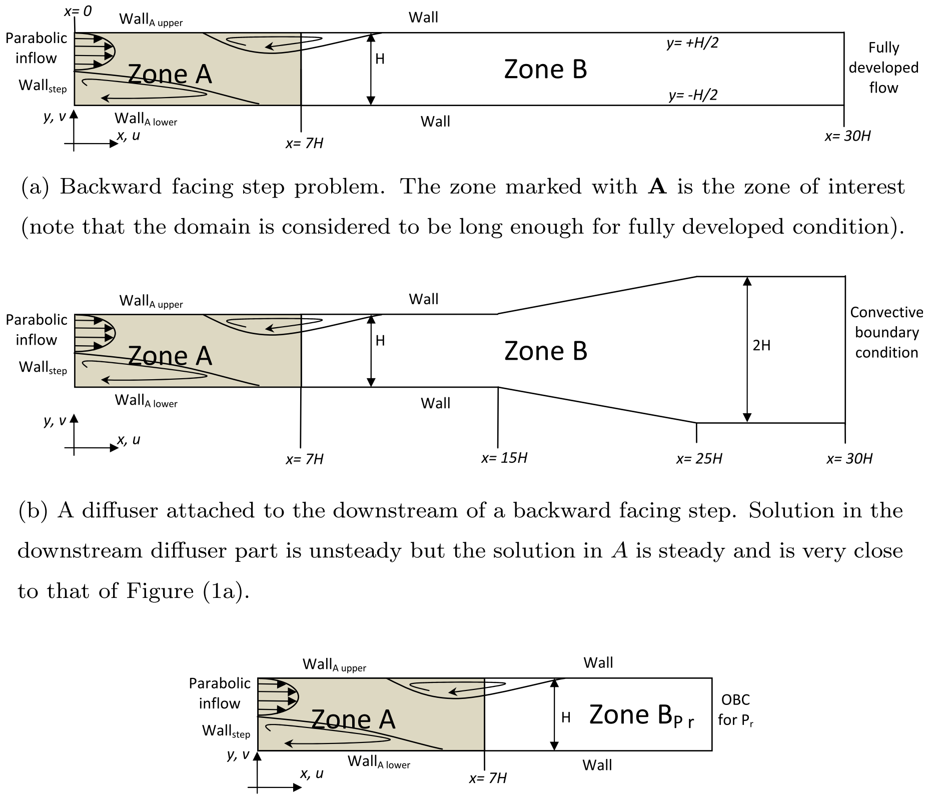
1. Introduction

To solve an incompressible flow problem, one needs to extend the computational domain to locations where the flow properties could be found from physical evidence. For example, in a duct flow, the physical evidence shows that for a long enough computational domain, the real world flow is fully developed. Unfortunately, a long enough computational domain is an expensive one. Assume that there exist an artificial boundary condition which when applied to a shorter domain, generates approximately the same result as the long enough domain with fully developed condition does, at least in the domain of interest. Hereafter, we call such a boundary condition an open boundary condition (OBC). Using this OBC, the computational costs could be reduced while the accuracy is not sacrificed.

One of the primitive OBCs for internal flow is the fully developed condition, which is based on the physical evidence and needs a long enough computational domain to generate accurate results. More advanced OBCs include those based on a

* Corresponding author. Tel.: +98 21 66165558; fax: +98 21 66000021.
E-mail addresses: msani@mech.sharif.edu (M. Sani), mssaidi@sharif.edu (M.S. Saidi).

0021-9991/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.04.005



(a) Backward facing step problem. The zone marked with **A** is the zone of interest (note that the domain is considered to be long enough for fully developed condition).

(b) A diffuser attached to the downstream of a backward facing step. Solution in the downstream diffuser part is unsteady but the solution in A is steady and is very close to that of Figure (1a).

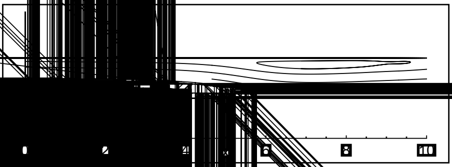
(c) Backward facing step with OBC applied to the shorter domain. The solution in A is very close to that of Figures (1a and 1b).

Fig. 1. Different problems which have zone *A* in common. Their solution in that zone is virtually the same.

hybrid coupling of discretization method with an infinite element [1] or boundary element method [2]. Non-reflecting boundary conditions are devised to absorb waves incident on the boundary. This type reduces to the fully developed flow for steady state problems. A good review on the convective boundary conditions is given by Jin and Braza [3]. Another way to absorb noises from the boundary is to attach a buffer layer to the computational domain, as proposed by Liu and Lin [4]. The method provides damping to attenuate spurious noises from the boundary. Johansson [5] and Halpern and Schatzman [6] linearized Navier–Stokes equations about the solution at far downstream (which was a constant flow) and used analytical Laplace–Fourier technique to solve these simplified equations and suggested a set of boundary condition at the outlet.

Constant pressure condition which may be applied at the far-field to external flow problems (e.g. used by Manzari [7]) is not well suited to the internal flow ones, because it needs *a priori* information about the pressure loss which is usually unknown. Moreover, it requires long enough computational domain for the streamlines to become parallel. Gartling [8] used constant total normal stress and parallel flow boundary conditions at the outlet to study the backward facing step problem. Traction (stress) free boundary condition assumes that viscous and pressure stresses are in balance $(-p+\ (1/Re)\partial u/\partial x=0)$ and the transverse velocity does not change normal to the boundary $(\partial v/\partial x=0)$. Sani and Gresho [9] used natural boundary conditions $(-p+\ (2/Re)\partial u/\partial x=0$ and $\partial u/\partial x+\partial v/\partial y=0)$ which allows traction on the boundary. Recently, Liu [10] used $\nu\partial_n\vec{u}-p\vec{n}=\vec{g}$ and $\nu(\vec{\nabla}\vec{u}+(\vec{\nabla}\vec{u})^T)\vec{n}-p\vec{n}=\vec{g}$, respectively as open and traction boundary conditions in his finite element work. His numerical method was based on the Pressure Poison Equation (PPE) formulation. Fournier et al. [11] used the approximate boundary layer equations as the governing equations on the outflow boundary for wall-bounded flows.

Originally, Papanastasiou et al. [12] introduced free boundary condition in the context of mixed finite element methods. The idea was based on evaluating the weak formulation integrals on the boundaries, using finite element shape functions instead of evaluating them with essential (Dirichlet) or natural (Neumann) conditions. This is equivalent to extending the validity of the weak form of the governing equations to the synthetic outflow, instead of replacing them with unknown essential or natural boundary conditions. Using this idea, they were able to reduce the length of the computational domain for the backward facing step problem at $Re=800$, from 35 units to 7 units, without appreciable adverse effect upstream. Wang and Sheu [13] implemented a similar idea by an implicit traction evaluation procedure based on the mixed finite element formulation. In their work, traction on the outlet boundary was related to the velocity and pressure distribution inside



unsteady downstream flow pattern corresponding to the problem defined in Fig. 1b.

3. Data reconstruction procedure

Computational fluid dynamics is a tool to approximate the behavior of continuous fluid systems by reducing their infinite degrees of freedom to some finite affordable degrees of freedom. This is usually done by finding approximation to the solution in some finite number of space and time locations. If an approximation to the solution for other space/time locations is sought, some kind of data reconstruction (interpolation) is used. The natural tool for this purpose in finite element method is shape function reconstruction [18]. In unstructured finite volume methods there are two schools of thought. One way is using the shape function tool as in finite element method, e.g. [17,19]. The other is based on interpolation using the neighboring points where approximate data are available (usually with distance weighting).

One common way to reconstruct data without shape function, is to find some approximation to the gradients. Then interpolation in the neighborhood of the cell center $\vec{r}_{P_0}$ is simply carried out by

$$\phi(\vec{r}) \approx \phi_{P_0} + (\vec{\nabla}\phi)_{P_0} \cdot (\vec{r} - \vec{r}_{P_0}) \tag{1}$$

which is valid to second order. Least square procedure could be used to evaluate the gradient [20]. Since Eq. (1) is valid near $\vec{r}_{P_0}$, it gives ϕ at its neighbors, $\vec{r}_{P_j}$, as

$$\phi_{P_j} = \phi_{P_0} + (\vec{\nabla}\phi)_{P_0} \cdot (\vec{r}_{P_j} - \vec{r}_{P_0}) = \phi_{P_0} + (\vec{\nabla}\phi)_{P_0} \cdot \vec{d}_j. \tag{2}$$

Rearranging for the gradient gives:

$$(\vec{\nabla}\phi)_{P_0} \cdot \vec{d}_j = \phi_{P_j} - \phi_{P_0}. \tag{3}$$

This is always an over determined system of equations for gradient components since in three (two) dimensions there are at least four (three) neighbors for each cell but three (two) components of the gradient are to be solved for. Following the standard least square procedure, the gradient components could be obtained as:

$$(\vec{\nabla}\phi)_{P_0} = D^{-1} \sum_j d_j^T (\phi_{P_j} - \phi_{P_0}), \tag{4}$$

where, j runs over all of the neighbors of P_0, d_j^T is the single column notation for $\vec{d}_j$ and D is a symmetric three by three (two by two in 2D) geometry-dependent matrix defined as:

$$D = \sum_j d_j^T d_j = \begin{bmatrix} a & b & c \\ b & d & e \\ c & e & f \end{bmatrix}. \tag{5}$$

The inverse of D is computed analytically as:

$$D^{-1} = \frac{1}{adf + 2bce - ae^2 - dc^2 - fb^2} \begin{bmatrix} df - e^2 & ce - bf & be - cd \\ ce - bf & af - c^2 & bc - ae \\ be - cd & bc - ae & ad - b^2 \end{bmatrix}. \tag{6}$$

These give:

$$\phi(\vec{r}) \approx \phi_{P_0} + \left(D^{-1} \sum_j d_j^T (\phi_{P_j} - \phi_{P_0}) \right) \cdot (\vec{r} - \vec{r}_{P_0}) = w_0(\vec{r})\phi_{P_0} + \sum_j w_j(\vec{r})\phi_{P_j} \tag{7}$$

in which $w_j(\vec{r}) = D^{-1} d_j^T \cdot (\vec{r} - \vec{r}_{P_0})$ and $w_0(\vec{r}) = 1 - \sum_j w_j(\vec{r})$. This is effectively a weighted data reconstruction method.

4. Governing equations and their discretization

For the incompressible flow, the continuity and momentum equations are:

$$\int_\Omega \rho \vec{V} \cdot d\vec{A} = 0, \tag{8}$$

$$\frac{\partial}{\partial t} \int_\Omega \rho \phi dV + \int_\Gamma \phi(\rho \vec{V} \cdot d\vec{A}) = S_p + \int_\Gamma D_\phi \frac{\partial \phi}{\partial n} dA + S_\phi, \tag{9}$$

where ϕ could be any transported scalar including velocity components (u, v, w) and S_p represents the pressure term which is only applicable to the momentum equations as:

$$\vec{S}_p = -\int_\Gamma p d\vec{A}. \tag{10}$$

Although the methods presented here on treating open boundaries, do not depend on any particular discretization scheme, for the sake of clarity, we follow with discretizing these equations on a co-located cell centered polyhedral grid by a second order implicit finite volume method.

For integration in time, we follow the implicit second order three time levels method. The unsteady equations are symbolically rewritten as:

$$\frac{dy}{dt} = H, \tag{11}$$

where y represents the volume integral in Eq. (9) and H contains all other terms. Discretization gives:

$$\frac{3y^n - 4y^{n-1} + y^{n-2}}{2\Delta t} = \left(\frac{dy}{dt}\right)^n + O(\Delta t^2) = H^n + O(\Delta t^2). \tag{12}$$

Applying the above time discretization, Eq. (9) could be discretized in space for a polyhedral cell, P_0, having N_j faces to:

$$a_{P_0}\phi_{P_0} + \sum_j^{N_j} a_{P_j}\phi_{P_j} = b_{P_0} \tag{13}$$

in which a_{P_0}, a_{P_j} and b_{P_0} carry the effects of implicitly discretized integrals. The volume integrals are discretized to second order using the mid-point rule:

$$\int_{\Delta V_{P_0}} \psi dV \approx \psi_{P_0}\Delta V_{P_0}. \tag{14}$$

Convection surface integrals are discretized using the same rule as:

$$\int_j \phi(\rho\vec{V}\cdot d\vec{A}) \approx \dot{m}_j\phi_j, \tag{15}$$

where j represents the face center. To avoid pressure checker-boarding, the mass flux in Eq. (15) is obtained from the standard Rhie-Chow [21] (momentum based) interpolation between cells P_{0_j} and P_{1_j}, which have face j in common as:

$$u_{n_j} = \frac{\dot{m}_j}{\rho A_j} = \overline{(\vec{u})}_j\cdot\hat{n}_j + \overline{\left(\frac{\Delta V}{a_0}\frac{\delta P}{\delta n_j}\right)}_j - \overline{\left(\frac{\Delta V}{a_0}\right)}_j\left(\frac{\delta P}{\delta n_j}\right)_j, \tag{16}$$

where $\overline{(\cdot)}_j$ means any interpolation to the face center, like CDS, a_0 is the main diagonal element in the discretized momentum equation of the corresponding cell and $\delta(\cdot)/\delta n$ is the discretized face normal derivative operator. To find an approximation to ϕ_j for Eq. (15), any upwind scheme could be used. Here, the second order upwind (SOU) interpolation is used. First, using the mass flux, the upwind cell, P_U, is found. Then using ϕ_{P_U} and its gradient (c.f., Section 3), face center value is approximated to second order as:

$$\phi_j \approx \phi_{P_U} + (\vec{\nabla}\phi)_{P_U}\cdot(\vec{r}_j - \vec{r}_{P_U}). \tag{17}$$

Diffusion surface integrals are discretized using

$$\int_j D_\phi\frac{\partial\phi}{\partial n}dA \approx D_\phi\frac{\delta\phi}{\delta n_j}A_j. \tag{18}$$

To approximate the normal derivative, virtual points are used. The virtual point in cell P_0 related to face j is defined as the normal projection of the cell center on the face normal:

$$\vec{r}_{P'_0} = \vec{r}_j + [\hat{n}_j\cdot(\vec{r}_{P_0} - \vec{r}_j)]\hat{n}_j, \tag{19}$$

where $\hat{n}_j$ is the unit normal vector of the face j (Fig. 3). The virtual point in cell P_1 is defined as the mirror image of the virtual point in P_0 with respect to the common face. The line connecting virtual points P'_0 and P'_1 is orthogonal to the face and passes through the face center. Therefore, to second order,

$$\frac{\delta\phi}{\delta n_j} \approx \frac{\phi_{P'_1} - \phi_{P'_0}}{|\vec{r}_{P'_1} - \vec{r}_{P'_0}|}, \tag{20}$$

where values at virtual points are obtained using gradient based interpolation, e.g. $\phi_{P'_0} \approx \phi_{P_0} + (\vec{\nabla}\phi)_{P_0}\cdot(\vec{r}_{P'_0} - \vec{r}_{P_0})$.

Continuity equation is used with SIMPLE algorithm to derive the pressure correction equation. For this algorithm, currently available velocity components and pressure are assumed to be predictions to the correct values, requiring a set of corrections which are related to each other as:

$$\vec{u}'_{P_0} \approx -\frac{\Delta V_{P_0}}{a_0}(\vec{\delta}P')_{P_0}, \tag{21}$$

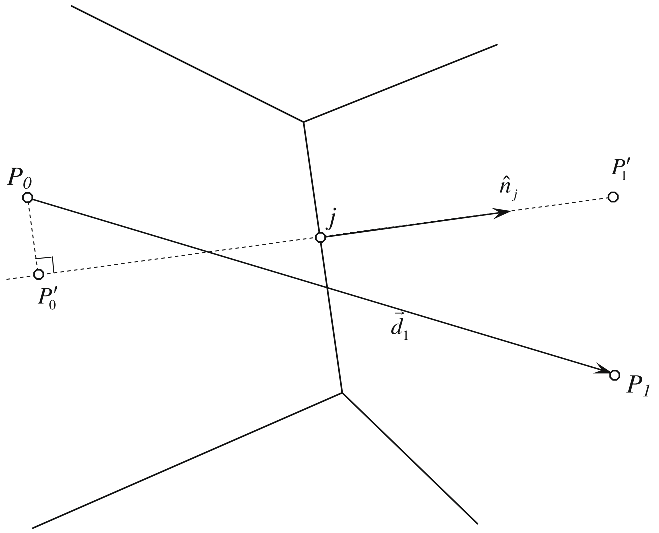
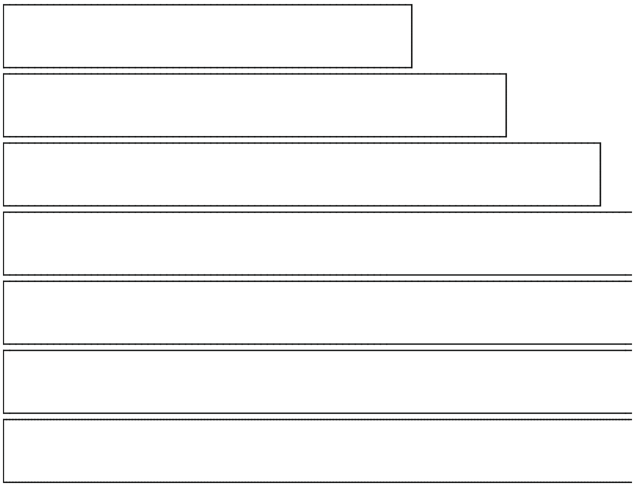


Fig. 3. Definition of virtual points.



where $\vec{\delta}$ is the discrete gradient operator. Performing the required mass flux correction from the velocity corrections, and substituting in the discretized continuity equation gives the pressure correction equation as

$$\sum_{j}^{N_j}\left(\rho\overline{\left(\frac{\Delta V}{a_0}\right)}_j(\vec{\delta}P')_j.\vec{A}_j\right)=\sum_{j}^{N_j}(\dot{m}_j), \tag{22}$$

where $(\vec{\delta}P')_j.\vec{A}_j$ is obtained using Eq. (20) and

$$(\vec{\delta}P')_j.\vec{A}_j=\frac{\delta P'}{\delta n}A_j. \tag{23}$$

To avoid special treatment for cells near boundary, ghost cells are used outside the physical domain. Therefore, each boundary face, j, is surrounded by one real cell, P_0, and one ghost cell, P_1. The ghost cell center is defined as the mirror image of the virtual point inside the real cell, P'_0, with respect to the boundary face. With this choice, face center value and face normal derivative are easily computed to second order as

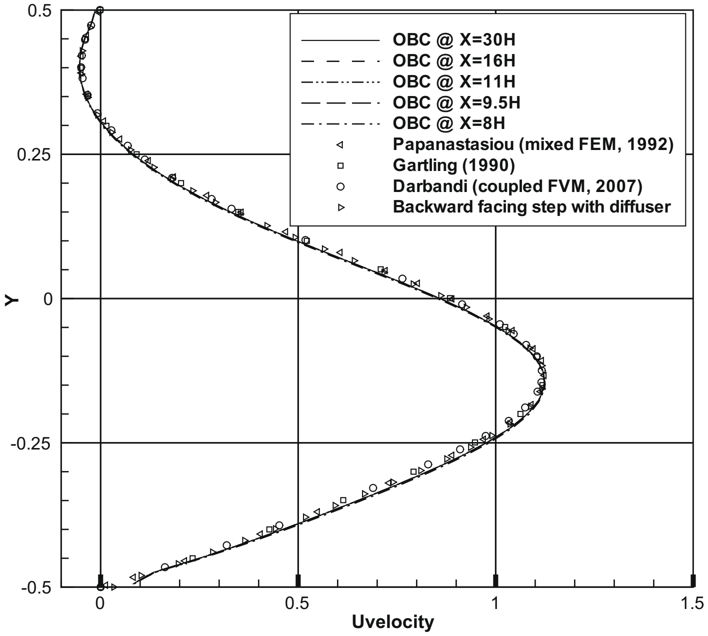


Fig. 5. *u* Velocity profile at *x* = 7.

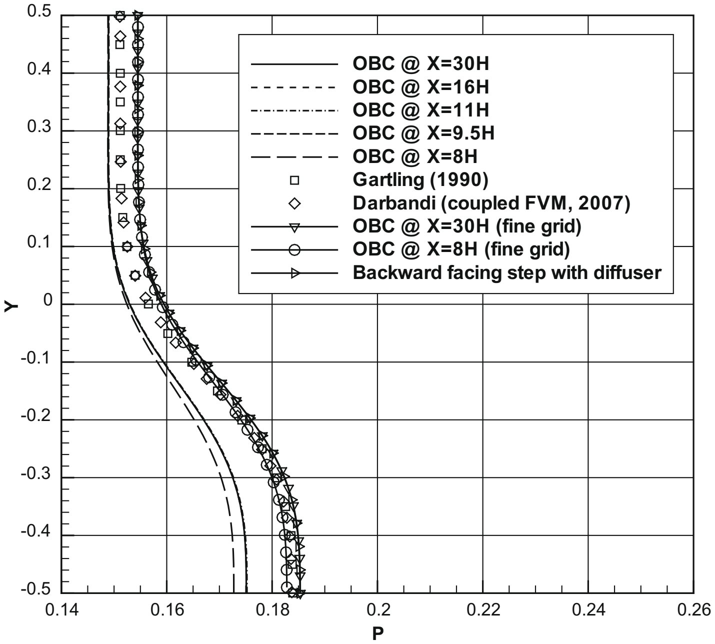


Fig. 6. Pressure profile at *x* = 7*H*.

$$\phi_j \approx \frac{\phi_{P_1} + \phi_{P_0'}}{2}, \tag{24}$$

$$\frac{\partial \phi}{\partial n_j} \approx \frac{\phi_{P_1} - \phi_{P_0'}}{|\vec{r}_{P_1} - \vec{r}_{P_0'}|}. \tag{25}$$

Boundary conditions are considered as the governing equations for the ghost cells.

5. Overall mass balance enforcing

Segregating the solution of the momentum equations from the continuity driven pressure correction equation, gives rise to a velocity field which may violate the overall mass balance. To avoid this overall mass imbalance in the intermediate solution, mass balance is enforced by updating the mass flux over the open boundary before solving the pressure correction equation. The overall mass defect is computed from

$$\dot{m}_{defect} = \sum_{i}^{inflows} \dot{m}_i - \sum_{i}^{outflows} \dot{m}_i. \tag{26}$$

The share of each face from this defect is computed in proportion to its face area:

$$\dot{m}_j^c = \frac{A_j}{\sum_i^{outflows} A_i} \dot{m}_{defect}. \tag{27}$$

This correction excites the pressure correction equation by generating a source term in the equation of the last row of cells near the outflow boundary, thereby, forcing this equation to balance the overall mass flow. At convergence, this correction vanishes and does not affect the final solution, but in the course of segregated iterations, it prevents the solution to diverge because of the overall mass imbalance. With this mass flux correction, boundary conditions of the pressure correction equation become Neumann condition ($\delta P' / \delta n = 0$).

6. Implicit reconstruction treatment of open boundary condition

The idea of open (free) boundary condition (OBC) [12] is to extend the validity of the numerical discretizations to the boundary. In mixed finite element, it is applied by evaluating the weak formulation integrals using the shape function data reconstruction tool, thereby, linking nodal values over the boundary to the domain internal ones. The fully implicit conservative coupled finite volume method of Darbandi et al. [17] uses the same shape function based data reconstruction to link the boundary nodal values to the domain internal ones. In both cases, all of the velocity components and pressure nodal values are to be solved together (coupled FVM and mixed FEM procedures).

Here these two constraints are removed using the segregated procedure presented in Section 2 and the gradient based data reconstruction procedure of Section 3. To mimic what is called fully implicit conservative method by Darbandi et al. [17], it suffices to implicitly link the value of any variable of interest, ϕ, at ghost cells on open boundaries to the ones of the corresponding real cells. This effectively provides ϕ or $\delta\phi/\delta n$ values for the conservation equations (so the method remains conservative). Of course, this does not guarantee overall mass conservation, because the mass and momentum equations are not solved together (the issue which was addressed in Section 5).

To implicitly link the value at a ghost cell (P_G) over an open boundary, its data is reconstructed from the corresponding real cell (P_R) by

$$\phi_{P_G} = w_0(\vec{r}_{P_G})\phi_{P_R} + \sum_j w_j(\vec{r}_{P_G})\phi_{P_j}, \tag{28}$$

where G and R are used to represent the ghost and real cells and j runs over the neighbors of the real cell. The equation for the ghost cell should be rearranged so as to comply with implicit reconstruction. We propose the equation for the ghost cell as:

$$(1 - w_{P_G})\phi_{P_G} - w_0\phi_{P_R} = \sum_{j \neq G} w_j \phi_{P_j}, \tag{29}$$

where w(s) are evaluated with $\vec{r} = \vec{r}_{P_G}$. Eq. (29) is to be used for the ghost cell as its governing equation. Unfortunately, using Eq. (29), the computational stencil of the ghost cell becomes extended to the neighbors of the corresponding real cell. To avoid this large computational stencil, the right hand side is treated explicitly (lagging). By lagging, it is meant that the values of the last outer iteration are used. This does not have any effect on the converged solution but may delay the convergence.

It is worth noting that the proposed method does not depend on using ghost cells and the same could be done for example for face centers. This, of course, requires special treatment for near boundary cells (like using one sided derivatives).

Eq. (29) could not be applied for pressure because no pressure equation is used. Pressure correction equation has Neumann boundary condition from the overall mass balance enforcement and could not be used for this purpose. To apply data reconstruction for pressure, ghost cell values are obtained by gradient based data reconstruction before solving the momentum equations (step (2) of Section 2).

7. Backward facing step at *Re* = 800

To investigate the validity of the OBC extension to the segregated flow solvers with implicit reconstruction, backward facing step problem is solved at *Re* = 800. As a standard test case, researchers on OBC have solved this problem on successively

Table 1
Comparison of the computational costs for different domain cuts. Non-linear convergence criterion: $\sqrt{\sum(\Delta\phi)^2} < 10^{-7} \times \sqrt{\sum\phi^2}$.

Case X_{end}/H=	30	16	11	9.5	8	6.5
Relative number of outer iterations	100	100	100.12	99.92	95.96	69.21
Relative memory required	100	68.75	36.67	31.67	26.67	21.67
Relative computational time	100	31.1	19.5	11.6	7.94	4.75

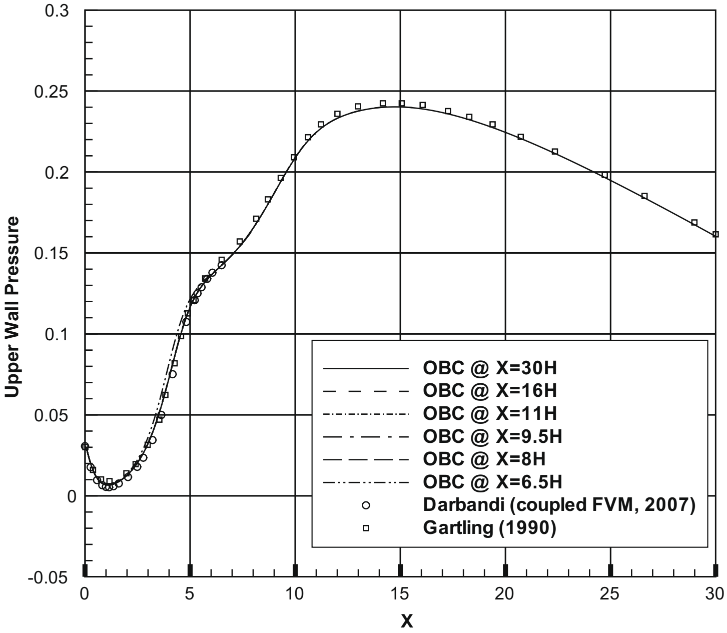


Fig. 7. Pressure distribution along the upper wall.

truncated domains to investigate the capabilities of their methods. The problem is defined in Fig. 1(a). The flow enters the domain via the entrance at upper half of the left side with a parabolic profile defined by $u(y) = 6U_{mean}(y/H)(2 - 4y/H)$ where the origin for the coordinate system is placed on the center of the left side of the domain. *Re* number is defined based on the channel height (*H*, twice the step height) and mean inlet velocity (U_{mean}).

The fully developed boundary condition generates accurate results when the domain length is larger than $X_{end} = 30H$, where *H* is the channel height. Papanastasiou et al. [12], using mixed finite element, were able to reduce the domain length to $X_{end} = 7H$ without appreciable change upstream. Darbandi et al. [17] used a coupled finite volume method based on the shape function data reconstruction and were able to place the open boundary at $X_{end} = 6.5H$. It is worth noting that the lower recirculation zone is extended from the step to $X \approx 6.1H$ and the upper separation bubble extends from $X \approx 4.9$–$10.5H$. With this in mind, Darbandi et al. [17] used truncated domains with OBC at $X_{end} = 6.5$, 8.0, 9.5, 11.0, 16.0 and 30.0*H* to test their method. The cut at 11.0*H* is right after the reattachment point of the upper eddy. The cut at 9.5*H* is located before the reattachment point of that eddy. The cut at 6.5*H* is also a sever cut, because it cuts through the upper eddy near its upstream end. Moreover, it is close to the lower eddy reattachment point.

We follow Darbandi et al. [17] in placing cuts and also use the same uniform mesh to compare our results with theirs. The mesh is a uniformly distributed one, with 301 × 41 nodes on the longest domain, and the upstream grid point distribution is not affected with the cutting. It is worth noting that the grid is not clustered near the walls, the inlet or outlet boundaries. This opposes the good engineering practice in computational fluid dynamics where the mesh is refined near predictable high gradient zones to capture fine events. Nevertheless, using a uniform grid and obtaining good results, more strongly confirms the validity of the OBC treatment.

Fig. 4 shows the streamlines for different domain cuts. As is evident, shrinking the domain length does not affect the upstream solution. To be more specific, the locations of the reattachment points of the lower and upper eddies are correctly captured (remained unchanged) with progressively moving the open boundary towards the step, even though it cuts the upper eddy near its upstream end. The exception is a very small change in the upper eddy separation point. This was also observed by Darbandi et al. [17] and was removed by refining the grid. Therefore, this behavior is not caused by the OBC but is related to the grid having insufficient resolution. We also used a finer grid (each cell divided to four cells) and reached to the same conclusion.

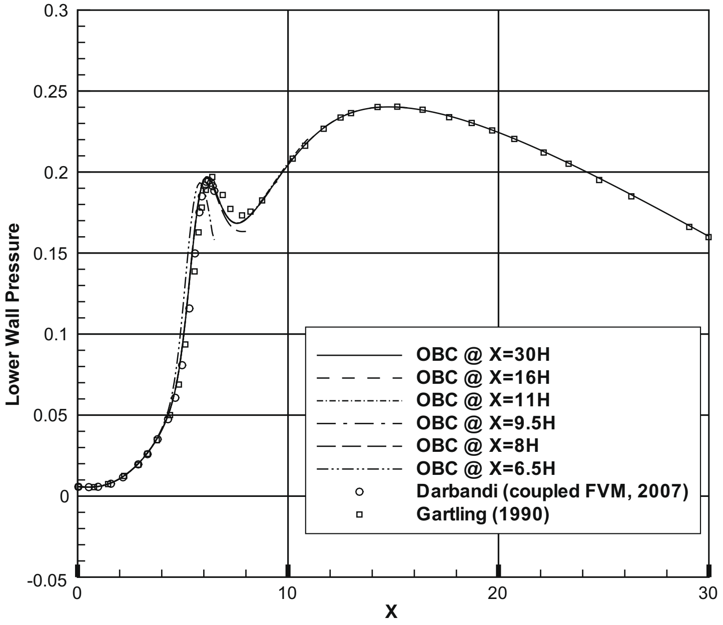


Fig. 8. Pressure distribution along the lower wall.

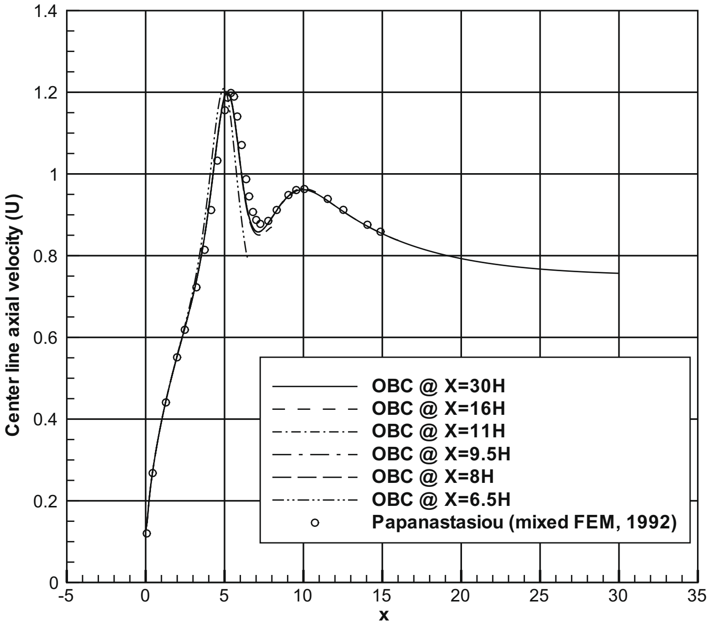
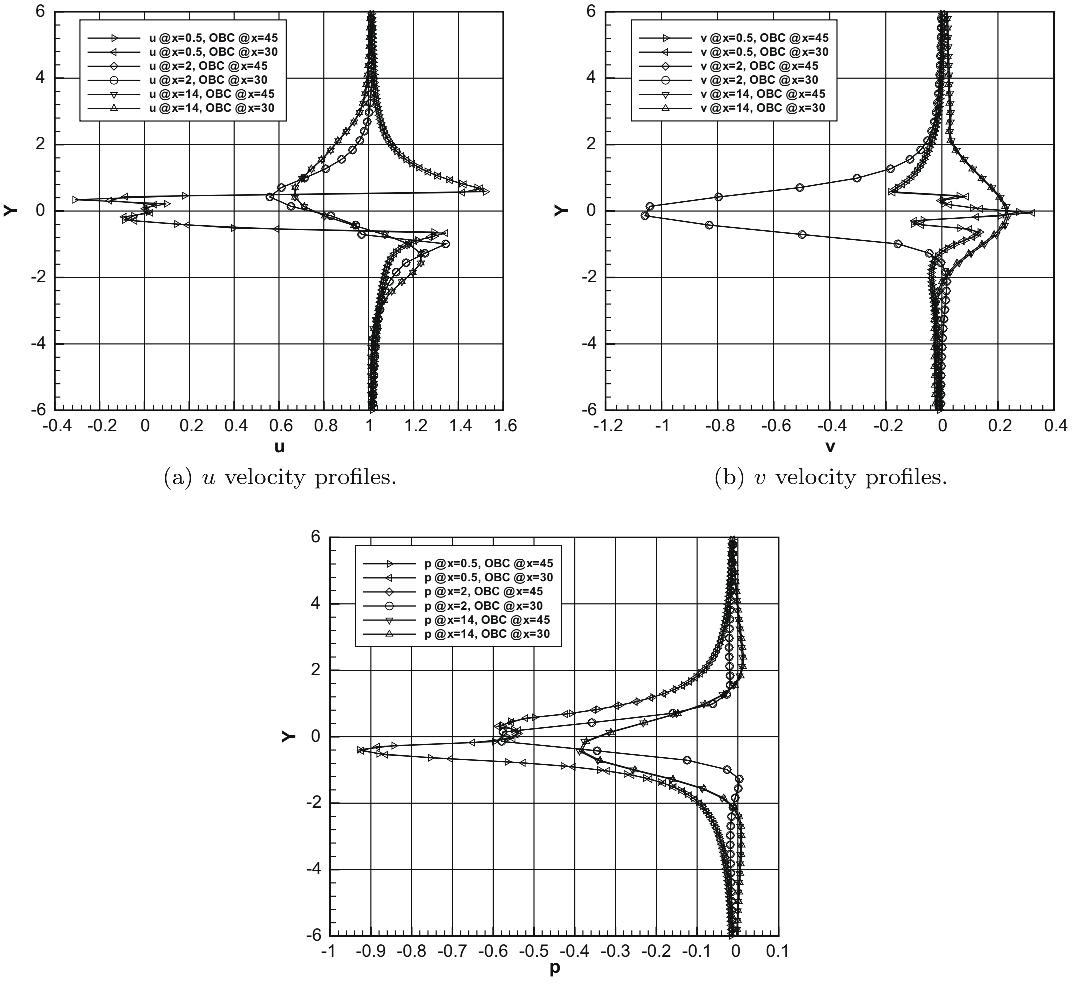


Fig. 9. Axial velocity along the center line.

It is worth noting that shrinking the domain reduced the number of cells inside the domain in linear proportion to the domain length. This caused the system of equations to become smaller thereby reducing the memory and run time costs. In comparison, using $X_{end} = 30H$ required a computational time approximately 21 times that of $X_{end} = 6.5H$. The memory costs scaled linearly with domain length (4.6 times). Table 1 summarizes relative computational costs.

More quantitative comparison is made by extracting velocity and pressure profiles. The location $X = 7H$ is a critical location because it is very close to the center of the upper eddy. This section is used by other researchers, e.g. [8,12,17], so comparison could be made. Axial velocity and pressure profiles are compared in Figs. 5 and 6. Axial velocity is in good agreement but pressure is not as good. Because there were different pressure bases, we have moved the base for all of the cases, so that the pressure at the origin becomes zero. As is evident, the trend for all of the results are the same but values do not agree very



(a) u velocity profiles.

(b) v velocity profiles.

(c) Pressure profiles.

Fig. 12. Horizontal and vertical velocity and pressure profiles at $x = 0.5$, 2 and 14 for flow passing a cylinder at $t = 80$ ($Re = 1000$).

faces per control volume for the structured mesh used. This is twice the number of faces each control volume has in the cell centered schemes. Of course, more degrees of freedom and more faces per control volume requires much more computational cost, which guarantees a more accurate resolution of the flow field. Considering the complex behavior of the flow at section $X = 7H$, and the low resolution mesh used in both works, this 5% difference is tolerable. Nevertheless, because the results for different domain cuts agree well with each other, this is not related to the application of OBC. To investigate if the mesh resolution is the reason, we divided each computational cell to four cells. The results of this finer mesh are added to the figure. Since for this finer mesh, profiles match with that of Darbandi et al. [17], the error should be related to the truncation errors of the numerical scheme. Again, the results of $X_{end} = 30H$ match with that of $X_{end} = 8H$ for the finer grid, which proves that OBC is applied correctly. The same comment applies to the other figures which follow.

Pressure distribution along the upper and lower walls are shown in Figs. 7 and 8. The linear tail portion of the figures shows the fully developed zone. Cutting the domain even at $X = 6.5H$ does not destroy the quality of the results. Axial velocity component along the domain centerline is presented in Fig. 9, which shows close agreement with the reference work of [12] using mixed finite element.

All of the above results were obtained using time step size equal to 1×10^{30} which gives the steady state results. Although not reported here, the impulsively started backward facing step problem is also solved with time steps ranging from $\Delta t = 0.001–1 \times 10^{30}$ on all of the truncated domains. All cases converged to the same steady state result. Unsteady results were in close match together and to that of the full length domain with convective outflow boundary conditions.

8. Flow passing a circular cylinder

The method described above and applied to the benchmark backward facing step problem is not restricted to the internal flow problems. Fig. 10 shows the problem definition for the flow passing a circular cylinder. Fluid density is set to unity. The mesh is a combination of clustered body conforming quad cells near the cylinder and triangular cells covering the rest of the domain. The related Strouhal number defined as $St = fD/V_\infty$ describes the frequency of the vortex shedding.

Table 2 summarizes the results for $Re = 200$ and 1000. As is evident, St numbers are in close match with Manzari [7] and He et al. [22] results, which guarantees the well behavior of the method in external flow problems.

To check whether the domain length could be reduced, for $Re = 1000$, the OBC was moved from $X_{end} = 45$ to $X_{end} = 30$, which means that the domain length was shrunk by 25%. As before, the upstream mesh left untouched. The streamlines for both of the cases are shown in Fig. 11. Both simulation results are overlayed in Fig. 11(c) which shows almost no difference. To magnify the differences, a close up near the cylinder is chosen. Even with this magnification, the streamlines appear essentially the same. It is worth noting that the streamlines are drawn for $t = 80$ which is approximately 16 times the vortex shedding period. After such a long time, numerical errors may accumulate. Results clearly demonstrate that even after such an accumulation, the streamlines do not show any meaningful difference. Profiles of the velocity components and pressure at the positions $x = 0.5$, 2, 14 at $t = 80$ are shown in Fig. 12. As is evident, the results for the full length and truncated length domains are essentially equivalent. St number related to this truncated case is also presented in Table 2 which shows no difference compared to the full length domain value.

9. Conclusion

From the advent of open (free) boundary conditions they were confined to two restrictions; shape function data reconstruction and coupled (mixed) pressure–velocity solvers. In this work, both restrictions were removed by implicitly coupling the open boundary data with domain discretized equations. The key elements were segregating the pressure and velocity equations (e.g. by means of SIMPLE algorithm), overall mass balance enforcing and lagging the implicit shape function independent data reconstruction. As a result, the proposed open boundary treatment could be applied to virtually any polyhedral grid solver no matter it is coupled or segregated and no matter it uses shape function data reconstruction or not. The compatibility of the proposed procedure with previous works which had used coupled solution algorithms and shape function reconstruction was shown using the backward facing step problem. The tests showed that the method works perfectly. Applying such kind of boundary condition to the sample backward facing step problem allowed the use of a very smaller domain (6.5H vs. 30H), which reduced the memory requirement by a factor of 4.6 and speeded up the solution with a factor of 21. To prove the applicability of the method to external flow problems, flow passing a circular cylinder was also solved. The domain reduction perfectly works in this case, too.

Acknowledgments

This work was carried out in Center of Excellence in Energy Conversion, Sharif University of Technology. We are also grateful to Professor Darbandi of Aerospace Department who provided us with some data from his previous work on the subject.

References

[1] S.S. Saini, P. Bettess, O.C. Zienkiewicz, Coupled hydrodynamic response of a gravity dam using finite and infinite elements, Earthquake Eng. Struct. Dynam. 6 (1978) 33–374.

[2] C.A. Felippa, Interfacing finite element and boundary element discretizations, Appl. Math. Mech. English Ed. 5 (1981) 383–386.

[3] G. Jin, M. Braza, A nonreflecting outlet boundary condition for incompressible unsteady Navier–Stokes calculations, J. Comput. Phys. 107 (1993) 239–253.

[4] C. Liu, Z. Liu, High order finite difference and multigrid methods for spatially evolving instability in a planar channel, J. Comput. Phys. 106 (1993) 92–100.

[5] B.C.V. Johansson, Boundary conditions for open boundaries for the incompressible Navier–Stokes equation, J. Comput. Phys. 105 (1993) 233–251.

[6] L. Halpern, M. Schatzman, Artificial boundary conditions for incompressible viscous flows, SIAM J. Math. Anal. 20 (1989) 308–353.

[7] M. Manzari, A time-accurate finite element algorithm for incompressible flow problems, Int. J. Numer. Methods Heat & Fluid Flow 13 (2) (2003) 158–177.

[8] D.K. Gartling, A test problem for outflow boundary conditions-flow over a backward-facing step, Int. J. Numer. Methods Fluids 11 (1990) 953–967.

[9] R.L. Sani, P.M. Gresho, Resume and remarks on the open boundary condition minisymposium, Int. J. Numer. Methods Fluids 18 (1994) 983–1008.

[10] J. Liu, Open and traction boundary conditions for the incompressible Navier–Stokes equations, J. Comput. Phys. 228 (19) (2009) 7250–7267, ISSN 0021-9991. doi:10.1016/j.jcp.2009.06.021.

[11] G. Fournier, F. Golanski, A. Pollard, A novel outflow boundary condition for incompressible laminar wall-bounded flows, J. Comput. Phys. 227 (15) (2008) 7077–7082, doi:10.1016/j.jcp.2008.03.038.

[12] T.C. Papanastasiou, N. Malamataris, K. Ellwood, New outflow boundary condition, Int. J. Numer. Methods Fluid 14 (5) (1992) 587–608.

[13] M. Wang, T. Sheu, Implementation of a free boundary condition to Navier–Stokes equations, Int. J. Numer. Methods Heat Fluid Flow 7 (1997) 95–111.

[14] D.F. Griffiths, The no boundary condition outflow boundary condition, Int. J. Numer. Methods Fluid 24 (1997) 393–411.

[15] M. Darbandi, G.E. Schneider, S. Vakilipour, A full conservative approach to simulate outflow at artificial boundaries, in: Proceedings of the 12th Annual Conference of the CFD Society of Canada, CFDSC, Ottawa, ON, Canada.

[16] M. Darbandi, S. Vakilipour, Implementation of free boundary condition at arbitrary sections using implicit conservative statements, in: Proceedings of the 4th European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2004.

[17] M. Darbandi, S. Vakilipour, Using fully implicit conservative statements to close open boundaries passing through recirculations, Int. J. Numer. Methods Fluids 53 (3) (2007) 371–389.

[18] R. Lohner, Applied Computational Fluid Dynamics Techniques: An Introduction Based on Finite Element Methods, second ed., John Wiley & Sons, 2008, ISBN 978-0-470-51907-3.

[19] T.T. Bui, A parallel, finite-volume algorithm for large-eddy simulation of turbulent flows, Comput. Fluid 29 (8) (2000) 877–915.

[20] H. Hadzic, Development and Application of a Finite Volume Method for the Computation of Flows Around Moving Bodies on Unstructured, Overlapping Grids, Ph.D. Thesis, Technischen Universitat Hamburg-Harburg, 2005.

[21] C.M. Rhie, W.L. Chow, Numerical study of the turbulent flow past an airfoil with trailing edge separation, AIAA J. 21 (11) (1983) 1525–1532.

[22] J.-W. He, R. Glowinski, R. Metcalfe, A. Nordlander, J. Periaux, Active control and drag optimization for flow past a circular cylinder, J. Comput. Phys. 163 (2000) 83–117.