A simple and efficient outflow boundary condition for the incompressible Navier–Stokes equations

Yibao Li, Jung-II Choi, Yongho Choic & Junseok Kim

Published online: 07 Nov 2016.

Submit your article to this journal

Article views: 74

View related articles

View Crossmark data

Taylor & Francis
Taylor & Francis Group

A simple and efficient outflow boundary condition for the incompressible Navier–Stokes equations

Yibao Li[a], Jung-II Choi[b], Yongho Choi[c] and Junseok Kim[c]

[a] School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, PR China; [b] Department of Computational Science and Engineering, Yonsei University, Seoul, Republic of Korea; [c] Department of Mathematics, Korea University, Seoul, Republic of Korea

ABSTRACT

Many researchers have proposed special treatments for outlet boundary conditions owing to lack of information at the outlet. Among them, the simplest method requires a large enough computational domain to prevent or reduce numerical errors at the boundaries. However, an efficient method generally requires special treatment to overcome the problems raised by the outlet boundary condition used. For example, mass flux is not conserved and the fluid field is not divergence-free at the outlet boundary. Overcoming these problems requires additional computational cost. In this paper, we present a simple and efficient outflow boundary condition for the incompressible Navier–Stokes equations, aiming to reduce the computational domain for simulating flow inside a long channel in the streamwise direction. The proposed outflow boundary condition is based on the transparent equation, where a weak formulation is used. The pressure boundary condition is derived by using the Navier–Stokes equations and the outlet flow boundary condition. In the numerical algorithm, a staggered marker-and-cell grid is used and temporal discretization is based on a projection method. The intermediate velocity boundary condition is consistently adopted to handle the velocity–pressure coupling. Characteristic numerical experiments are presented to demonstrate the robustness and accuracy of the proposed numerical scheme. Furthermore, the agreement of computational results from small and large domains suggests that our proposed outflow boundary condition can significantly reduce computational domain sizes.

1. Introduction

Microfluidics, which is mostly in a low Reynolds number flow regime according to its velocity and length scales, offers fundamentally new capabilities for the control of concentrations of molecules in space and time (Whitesides, 2006). There is also much research into sophisticated three-dimensional hydrodynamics and pollutant modelling (Chau & Jiang, 2001, 2004; Epely-Chauvin, De Cesare, & Schwindt, 2014; Gholami, Akbar, Minatour, Bonakdari, & Javadi, 2014; Liu & Yang, 2014; Wu & Chau, 2006). Thus, microfluidics has become important for many lab-on-a-chip applications such as enzymatic kinetics (Pabit & Hagen, 2002), protein folding (Gambin, Simonnet, VanDelinder, Deniz, & Groisman, 2010), single molecule detection (De Mello & Edel, 2007), and flow cytometry (Mao, Lin, Dong, & Huang, 2009; Wang et al., 2005; Yun et al., 2010). The aspect ratio of a typical micro-channel is very large. Therefore, introducing proper inflow and outflow boundary conditions in numerical computations allows us to use a reduced

computational domain for simulating flow inside such a long channel in the streamwise direction. For the inlet boundary conditions, Han, Lu, and Bao (1994) performed a comparative study with different types of boundary conditions at the inlet and found that a parabolic profile is simple but suitable for describing the inflow conditions. However, many researchers (Dong, Karniadakis, & Chryssostomidis, 2014; Guermond & Shen, 2003; Hagstrom, 1991; Halpern, 1986; Halpern & Schatzman, 1989; Hasan, Anwer, & Sanghi, 2005; Hedstrom, 1979; Jin & Braza, 1993; Johansson, 1993; Jordan, 2014; Kim & Moin, 1985; Kirkpatrick & Armfield, 2008; Lei, Fedosov, & Karniadakis, 2011; Liu, 2009; Mateescu, Munoz, & Scholz, 2010; Ol'shanskii & Staroverov, 2000; Poux, Glockner, Ahusborde, & Azaïez, 2012; Poux, Glockner, & Azaïez, 2011; Regulagadda, Naterer, & Dincer, 2011; Rudy & Strikwerda, 1980; Sani & Gresho, 1994; Sani, Shen, Pironneau, & Gresho, 2006; Sohankar, Norberg, & Davidson, 1998; Thompson, 1987) have proposed special treatments for outlet boundary

conditions owing to the lack of information at the outlet. Four different types of boundary conditions – Dirichlet (Han et al., 1994), Neumann (Kim & Moin, 1985; Kirkpatrick & Armfield, 2008), convective (Ol'shanskii & Staroverov, 2000), and nonreflecting (Jin & Braza, 1993) – are widely applied to describe flow conditions at the outlet.

Dirichlet and Neumann boundary conditions have been widely used for outflow conditions. A Dirichlet boundary condition with a velocity profile (e.g. the Poiseuille profile in a channel) requires pre-knowledge of the flow conditions at the outlet for specifying the velocity profile explicitly, while this naturally provides a global mass flux balance. A Neumann boundary condition does not need pre-knowledge of the flow conditions; however, it requires a large enough computational domain to prevent or reduce numerical errors at the boundaries. Because of this requirement, additional computational cost is expected for applying a Neumann boundary condition. To overcome the limitations of Dirichlet and Neumann boundary conditions at the outlet, a convective boundary condition is often used for the outflow boundary condition. Ol'shanskii and Staroverov (2000) proposed a convective boundary condition that is based on a first-order advection equation (Halpern & Schatzman, 1989) and a scaling method (Hagstrom, 1991). Typically, the advection velocity is assumed to be a constant or have a Poiseuille profile, but is somewhat arbitrary. Therefore, the advection velocity should be supplemented by a correction velocity at each time step to conserve mass flux. Nevertheless, once mass flux has been kept as a constant, the convective boundary condition becomes a fixed Dirichlet boundary condition. Another effective method derived from a wave equation is the nonreflecting boundary method (Hedstrom, 1979; Jin & Braza, 1993; Johansson, 1993; Rudy & Strikwerda, 1980; Sani & Gresho, 1994; Thompson, 1987), which performs well to minimize the spurious artifacts at the outlet. This kind of model is suitable for wake and jet flow with moderate and high Reynolds numbers. Recently, Dong et al. (2014) proposed an efficient outflow boundary condition that can allow for the influx of kinetic energy into the domain but prevent uncontrolled growth in the energy of the domain. To decouple the pressure and velocity computation, they developed a rotational velocity-correction type strategy. The proposed method is efficient and can maximize the domain truncation without adversely affecting the flow physics and enable simulations even with much higher Reynolds number.

In this paper, we propose an efficient and simple transparent boundary condition to simulate channel-type flows efficiently without trivial boundary treatment, especially for low Reynolds number. A pressure boundary condition is also proposed to match with the Navier–Stokes equation. In the numerical algorithm, a staggered marker-and-cell grid is used and temporal discretization is based on a projection method. The intermediate velocity boundary condition is consistently adopted for handling the velocity–pressure coupling. Characteristic numerical experiments are presented to demonstrate the robustness and accuracy of the numerical scheme. Furthermore, the agreement of computational results from small and large domains suggests that our proposed outflow boundary condition can significantly reduce computational domain sizes.

The paper is organized as follows: in Section 2, velocity and pressure boundary conditions for an incompressible Navier–Stokes equation are presented. Section 3 outlines the numerical method. Section 4 provides a variety of results. Finally, our conclusions are given in Section 5.

2. Governing equations and boundary conditions

The non-dimensional Navier–Stokes equation and the continuity equation for time-dependent incompressible viscous flows can be written as

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{\text{Re}} \Delta \mathbf{u}, \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{2}$$

where $\text{Re} = \rho U L / \mu$ is the Reynolds number, ρ and μ are fluid density and viscosity, respectively, U and L are characteristic velocity and length scales, respectively, $p(\mathbf{x}, t)$ is the pressure, $\mathbf{u}(\mathbf{x}, t) = (u(\mathbf{x}, t), v(\mathbf{x}, t))$ is the velocity, where u and v are its components in the x- and y-directions, respectively, $\mathbf{x} = (x, y)$ are Cartesian coordinates on the domain $\Omega \subset \mathbf{R}^2$, and t is the temporal variable. The domain boundary consists of $\partial \Omega = \partial \Omega_{wall} \bigcup \partial \Omega_{in} \bigcup \partial \Omega_{out}$, where $\partial \Omega_{wall}$, $\partial \Omega_{in}$, and $\partial \Omega_{out}$ are the wall, inlet, and outlet boundaries, respectively. Here, the velocities on the boundary are defined as

$$\mathbf{u}|_{\partial \Omega} = \begin{cases} \mathbf{u}|_{\partial \Omega_{wall}} & \text{is zero,} \\ \mathbf{u}|_{\partial \Omega_{in}} & \text{is a given function,} \\ \mathbf{u}|_{\partial \Omega_{out}} & \text{is computed by an outflow} \\ & \quad \text{boundary condition.} \end{cases}$$

Figure 1 shows a schematic representation of the inflow–outflow conditions. For simplicity of exposition, we shall describe the inlet and outlet in the x-direction. The other directions can be similarly defined.
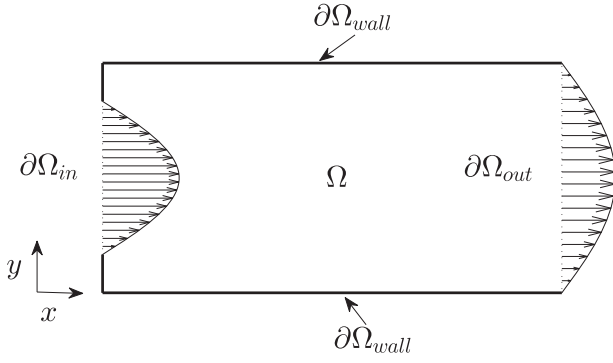
Figure 1. Schematic illustration of inflow–outflow boundary conditions.

2.1. Velocity boundary conditions

At the inlet, the boundary condition can be chosen based on the physical problem. In our paper, we consider inflow parallel to the wall. Thus for $\partial\Omega_{in}$, we use the following boundary condition:

$$u = \hat{u}(y, t) \quad \text{and} \quad v = 0, \tag{3}$$

where $\hat{u}(y, t)$ is a given x-directional inflow velocity function. Jin and Braza (1993) developed a nonreflecting outlet boundary condition for incompressible unsteady Navier–Stokes calculations:

$$u_t + uu_x - \frac{1}{\text{Re}}u_{yy} = 0 \quad \text{and} \quad v_t + uv_x - \frac{1}{\text{Re}}v_{yy} = 0. \tag{4}$$

Halpern and Schatzman (1989) proposed the convective boundary conditions

$$u_t + \mathbf{U} \cdot \nabla u = 0, \tag{5}$$

where $\mathbf{U}(\mathbf{x}, t) = (U(\mathbf{x}, t), V(\mathbf{x}, t))$ is a known flow. Ol'shanskii and Staroverov (2000) proposed a simple convective boundary condition:

$$u_t + Uu_x = 0 \quad \text{and} \quad v = 0. \tag{6}$$

The computed velocity U is supplemented by a correction velocity to balance the mass flux, i.e.

$$\int_{\partial\Omega_{in}} u \, \mathrm{d}y = \int_{\partial\Omega_{out}} u \, \mathrm{d}y.$$

The function U is formally chosen as constant or as a Poiseuille profile. If the outlet flow is unknown, the velocity U can be calculated by taking the integral of both sides of Equation (6):

$$U = \frac{\int_{\partial\Omega_{out}} u_t \, \mathrm{d}y}{\int_{\partial\Omega_{out}} u_x \, \mathrm{d}y} = \frac{\left(\int_{\partial\Omega_{out}} u \, \mathrm{d}y\right)_t}{\int_{\partial\Omega_{out}} u_x \, \mathrm{d}y} = \frac{\left(\int_{\partial\Omega_{in}} u \, \mathrm{d}y\right)_t}{\int_{\partial\Omega_{out}} u_x \, \mathrm{d}y}. \tag{7}$$

However, when the inflow is a steady flow, the constant flow rate at the inlet can lead to a zero velocity U at the outlet. Thus, the outflow boundary condition, Equation (6), will reduce to a Dirichlet boundary condition, i.e. $u_t = 0$. It should be noted that, for unsteady flow, the outflow is clearly not constant. Sohankar et al. (1998) also proposed to set U as a convective velocity but to add a correction velocity to guarantee the balance between the inlet and outlet at each time step. The correction velocity is a constant over the entire outlet and is zero when the outflow reaches a steady state. However, the choice of the convective velocity is somewhat arbitrary. Note that, in this approach, to compare with other boundary conditions, we will perform several tests by using a convective boundary condition. The chosen convective velocity U is a constant Poiseuille profile when the outflow is in the steady state; otherwise, it will be computed by using Equation (7).

The fluid flow at low Reynolds number is generally characterized by smooth and constant fluid motion because viscous forces are dominant. Therefore, we can assume that the flow is only moved by the convection and that its main direction is normal to the outflow boundary. The first assumption implies that the transparent boundary condition $\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u} = 0$ is satisfied and the second assumption implies $v = 0$ because the outflow boundary is perpendicular to the x-direction in our paper. Therefore, to simulate the fluid flow at low Reynolds number without trivial boundary treatment, we propose to use a simple and efficient transparent boundary condition:

$$u_t + uu_x = 0 \quad \text{and} \quad v = 0. \tag{8}$$

These conditions are admissible provided that the outlet boundary is not located quite nearby the downstream. Note that in solving the incompressible Navier–Stokes equations at the inflow and outflow boundaries, both the velocity-divergence and the velocity–pressure forms should be considered to balance the mass flux. This is also a new aspect of this paper. The form (8) may suffer from some numerical difficulties, which will be described in Section 3.1. Thus, by using the divergence-free condition $\nabla \cdot \mathbf{u} = 0$, we can rewrite Equation (8) as

$$u_t + uu_x = u_t + uu_x + u(u_x + v_y)$$
$$= u_t + (u^2)_x = 0 \quad \text{and} \quad v = 0. \tag{9}$$

2.2. Pressure boundary conditions

Taking the inner product of both sides of Equation (1) with $\mathbf{n}$, which is the unit normal vector to the domain boundary, we obtain

$$\mathbf{n} \cdot \nabla p = \mathbf{n} \cdot \left(-\mathbf{u}_t - \mathbf{u} \cdot \nabla \mathbf{u} + \frac{1}{\text{Re}} \Delta \mathbf{u} \right). \qquad (10)$$

On $\partial \Omega_{wall}$, we get

$$\mathbf{n} \cdot \nabla p = \frac{\mathbf{n} \cdot \Delta \mathbf{u}}{\text{Re}}. \qquad (11)$$

Here we have used the no-slip boundary condition, i.e. $\mathbf{u} = 0$. On $\partial \Omega_{in}$, by using Equation (10), we get

$$p_x = -u_t - u u_x - v u_y + \frac{1}{\text{Re}} (u_{xx} + u_{yy})$$

$$= -\hat{u}_t + \frac{1}{\text{Re}} \hat{u}_{yy}. \qquad (12)$$

Here, we have used the inflow boundary condition (3). For the outlet boundary, by substituting Equation (8) into Equation (10), we get

$$p_x = \frac{1}{\text{Re}} (u_{xx} + u_{yy}). \qquad (13)$$

Thus, the pressure boundary condition is directly derived by using the Navier–Stokes equation and outlet flow boundary conditions. The pressure condition is mainly balanced with the viscous stress terms at the boundary. All the pressure boundary conditions are of Neumann type. Thus, the method used with the projection method would be convenient and simple. It is noteworthy that the present form of the pressure boundary condition is more plausible because the flows in microfluidics are mostly viscous dominated. For high Reynolds number flows, the pressure boundary condition becomes a homogeneous Neumann boundary condition. This is because the viscous term $(u_{xx} + u_{yy})/\text{Re}$ is negligible.

3. Numerical methods

We use a staggered grid, where the pressure field is stored at cell centres and velocities at cell interfaces (see Figure 2). Let h be a uniform mesh spacing. The centre of each cell, Ω_{ij}, is located at $(x_i, y_j) = ((i - 0.5)h, (j - 0.5)h)$ on the computational domain $\Omega = (0, \ L_x) \times (0, \ L_y)$, for $i = 1, \ldots, N_x$ and $j = 1, \ldots, N_y$. N_x and N_y are the numbers of cells in the x- and y-directions, respectively. The cell edges are located at $(x_{i+\frac{1}{2}}, y_j) = (ih, (j - 0.5)h)$ and $(x_i, y_{j+\frac{1}{2}}) = ((i - 0.5)h, jh)$.

At the beginning of each time step, given $\mathbf{u}^n$, we want to find $\mathbf{u}^{n+1}$ and p^{n+1} by solving the following
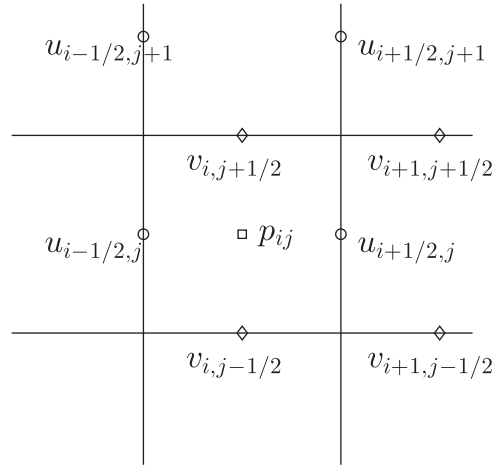

Figure 2. Velocities defined at cell boundaries and the pressure field defined at the cell centres.

discretized equations in time with a projection method (Chorin, 1968):

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = -(\mathbf{u} \cdot \nabla_d \mathbf{u})^n - \nabla_d p^{n+1} + \frac{1}{\text{Re}} \Delta_d \mathbf{u}^n, \qquad (14)$$

$$\nabla_d \cdot \mathbf{u}^{n+1} = 0. \qquad (15)$$

Here, ∇_d and Δ_d denote the centred difference approximations for the gradient and Laplacian operators, respectively:

$$\nabla_d p = \left(\frac{p_{i+1,j} - p_{i,j}}{h}, \frac{p_{i,j+1} - p_{i,j}}{h} \right),$$

$$\Delta_d \mathbf{u} = (\Delta_d u_{i+\frac{1}{2},j}, \Delta_d v_{i,j+\frac{1}{2}})$$

$$= \left(\frac{u_{i+\frac{3}{2},j} + u_{i-\frac{1}{2},j} + u_{i+\frac{1}{2},j+1} + u_{i+\frac{1}{2},j-1} - 4u_{i+\frac{1}{2},j}}{h^2}, \right.$$

$$\left. \frac{v_{i,j+\frac{3}{2}} + v_{i,j-\frac{1}{2}} + v_{i+1,j+\frac{1}{2}} + v_{i-1,j+\frac{1}{2}} - 4v_{i,j+\frac{1}{2}}}{h^2} \right).$$

The outline of the main procedure in one time step is as follows.

Step 1. Initialize $\mathbf{u}^0$ to be the divergence-free velocity field.

Step 2. Update $\mathbf{u}^{n+1}$ and $\mathbf{n} \cdot \nabla_d p^{n+1}$ on the entire boundaries. The computational details will be given in Section 3.1.

Step 3. Solve an intermediate velocity field $\tilde{\mathbf{u}}$ without the pressure gradient term,

$$\frac{\tilde{\mathbf{u}} - \mathbf{u}^n}{\Delta t} = -\mathbf{u}^n \cdot \nabla_d \mathbf{u}^n + \frac{1}{\text{Re}} \Delta_d \mathbf{u}^n. \qquad (16)$$

Step 4. Solve the Poisson equation for the pressure,

$$\Delta_d p^{n+1} = \frac{1}{\Delta t} \nabla_d \cdot \tilde{\mathbf{u}}. \qquad (17)$$

The resulting linear system of Equation (17) is solved using a multigrid method. Note that Equation (17) is derived by applying the divergence operator to Equation (18):

$$\frac{\mathbf{u}^{n+1} - \tilde{\mathbf{u}}}{\Delta t} = -\nabla_d p^{n+1}. \qquad (18)$$

It should be pointed out that, to find the solution for pressure in Equation (17), we should know the boundary condition of the intermediate velocity. Our proposed intermediate velocity boundary condition is described in Section 3.2.

Step 5. Update the divergence-free velocity,

$$\mathbf{u}^{n+1} = \tilde{\mathbf{u}} - \Delta t \nabla_d p^{n+1}. \qquad (19)$$

These steps complete one time step. For more detail, see Li, Jung, Lee, Lee, and Kim (2012). In this study, we focus on introducing efficient outflow boundary conditions and pressure boundary conditions.

3.1. *Boundary conditions for velocity and pressure gradient*

In this section, we will give detailed descriptions of boundary conditions for the velocity and pressure gradient on the entire boundary. First, we will consider the velocity boundary condition. The velocities u^{n+1} on $\partial\Omega_{in}$ and $\partial\Omega_{wall}$ are given as a known function and zero, respectively. The outflow boundary condition Equation (8) can be discretized with an upwind scheme as

$$u^{n+1}_{N_x+\frac{1}{2},j} = u^n_{N_x+\frac{1}{2},j} - \Delta t u^n_{N_x+\frac{1}{2},j} \bar{u}^n_{x_{N_x+\frac{1}{2},j}}. \qquad (20)$$

Here $\bar{u}^n_{x_{i+\frac{1}{2},j}}$ is computed using the upwind procedure:

$$\bar{u}^n_{x_{i+\frac{1}{2},j}} = \begin{cases} \dfrac{u^n_{i+\frac{1}{2},j} - u^n_{i-\frac{1}{2},j}}{h} & \text{if } u^n_{i+\frac{1}{2},j} > 0, \\ \dfrac{u^n_{i+\frac{3}{2},j} - u^n_{i+\frac{1}{2},j}}{h} & \text{otherwise.} \end{cases}$$

Note that Equation (19) yields $u^{n+1}_{N_x+\frac{1}{2},j} = 0$ when $u^n_{N_x+\frac{1}{2},j} = \bar{u}^n_{y_{N_x+\frac{1}{2},j}} = 0$ for every $j = 1, ..., N_y$. This means that, once u is zero at the boundary, u will always be zero. To avoid this trivial case, we use the conservation form

of the convective terms in Equation (9). The discretized form can be written as

$$u^{n+1}_{N_x+\frac{1}{2},j} = u^n_{N_x+\frac{1}{2},j} - \frac{\Delta t}{h}\left[(u^n_{N_x+\frac{1}{2},j})^2 - (u^n_{N_x-\frac{1}{2},j})^2\right], \qquad (21)$$

where we used $v=0$ on the outflow boundary condition. For the pressure boundary condition, we discretize Equation (12) on $\partial\Omega_{in}$ as

$$P^{n+1}_{x_{\frac{1}{2},j}} = -\frac{u^{n+1}_{\frac{1}{2},j} - u^n_{\frac{1}{2},j}}{\Delta t} + \frac{u^n_{\frac{1}{2},j+1} + u^n_{\frac{1}{2},j-1} - 2u^n_{\frac{1}{2},j}}{h^2 \mathrm{Re}}. \qquad (22)$$

At the wall, without loss of generality, we only describe the pressure in the y-direction. Discretizing Equation (11), we can get

$$P^{n+1}_{y_{i,\frac{1}{2}}} = \frac{v^n_{i,-\frac{1}{2}} + v^n_{i,\frac{3}{2}} - 2v^n_{i,\frac{1}{2}}}{h^2 \mathrm{Re}} + \frac{v^n_{i+1,\frac{1}{2}} + v^n_{i-1,\frac{1}{2}} - 2v^n_{i,\frac{1}{2}}}{h^2 \mathrm{Re}}. \qquad (23)$$

Since the no-slip boundary condition is used at the wall, we can get $v_{i,1/2} = 0$. Meanwhile, on the staggered grid, $v^n_{i,-1/2}$ can be simply computed as $-v^n_{i,3/2}$ to match the no-slip boundary condition. Thus in our approach, Equation (11) can be reduced to

$$\mathbf{n} \cdot \nabla p^{n+1} = 0.$$

For $\partial\Omega_{out}$, the pressure boundary condition can be discretized as

$$P^{n+1}_{x_{N_x+\frac{1}{2},j}} = \frac{u^n_{N_x+\frac{1}{2},j+1} + u^n_{N_x+\frac{1}{2},j-1} - 2u^n_{N_x+\frac{1}{2},j}}{h^2 \mathrm{Re}}$$
$$+ \frac{2u^n_{N_x+\frac{1}{2},j} - 5u^n_{N_x-\frac{1}{2},j} + 4u^n_{N_x-\frac{3}{2},j} - u^n_{N_x-\frac{5}{2},j}}{h^2 \mathrm{Re}}. \qquad (24)$$

Note that, since there is no information for $u_{N_x+\frac{3}{2},j}$, we use a second-order one-sided approximation to u_{xx}.

3.2. *Boundary condition for intermediate velocity*

Since we have used the projection method to discretize the time-dependent incompressible Navier–Stokes equation in time, it is necessary to define the boundary condition for the intermediate velocity field. In our approach, we can directly compute $\mathbf{u}^*$. Since we have calculated $\mathbf{u}^{n+1}$ and $\mathbf{n} \cdot \nabla_d p$ in *Step* 2 of Section 3, we can use

them straightforwardly. From Equation (19), we have

$$\tilde{\mathbf{u}} = \mathbf{u}^{n+1} + \Delta t \nabla p^{n+1}. \tag{25}$$

On $\partial\Omega_{wall}$, we get

$$\mathbf{n} \cdot \tilde{\mathbf{u}} = \mathbf{n} \cdot (\mathbf{u}^{n+1} + \Delta t \nabla_d p^{n+1}) = 0. \tag{26}$$

For the inflow boundary, by recalling Equation (19), we have

$$\tilde{u}_{\frac{1}{2},j} = u^{n+1}_{\frac{1}{2},j} + \Delta t p^{n+1}_{x_{\frac{1}{2},j}}$$

$$= u^n_{\frac{1}{2},j} + \frac{\Delta t}{h^2 \text{Re}} \left(u^n_{\frac{1}{2},j+1} + u^n_{\frac{1}{2},j-1} - 2u^n_{\frac{1}{2},j} \right). \tag{27}$$

Here, we have used the Dirichlet boundary condition for velocity and the homogeneous Neumann boundary condition for pressure. On the outflow boundary, we obtain

$$\tilde{u}_{N_x+\frac{1}{2},j} = u^{n+1}_{N_x+\frac{1}{2},j} + \Delta t p^{n+1}_{x_{N_x+\frac{1}{2},j}}$$

$$= u^{n+1}_{N_x+\frac{1}{2},j} + \Delta t \left(u^n_{N_x+\frac{1}{2},j+1} + u^n_{N_x+\frac{1}{2},j-1} \right.$$

$$\left. -2u^n_{N_x+\frac{1}{2},j} \right) / (h^2 \text{Re}) + \Delta t \left(2u^n_{N_x+\frac{1}{2},j} \right.$$

$$\left. -5u^n_{N_x-\frac{1}{2},j} + 4u^n_{N_x-\frac{3}{2},j} - u^n_{N_x-\frac{5}{2},j} \right) / (h^2 \text{Re}). \tag{28}$$

Thus, the intermediate velocity boundary condition is consistently adopted for the velocity–pressure coupling. With different projection methods, the gradient pressure term ($\nabla_d p$) is added (Kim, 2005; Kim & Moin, 1985; Poux, Glockner, & Azaïez, 2011; Tseng & Huang, 2014) or subtracted (Chorin, 1968; Li, Jung, Lee, Lee, & Kim, 2012; Li, Yun, Lee, Shin, Jeong, & Kim, 2013). Our proposed method for the intermediate velocity will work well if $\nabla_d p$ is subtracted. For the other case, the intermediate velocity boundary condition should be modified and can be derived by using the intermediate pressure boundary condition in a similar way. Next, we will briefly describe the extension of our proposed method to an implicit scheme for the momentum equation (1):

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = -(\mathbf{u} \cdot \nabla_d \mathbf{u})^n - \nabla_d p^{n+1} + \frac{1}{\text{Re}} \Delta_d \mathbf{u}^{n+1}. \tag{29}$$

Then we decompose Equation (29) into two steps:

$$\frac{\tilde{\mathbf{u}} - \mathbf{u}^n}{\Delta t} = -(\mathbf{u} \cdot \nabla_d \mathbf{u})^n + \frac{1}{\text{Re}} \Delta_d \tilde{\mathbf{u}}, \tag{30}$$

$$\frac{\mathbf{u}^{n+1} - \tilde{\mathbf{u}}}{\Delta t} = -\nabla_d p^{n+1}. \tag{31}$$

Since the boundary conditions for velocity and pressure gradient are known in Section 3.2, we can straightforwardly compute the intermediate velocity values in the

domain boundary as

$$\mathbf{n} \cdot \tilde{\mathbf{u}} = \mathbf{n} \cdot \mathbf{u}^{n+1} + \Delta t \mathbf{n} \cdot \nabla_d p^{n+1}. \tag{32}$$

However, in that case the time step may be restricted by numerical stability, since explicit time integration for the diffusion term is considered on the boundary. One possible way to improve numerical stability is to replace $\Delta_d \mathbf{u}^n$ by $\Delta_d \tilde{\mathbf{u}}$ in Equations (22)–(24) and substitute them into Equation (32). After that, we can combine Equations (30) and (32) and find the solutions of the intermediate velocity. Once $\tilde{\mathbf{u}}$ is known, we can straightforwardly compute p^{n+1} and $\mathbf{u}^{n+1}$ by using Equations (17) and (19), respectively. Note that, in this paper, we use Equations (14) and (15) so that we only need to solve Equation (17). However, if the Reynolds number is much smaller, implicit or Crank–Nicolson schemes should be considered. It should also be noted that the projection method on staggered grids has the property that an arbitrary boundary condition for the pressure gradient can be used first, and then a corresponding boundary condition can be provided for the intermediate velocity through *Step* 5. The corresponding boundary condition should certainly make the outflow satisfy its boundary condition.

3.3. Mass flux correction algorithm

Undesirable mass flux loss in a global or local sense is possible because of boundary singularities in a finite difference framework. The proposed outflow conditions will leverage for unexpected mass flux loss by addressing the coupling effect between pressure and velocity at the singular point (see the circle in Figure 3(a)). The centred difference approximation for the velocity in Equations (21) and (28) at the singular point is not a good choice and may lead to flux loss because of oscillation of the values. Furthermore, the present boundary conditions require a transient time to achieve mass balance, which will be discussed in the next section. Thus, we propose a simple correction algorithm to maintain the global mass flux. The idea is summarized as follows.

Step 1. Update the velocity with the outlet boundary condition, $\tilde{u}$,

$$\tilde{u}_{N_x+\frac{1}{2},j} = u^n_{N_x+\frac{1}{2},j} - \Delta t \left((u^n_{N_x+\frac{1}{2},j})^2 \right.$$

$$\left. -(u^n_{N_x-\frac{1}{2},j})^2 \right) / h. \tag{33}$$

Step 2. Set $u^{n+1}_{N_x+\frac{1}{2},j} = \epsilon \tilde{u}_{N_x+\frac{1}{2},j}$ if $\tilde{u}_{N_x+\frac{1}{2},j} > 0$, and set $u^{n+1}_{N_x+\frac{1}{2},j} = \tilde{u}_{N_x+\frac{1}{2},j}$ otherwise. Here, ϵ can be determined from the balance of mass flux
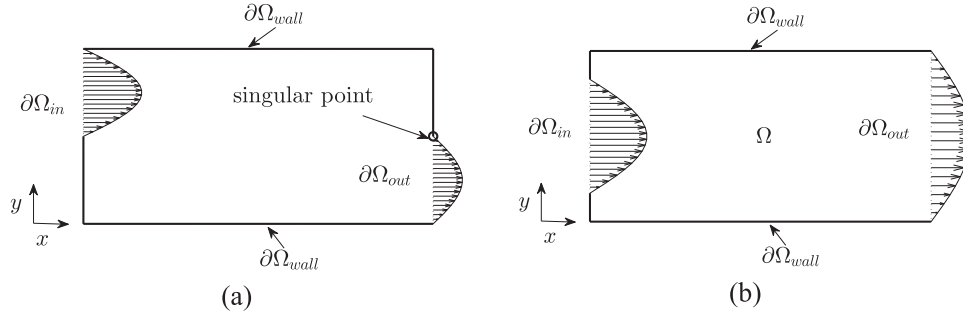
Figure 3. Schematic illustration of channels with (a) and without (b) a singular point in the outlet.

between the inlet and outlet, i.e.

$$\sum_{j=1}^{N_y} u_{\frac{1}{2},j}^{n+1} = \sum_{j=1}^{N_y} u_{N_x+\frac{1}{2},j}^{n+1}. \qquad (34)$$

The following should be pointed out.

(1) Our scheme satisfies the divergence-free condition with the mass flux correction algorithm because Equations (15) and (18) are used in the projection method at every time step.

(2) Since the mass correction method also leads to spurious velocities, our proposed outflow boundary condition does not require it if the outflow is parallel to the wall (see Figure 3 (b)), because in this case the mass flux loss is much smaller, which will be shown in the next section.

(3) In the case of a singular point at the outlet, the correction algorithm is applied after several flow-through time periods t_0, which satisfies $t_0 \int_{\partial\Omega_{in}} u \, dy = \alpha L_x L_y$. Here, α is the flow-through time. In our approach, we set $\alpha = 3$. This is because transient flow fields may promote numerical instability.

(4) In the case without a singular point at the outlet boundary, we can use the mass flux correction algorithm only once to save transient time, i.e. $\alpha = 1$.

4. Numerical results

In this section, to demonstrate the performance of our proposed scheme we perform several numerical experiments for parabolic flow, convergence testing, unsteady flow in a channel, backward-facing step flow, and outflow with a block on the outlet.

4.1. Parabolic flow

The first numerical experiment is performed for parabolic flow through a channel. The channel ranges from 0

to 2 in the streamwise (x) direction and from 0 to 1 in the normal (y) direction. The initial velocity field consists of random perturbations with a maximum amplitude of 2; that is, $u(x, y, 0) = \text{rand}(x, y)$ and $v(x, y, 0) = \text{rand}(x, y)$, where rand($x,y$) is a random number between -1 and 1. The inflow velocity is set as $u(0, y, t) = 8(y - y^2)$ and $v(0, y, t) = 0$. A no-slip boundary condition is applied at the top and bottom walls; i.e. $(u, v) = (0, 0)$. It is well known that, at the beginning of a random velocity field, the wake gradually attains a steady state for a low Reynolds number. At steady state, the shape of the flow is the same as that of the inflow. The evolution is run up to $T = 7.813$ with a time step of $\Delta t = 4h^2$. Here $h = 1/64$ and Re $= 100$ are used. Note that, although $\mathbf{u}^0$ does not satisfy the divergence-free condition, it converges rather quickly to a divergence-free velocity field in five time steps. It should be noted that, although when we begin with $\mathbf{u}^0$ the projection method used can obtain a divergence-free velocity field in one time step, it will take many more iterations to ensure that the l_2 normal error of divergence of the velocity field is lower than 1E-6. Therefore, to reduce the computational cost, we divided each time step process into five steps with several iterations. Figure 4 shows the evolution of parabolic flow. The results show that our method performs well for simulating parabolic flow. To compare the velocity at the outlet, in Figure 5(a), we show the results obtained by our proposed models with and without a flux correction algorithm, along with those of the convective model and the nonreflecting model. Here U is set as a Poiseuille profile, since the outlet flow is known in this test. Furthermore, we calculate the discrete l_2-norm of error, which is

$$\sqrt{\sum_{j=1}^{N_y} (u_{N_x+\frac{1}{2},j} - u_{\frac{1}{2},j})^2 / N_y}.$$

The errors are 8.49E-3, 8.48E-3, 1.07E-3, and 2.26E-2 for our proposed models with and without a flux correction algorithm, the convective model, and the nonreflecting
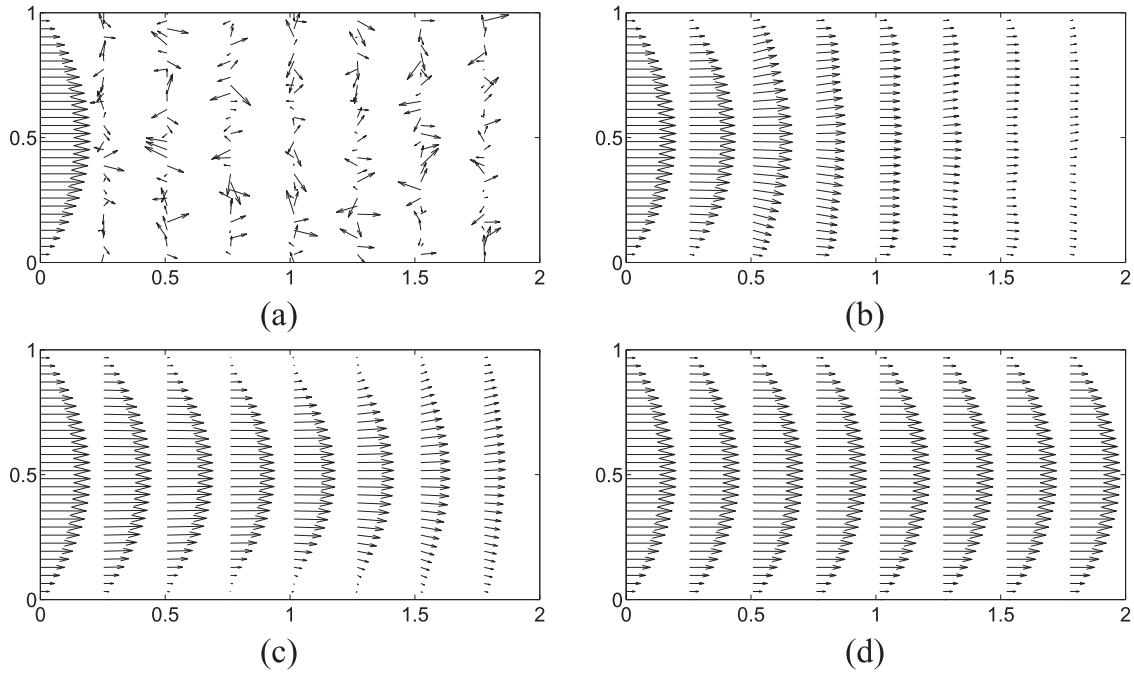
Figure 4. Snapshots of velocity field at times $t = 0$ (a), $t = 0.391$ (b), $t = 1.563$ (c), and $t = 7.813$ (d).
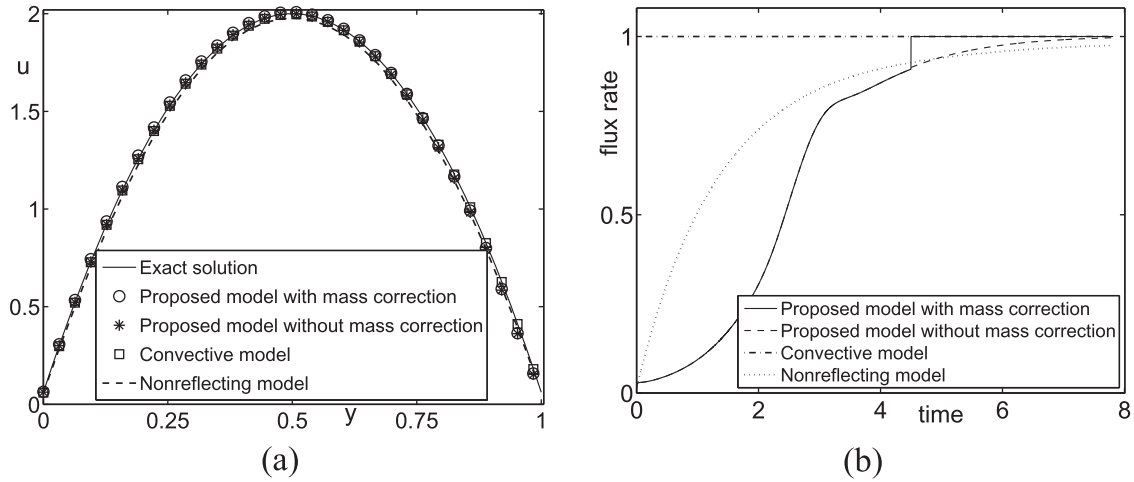

Figure 5. Comparisons of our proposed models with and without a flux correction algorithm to the convective model and the nonreflecting model. (a) Velocity profiles at the outflow boundary position $x = 2$. (b) Flux loss rate, which is computed as the ratio of inlet to outlet. Note the reason for the jump in flux rate: after three flow-through time period intervals, the fluid retains mass flux conservation in our approach. Without a mass flux correction method, the mass flux will be conservative after more flow-through time periods.

model, respectively. These results indicate that the first three models can efficiently emulate parabolic flow. The reason for the failure of the nonreflecting model is that it is derived from a wave equation and not for parabolic flow. In Figure 5(b), we illustrate the flux loss rate, which is computed by the ratio of inlet and outlet fluxes. As can be seen, the convective model and our model satisfy flux conservation, whereas the nonreflecting model does not.

Note that our correction algorithm is performed after

$$t_0 = \alpha L_x L_y \left/ \left(\sum_{j=1}^{N_y} u_{1/2,j} h \right) \right. = 4.5$$

before $\alpha = 3$ is introduced. The jump in flux rate as shown in Figure 5(b) is due to the correction algorithm after three time periods. The numerical jump disappears

later. It should be pointed out that, without the mass correction algorithm, our model also maintains flux conservation after more fluid exits the outlet boundary, because our outflow boundary condition and pressure boundary conditions are defined by matching with the Navier–Stokes equation. Unless otherwise mentioned, we will not use our correction algorithm for the channels, which have no singular points.

4.2. Convergence test

We next perform spatial and temporal convergence tests of the proposed method. To obtain the spatial convergence rate, we perform a number of simulations with increasingly fine grids $h = 1/2^n$ for $n = 5, 6, 7$, and 8 on the domain $\Omega = (0, 2) \times (0, 1)$. The initial condition and parameters are the same as in Section 4.1. It is easy to find that, at equilibrium, the constant pressure gradient equals u_{yy}/Re, i.e. 16/Re. Thus the equilibrium solution of parabolic flow is given by

$$u(x, y) = 8y(1 - y), \quad v(x, y) = 0, \quad \text{and}$$

$$p(x, y) = 16(1 - x)/\text{Re}.$$

Observing the evolution, which is described in Section 4.1, we can see that the outflow reaches a steady state after $t = 7.813$. Thus, in this section, we also run the simulations up to $T = 7.813$. We define the error of the numerical solution on a grid as the discrete l_2-norm of the difference between the numerical solution u_h and the exact solution, u^{exact}: $e_{ij}^h = u_{ij}^h - u_{ij}^{\text{exact}}$. The other terms are defined similarly. The rates of convergence are defined as the ratio of successive errors: $\log_2(\|e^h\|_2/\|e^{h/2}\|_2)$ and $\log_2(\|e^h\|_\infty/\|e^{h/2}\|_\infty)$. Here $\|e\|_2^2$ is a discrete l_2 norm

and is defined as

$$\|e\|_2^2 = \sum_{i=1}^{Nx} \sum_{j=1}^{Ny} e_{ij}^2/(NxNy),$$

and $\|e\|_\infty$ is a discrete l_∞ norm, which is defined as $\|e\|_\infty = \max(|e_{ij}|)$. Since we refined the spatial and temporal grids by factors of 4 and 2, respectively, the ratio of successive errors should increase by a factor of 2. The errors and rates of convergence obtained using these definitions are given in Table 1. The results suggest that the scheme is almost second-order accurate in space, as expected from the discretization.

To obtain the convergence rate for temporal discretization, we fix the space step size as $h = 1/64$ and choose a set of decreasing time steps $\Delta t = 1.953\text{E-}3$, 9.766E-4, 4.883E-4, and 2.441E-4. The errors and rates of convergence obtained using these definitions are given in Table 2. First-order accuracy with respect to time is observed, as expected from the discretization.

4.3. Backward-facing step flow

To examine the robustness and accuracy of the proposed method, we consider a backward-facing step flow. We then compare our results with the previous experimental (Armaly, Durst, & Pereira, 1983) and numerical (Barton, 1997; Kim & Moin, 1985) results.

The backward-facing step geometry with channel dimensions is shown in Figure 6. To verify the robustness of the proposed numerical scheme, we consider an initial condition of a random velocity field on the computational domain $\Omega = (0, 12) \times (0, 1)$. The inflow velocity is $u(0, y, t) = \max(24(1 - y)(y - 0.5), 0)$ and $v(0, y, t) = 0$. These are shown in Figure 7(a). The computations are performed with Re = 500, $h = 1/64$, and $\Delta t = 5h^2$.

Table 1. Error and convergence results with various mesh grids. $\Delta t = 2h^2$ is used.

Case		64 × 32		128 × 64		256 × 128		512 × 256
u	l_2-error	2.75E-2		7.747E-3		2.75E-3		5.607E-4
	Rate		1.88		1.86		1.87	
	l_∞-error	5.900E-2		1.562E-2		4.167E-3		1.150E-3
	Rate		1.92		1.91		1.86	
v	l_2-error	1.311E-3		3.147E-4		7.585E-5		2.162E-5
	Rate		2.06		2.05		1.81	
	l_∞-error	2.292E-3		5.623E-4		1.399E-4		3.915E-5
	Rate		2.03		2.00		1.84	
p	l_2-error	2.424E-2		6.888E-3		1.898E-3		5.163E-4
	Rate		1.81		1.86		1.87	
	l_∞-error	4.438E-2		1.273E-3		3.445E-3		9.438E-4
	Rate		1.80		1.89		1.87	
$\nabla_d \cdot \mathbf{u}$	l_2-error	4.449E-3		1.212E-3		3.335E-4		9.132E-5
	Rate		1.88		1.86		1.87	
	l_∞-error	5.090E-3		1.389E-3		3.827E-4		1.048E-4
	Rate		1.87		1.86		1.87	

Table 2. Error and convergence results with various time steps. 64×128 is fixed.

Case		Δt						
		1.953E-3		9.766E-4		4.883E-4		2.441E-4
u	l_2-error	7.744E-2		3.161E-2		1.598E-2		8.203E-3
	Rate		1.29		0.98		0.96	
	l_∞-error	1.700E-1		6.622E-2		3.262E-2		1.667E-2
	Rate		1.36		1.02		0.97	
v	l_2-error	3.547E-3		1.254E-3		5.982E-4		3.034E-4
	Rate		1.50		1.07		0.98	
	l_∞-error	6.265E-3		2.260E-3		1.094E-3		5.602E-4
	Rate		1.47		1.05		0.97	
p	l_2-error	5.003E-2		2.678E-2		1.459E-2		7.568E-3
	Rate		0.90		0.88		0.95	
	l_∞-error	9.491E-2		4.829E-2		2.660E-2		1.378E-3
	Rate		0.97		0.86		0.95	
$\nabla_d \cdot \mathbf{u}$	l_2-error	6.273E-2		2.571E-2		1.301E-2		6.670E-3
	Rate		1.29		0.98		0.96	
	l_∞-error	7.197E-2		2.950E-2		1.493E-2		7.654E-3
	Rate		1.29		0.98		0.96	
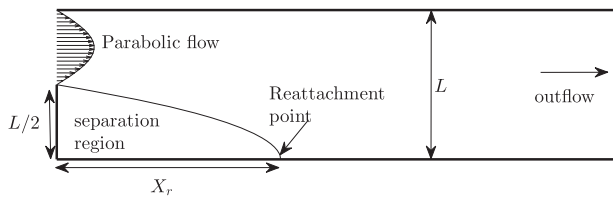
Figure 6. Schematic of a flow over a backward-facing step.

Figures 7(a)–7(d) show snapshots of the velocity field at times $t = 0, 1.465, 2.441$, and 48.828, respectively. As can be seen, our model can simulate backward-facing step flow well.

Plots of reattachment length at steady states with various Reynolds numbers $\mathrm{Re} = 100, 200, \ldots, 800$ are shown in Figure 8(b). Typically, wall shear stress can be used to identify the reattachment location; however, in our computations, the wall shear stress is calculated very nearly but not exactly at the wall. To obtain a more accurate result, we numerically compute the reattachment length X_r^h by using the quadratic polynomial $\alpha = a\beta^2 + b\beta + c$ with three given points, (x_m, y_3), (x_p, y_2), and (x_q, y_1) (see Figure 8(a)). Here $m \in [i, i+1]$, which satisfies $u_{i-1/2,3} \leq 0$ and $u_{i+1/2,3} > 0$ for some integer i. Then x_m is computed by the linear interpolation of nearby two points, i.e. $x_m = (x_{i+1}u_{i-1/2,3} - x_i u_{i+1/2,3})/(u_{i-1/2,3} - u_{i+1/2,3})$. Note that x_p and x_q are defined in a similar fashion. Finally, approximating the quadratic polynomial $\alpha = a\beta^2 + b\beta + c$, we can get $X_r^h = \alpha|_{\beta=0}$. To compare the proposed algorithm to the previous results, we put
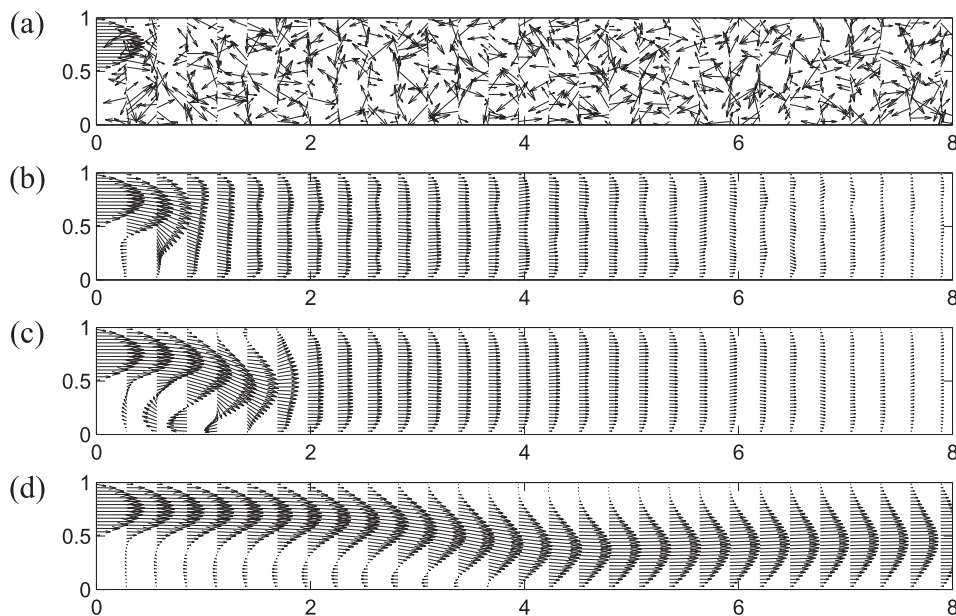
Figure 7. Snapshots of the velocity field at times $t=0$ (a), $t=1.465$ (b), $t=2.441$ (c), and $t=48.828$ (d).
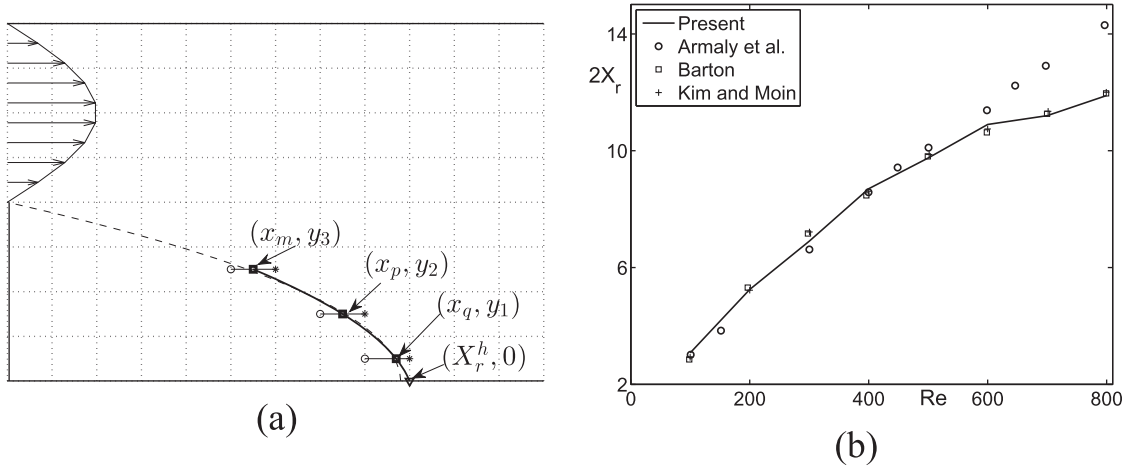
Figure 8. (a) Schematic of computing the reattachment length X_r^h in the numerical solution. Here, open circle, filled square, and star indicate the positive, zero, and negative values of u, respectively. (b) Reattachment length at steady state with various Reynolds numbers.
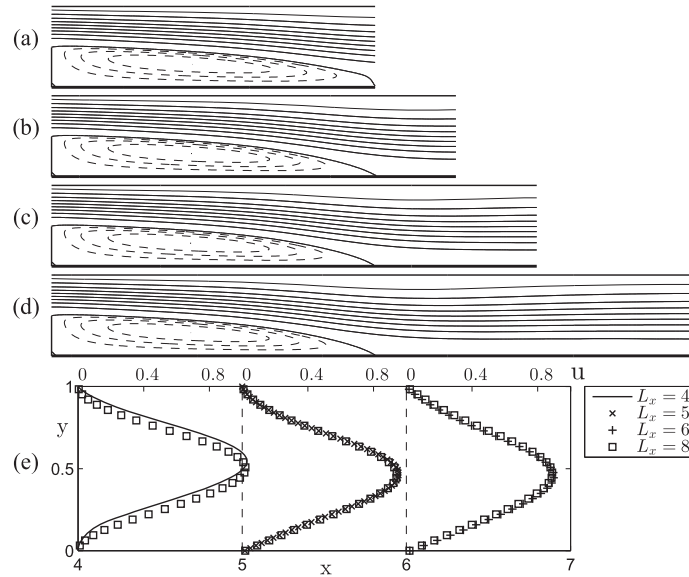

Figure 9. Snapshots of streamlines with four domain sizes: $\Omega = (0,4) \times (0,1)$ (a); $\Omega = (0,5) \times (0,1)$ (b); $\Omega = (0,6) \times (0,1)$ (c); and $\Omega = (0,8) \times (0,1)$ (d). Comparison of velocity with the three cases (e).

them together. From these results, we observe that the results obtained by using our proposed method are in good agreement with those from the previous numerical methods (Barton, 1997; Kim & Moin, 1985) and experiment (Armaly, Durst, & Pereira, 1983).

Figures 9(a)–9(e) are snapshots of streamlines for various domain sizes $\Omega = (0,4) \times (0,1)$, $(0,5) \times (0,1)$, $(0,6) \times (0,1)$, and $(0,8) \times (0,1)$, respectively. Here Re = 400 is used. Figure 9(f) shows a comparison of velocity at the outflow boundary for different domain sizes.

The agreement in our computational results obtained from a larger domain and a smaller domain suggests that our proposed outflow boundary condition is efficient. It should be pointed out that, if the computational

domain is smaller, then the numerical error increases (see Figure 9(a)). This is because our proposed method is simple, we have assumed that the flow only moves by convection, and that its main direction is normal to the outflow boundary.

4.4. Comparison with Neumann boundary condition

In this section, we will compare the results obtained by our proposed and Neumann outflow boundary conditions to show the efficiency and accuracy of our proposed method. The backward-facing step flow drawn
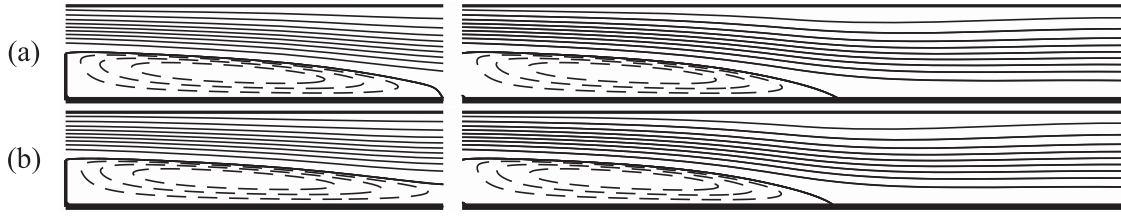
Figure 10. Snapshots of streamlines for the proposed outflow boundary (a) and a Neumann outflow boundary (b). From left to right, they are results for $\Omega = (0, 4) \times (0, 1)$ and $\Omega = (0, 7) \times (0, 1)$, respectively.

Table 3. The l_2-errors with various domain sizes for the proposed outflow boundary and the Neumann outflow boundary.

Case		Domain (Ω)			
		$(0, 4) \times (0, 1)$	$(0, 5) \times (0, 1)$	$(0, 6) \times (0, 1)$	$(0, 7) \times (0, 1)$
u	Neumann	9.871E-2	2.566E-2	1.599E-2	1.299E-2
	Proposed	3.494E-2	1.012E-2	6.010E-3	2.314E-3
v	Neumann	2.315E-2	5.442E-3	3.387E-3	2.796E-3
	Proposed	1.745E-2	4.147E-3	2.188E-3	1.051E-3

in Section 4.3 is considered. The parameters and initial condition are chosen to be the same as in Section 4.3.

The comparisons with the two boundary conditions are drawn in Figure 10. From left to right, they are results for $\Omega = (0, 4) \times (0, 1)$ and $\Omega = (0, 7) \times (0, 1)$, respectively. As can be observed from Figure 10(a), the agreement between the results computed from the larger and smaller domains is good. This suggests that our proposed outflow boundary condition is more efficient than the Neumann boundary condition.

Furthermore, we perform a number of simulations with increasing domains $\Omega = (0, L_x) \times (0, 1)$ for $L_x = 4, 5, 6$, and 7. Since there is no closed-form analytical solution for this problem, we consider a reference numerical solution, which is obtained in a large domain $\Omega = (0, 8) \times (0, 1)$. Therefore, we can compute the l_2 errors of velocities directly for each case. These l_2 errors are given in Table 3. The results suggest that, as the domain size increases, the l_2 errors converge. These results also suggest that our proposed outflow boundary condition is more accurate than the Neumann outflow boundary because the l_2 errors obtained by our proposed outflow boundary condition are much smaller than those obtained by the Neumann outflow boundary.

4.5. Efficiency of the proposed method

To investigate the efficiency of our proposed method, we measure CPU times needed to solve the following problem: $u(x, y, 0) = \text{rand}(x, y)$, $v(x, y, 0) = \text{rand}(x, y)$ on the domain $\Omega = (0, 4) \times (0, 1)$ with $2^{n+2} \times 2^n$ meshes for $n = 4, 5, 6, 7$, and 8. The inflow velocities are $u(0, y, t) = \max(24(1 - y)(y - 0.5), 0)$ and $v(0, y, t) =$
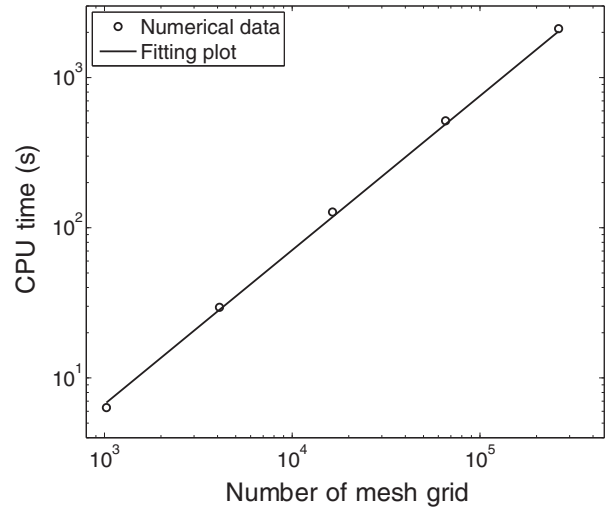

Figure 11. Logarithm of the total CPU time versus number of mesh grid ($N_x N_y$) in multigrid solver.

0. Here, $\text{Re} = 500$ is fixed. All simulations are run up to $1000\Delta t$ with the time step $\Delta t = h^2$. Note that each simulation is run until the maximum error of the residual is less than 10^{-7}. Since we refined the spatial grids by a factor of four, computation times should be increased ideally by a factor of four owing to the discretization (17), which is solved by using a multigrid method, with no special computing operators for the inflow and outflow boundary conditions added. Figure 11 shows the total CPU time versus number of mesh grid ($N_x N_y$). The straight fitting plot implies that the multigrid solver achieves $O(N_x N_y)$ efficiency. Furthermore, the computing time for the boundary conditions can be negligible because our method updates the boundary velocity explicitly. If an implicit update for the boundary condition is chosen, then the computational time is not negligible.

4.6. Three-dimensional flow

Our method can also be straightforwardly extended to three-dimensional flows. Here, we consider a backward-facing step flow with an initial condition of a random velocity field on the computational domain $\Omega = (0, 3) \times (0, 1) \times (0, 1)$. The inflow velocities are $u(0, y, z, t) =$
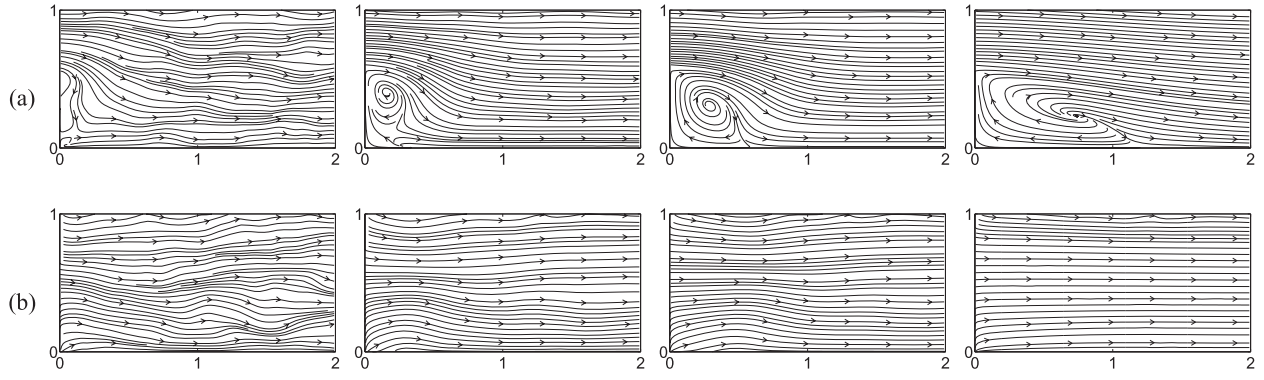
Figure 12. Snapshots of the velocity field in horizontal (a) and vertical (b) views. From left to right, the computational times are $t = 0.098$, 0.293, 0.586, and 2.441.
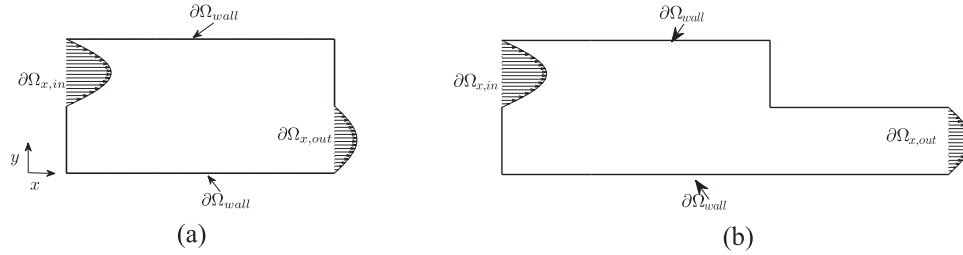

Figure 13. Schematic illustration of inflow–outflow boundary conditions (a). Schematic illustration of inflow–outflow boundary conditions and computing domain for finding the exact solution (b).

max$(24(1 - y)(y - 0.5)(1 - z)(z - 0.5), 0)$, $v(0, y, z, t) = 0$, and $w(0, y, z, t) = 0$. The computations are performed with Re $= 100$, $h = 1/64$, and $\Delta t = h^2$. Figures 12(a) and 12(b) show snapshots of the velocity field in horizontal and vertical views, respectively. From left to right, the computational times are $t = 0.098$, 0.293, 0.586, and 2.441. As can be seen from the figure, our proposed model can simulate three-dimensional flows well.

4.7. Block on the outlet

We now consider the flow geometry shown in Figure 13(a). A fluid enters the upper half of a rectilinear channel at the left domain boundary and exits the lower half of the channel at the right boundary. The inflow velocity is set as $u(0, y, 0) = \max(24(1 - y)(y - 0.5), 0)$ and $v(0, y, 0) = 0$ on $\Omega = (0, L_x) \times (0, 1)$ with $h = 1/64$, $\Delta t = 4h^2$, $L_x = 4$, and Re $= 500$. A block is set at $y > 0.5$ at the outflow boundary. Note that, in this case, there is a singularity point on the outflow boundary. Thus we should use the proposed mass correction method. Since there is no closed-form analytical solution for this problem, we consider a reference numerical solution, which is first obtained in a large domain $\Omega = (0, L_x + 2) \times (0, 1)$, as shown in Figure 13(b), and then in the domain $\Omega = (0, L_x) \times (0, 1)$. The other parameters are chosen to be

the same as in the above initial condition. Figure 14 shows contour plots of the stream functions at $t = 3.052$ and 42.724 for convective, nonreflecting, our proposed model, and a reference numerical solution, respectively. Here, U is computed by using Equation (7), since the outlet flow is unknown in this test. As can be seen, compared with the convective and proposed models, fluids cannot completely go across the outlet boundary by using the nonreflecting model. Since its outflow boundary condition is derived from a wave equation without flux correction, unphysical flow appears for lower Reynolds numbers. Also, the convective model can simulate inflow and outflow when a block is set on the outlet boundary, whereas the outlet flow boundary is reduced to a Dirichlet boundary though the outflow does not reach a steady flow. This is not a naturally occurring physical phenomenon. Compared with them, in our proposed model the outflow seems to be a parabolic-type flow, which is closer to the reference numerical solution. The reason for this is that our model allows the outflow to be adjusted on the basis of the current inflow and inside fluid, whereas the convective model works with a given artificial velocity and mass flux correction method.

Figures 15(a)–15(d) show the contour plot of the stream function with three domain sizes $L_x = 2, 4$, and 6 for convective, nonreflecting, our proposed models,
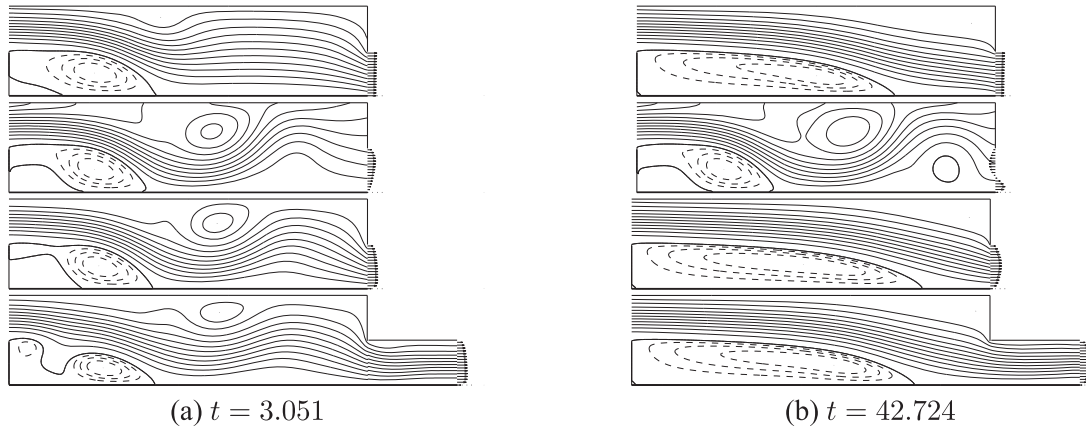
(a) $t = 3.051$ (b) $t = 42.724$

Figure 14. Evolution of outflow with a block obtained by using our proposed model. From top to bottom, they are for convective, non-reflecting, our proposed models, and the reference numerical solution, which is obtained in a larger domain as shown in Figure 13(b), respectively. The computational times are shown for the left and right sides of the figure.
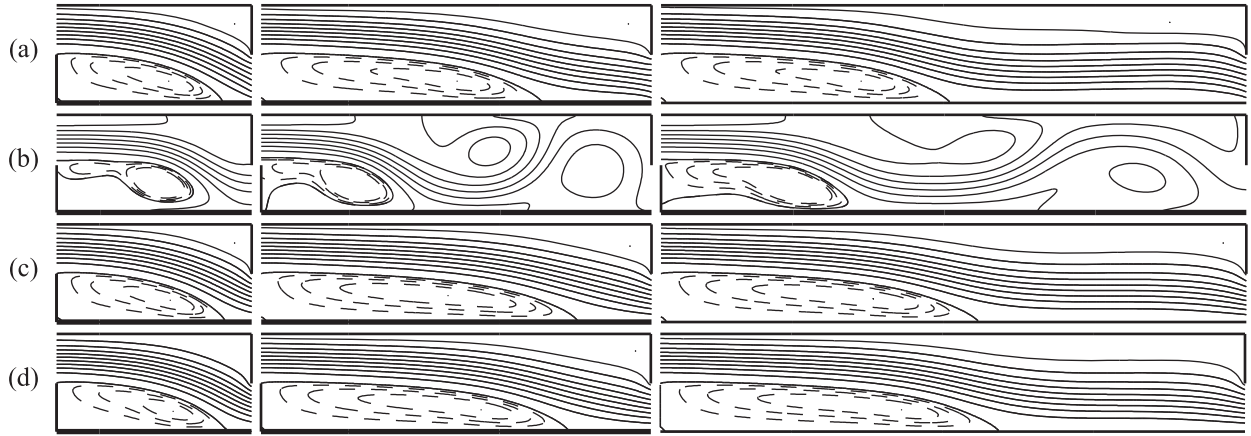

Figure 15. Comparison of three models with different domain sizes at $T = 61.035$. (a) Convective model, (b) nonreflecting model, (c) our proposed model, and (d) the reference numerical solution. From left to right, the domain sizes are $(0, 2) \times (0, 1)$, $(0, 4) \times (0, 1)$, and $(0, 6) \times (0, 1)$.
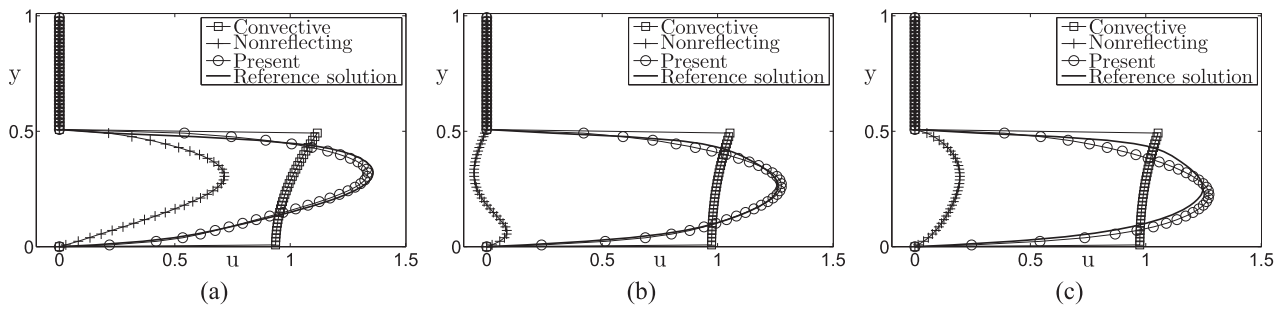

Figure 16. Comparison of the outlet velocity for three models with different domain sizes at $T = 61.035$. To compare them, we put the reference numerical solutions together. (a) $L_x = 2$, (b) $L_x = 4$, and (c) $L_x = 6$.

and the reference numerical solutions, respectively. In Figure 16, we compare the outlet velocity for the four cases with the three domain sizes.

As can be seen, the outflows obtained by using the nonreflecting model lead to flux loss with the increase in the domain size. Meanwhile, with the convective model, the outflow reaches a similar constant outflow profile for different domain sizes. Compared with the nonreflecting and convective models, our proposed model reaches a natural and efficient outflow, which seems to be a parabolic-type flow, and is closer to the reference numerical solution.

5. Conclusions

In this paper, we presented a simple and efficient outflow boundary condition for an incompressible Navier–Stokes equation. The proposed outflow boundary condition is based on the transparent equation, where a weak formulation is used. By matching with the Navier–Stokes equation adopted inside the domain, we exactly presented the pressure boundary condition. Furthermore, we proposed the pressure boundary for convective and nonreflecting models. In the numerical algorithm, a staggered marker-and-cell grid was used and temporal discretization was based on a projection method. The intermediate velocity boundary condition is consistently adopted to handle the velocity–pressure coupling. Although our numerical scheme is an explicit scheme in time and has first-order accuracy in time, we focus on proposing efficient outflow boundary conditions and pressure boundary conditions in this study. The extensions to implicit and Crank–Nicolson schemes are straightforward. Characteristic numerical experiments were presented to demonstrate the accuracy of our proposed model. Although a first-order equation is used on the outlet boundary, almost second-order accuracy is observed for our model in the convergence test, because we use a full second-order scheme to discretize the Navier–Stokes equation inside the domain. Compared with the convective model and our proposed model, the nonreflecting model is not a good choice for low Reynolds number flow and the outflow when a block is on the outlet boundary, because the nonreflecting model cannot maintain mass flux conservation and parabolic flow is not its solution. The convective model is simple and effective satisfying mass flux conservation, although once the mass flux of the inflow becomes constant, the convective model will be reduced to a Dirichlet boundary condition. Compared with these two methods, our proposed method allows the outlet flow to be adjusted on the basis of the current inflow and fluids inside the domain, when the Reynolds number is low. Our proposed method is simple compared to that of Dong et al. (2014). However, this comparison is in some ways unfair, because the method presented in Dong et al. (2014) can maximize the domain truncation without adversely affecting the flow physics even at higher Reynolds numbers, whereas our approach aims to propose an efficient and simple transparent boundary condition to simulate channel-type flows efficiently without trivial boundary treatment at lower Reynolds numbers. When the Reynolds number is high, our method requires a much larger computational domain to reduce numerical errors compared to that of Dong et al. (2014), owing to the simplicity of our proposed transparent boundary condition. In future, we will extend our work to simulate multiphase fluids.

Mass conservation should be satisfied for each fluid separately if the mass flux is to be conserved. Therefore, the inflow and outflow boundaries should be coupled with the phases in that case. In future work, we will investigate efficient inflow and outflow boundaries to conserve the total mass of each fluid.

References

Armaly, B. F., Durst, F., Pereira, J. C. F., & Schonung, B. (1983). Experimental and theoretical investigation of backward-facing flow. *Journal of Fluid Mechanics*, *127*, 473–496. doi:10.1017/S0022112083002839

Barton, I. E (1997). The entrance effect of laminar flow over a backward-facing step geometry. *International Journal for Numerical Methods in Fluids*, *25*, 633–644. doi:10.1002/(SICI)1097-0363(19970930)25:6%3C633::AID-FLD551%3E3.0.CO;2-H

Chau, W. K., & Jiang, Y. W. (2001). 3D numerical model for Pearl river estuary. *Journal of Hydraulic Engineering*, *127*, 72–82. doi:10.1061/(ASCE)0733-9429(2001)127:1(72)

Chau, K. W., & Jiang, Y. W. (2004). A three-dimensional pollutant transport model in orthogonal curvilinear and sigma coordinate system for Pearl river estuary. *International Journal of Environment and Pollution*, *21*, 188–198. doi:10.1504/IJEP.2004.004185

Chorin, J. A. (1968). Numerical solution of the Navier–Stokes equations. *Mathematics of Computation*, *22*, 745–762. doi:10.1090/S0025-5718-1968-0242392-2

De Mello, A. J., & Edel, J. B. (2007). Hydrodynamic focusing in microstructures: Improved detection efficiencies in sub-femtoliter probe volumes. *Journal of Applied Physics*, 101, 084903. doi:10.1063/1.2721752

Dong, S., Karniadakis, E. G., & ssostomidis, C. (2014). A robust and accurate outflow boundary condition for incompressible flow simulations on severely-truncated unbounded domains. *Journal of Computational Physics*, *261*, 83–105. doi:10.1016/j.jcp.2013.12.042

Epely-Chauvin, G., De Cesare, G., & Schwindt, S. (2014). Numerical modelling of plunge pool scour evolution in non-cohesive sediments. *Engineering Applications of Computational Fluid Mechanics*, 8, 477–487. doi:10.1080/19942060.2014.11083301

Gambin, Y., Simonnet, C., VanDelinder, V., Deniz, A., & Groisman, A. (2010). Ultrafast microfluidic mixer with three-dimensional flow focusing for studies of biochemical kinetics. *Lab on a Chip*, 10, 598–609. doi:10.1039/B914174J

Gholami, A., Akbar, A. A., Minatour, Y., Bonakdari, H., & Javadi, A. A. (2014). Experimental and numerical study on velocity fields and water surface profile in a strongly-curved 90 open channel bend. *Engineering Applications of Computational Fluid Mechanics*, 8, 447–461. doi:10.1080/19942060.2014.11015528

Guermond, J. L., & Shen, J. (2003). Velocity-correction projection methods for incompressible flows. *SIAM Journal on Numerical Analysis*, 41, 112–134. doi:10.1137/S0036142901395400

Hagstrom, T. (1991). Conditions at the downstream boundary for simulations of viscous, incompressible flow. *SIAM Journal on Scientific and Statistical Computing*, 12, 843–858. doi:10.1137/0912045

Halpern, L. (1986). Artificial boundary conditions for the linear advection–diffusion equations. *Mathematics of Computation*, 46, 25–438. doi:10.1090/S0025-5718-1986-0829617-8

Halpern, L., & Schatzman, M. (1989). Artificial boundary conditions for incompressible viscous flows. *SIAM Journal on Mathematical Analysis*, 20, 308–353. doi:10.1137/0520021

Han, H., Lu, J., & Bao, W. (1994). A discrete artificial boundary condition for steady incompressible viscous flows in a no-slip channel using a fast iterative method. *Journal of Computational Physics*, 114, 201–208. doi:10.1006/jcph.1994.1160

Hasan, N., Anwer, S. F., & Sanghi, S. (2005). On the outflow boundary condition for external incompressible flows: A new approach. *Journal of Computational Physics*, 206, 661–683. doi:10.1016/j.jcp.2004.12.025

Hedstrom, G. W. (1979). Nonreflecting boundary conditions for nonlinear hyperbolic systems. *Journal of Computational Physics*, 30, 222–237. doi:10.1016/0021-9991(79)90100-1

Jin, G., & Braza, M. (1993). A nonreflecting outlet boundary condition for incompressible unsteady Navier–Stokes calculations. *Journal of Computational Physics*, 107, 239–253. doi:10.1006/jcph.1993.1140

Johansson, V. (1993). Boundary conditions for open boundaries for the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 105, 233–251. doi:10.1006/jcph.1993.1071

Jordan, S. A. (2014). On the axisymmetric turbulent boundary layer growth along long thin circular cylinders. *Journal of Fluids Engineering*, 36, 051202. doi:10.1115/1.4026419

Kim, J. (2005). A continuous surface tension force formulation for diffuse-interface models. *Journal of Computational Physics*, 204, 784–804. doi:10.1115/1.4026419

Kim, J., & Moin, P. (1985). Application of a fractional-step method to incompressible Navier–Stokes equations. *Journal of Computational Physics*, 59, 308–323. doi:10.1016/0021-9991(85)90148-2

Kirkpatrick, M., & Armfield, S. (2008). Open boundary conditions in numerical simulations of unsteady incompressible flow. *ANZIAM Journal*, 50, 5760–5773. doi:10.21914/anziamj.v50i0.1457

Lei, H., Fedosov, D. A., & Karniadakis, G. E. (2011). Time-dependent and outflow boundary conditions for dissipative particle dynamics. *Journal of Computational Physics*, 230, 3765–3779. doi:10.1016/j.jcp.2011.02.003

Li, Y., Jung, E., Lee, W., Lee, H. G., & Kim, J. (2012). Volume preserving immersed boundary methods for two-phase fluid flows. *International Journal for Numerical Methods in Fluids*, 69, 842–858. doi:10.1002/fld.2616

Li, Y., Yun, A., Lee, D., Shin, J., Jeong, D., & Kim, J. (2013). Three-dimensional volume-conserving immersed boundary model for two-phase fluid flows. *Computer Methods in Applied Mechanics and Engineering*, 257, 36–46. doi:10.1016/j.cma.2013.01.009

Liu, J. (2009). Open and traction boundary conditions for the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 228, 7250–7267. doi:10.1016/j.jcp.2009.06.021

Liu, T., & Yang, J. (2014). Three-dimensional computations of water–air flow in a bottom spillway during gate opening. *Engineering Applications of Computational Fluid Mechanics*, 8, 104–115. doi:10.1080/19942060.2014.11015501

Mao, X., Lin, S. C. S., Dong, C., & Huang, T. J. (2009). Tunable liquid gradient refractive index (L-GRIN) lens with two degrees of freedom. *Lab on a Chip*, 9, 1583–1589. doi:10.1039/b822982a

Mateescu, D., Muñoz, M., & Scholz, O. (2010). Analysis of unsteady confined viscous flows with variable inflow velocity and oscillating walls. *ASME Journal of Fluids Engineering*, 132, 041105. doi:10.1115/1.4001184

Ol'shanskii, M. A., & Staroverov, V. M. (2000). On simulation of outflow boundary conditions in finite difference calculations for incompressible fluid. *International Journal for Numerical Methods in Fluids*, 33, 499–534. doi:10.1002/1097-0363(20000630)33:4 < 499::AID-FLD193.0.CO;2-7

Pabit, S. A., & Hagen, S. J. (2002). Laminar-flow fluid mixer for fast fluorescence kinetics studies. *Biophysical Journal*, 83, 2872–2878. Retrieved from http://dx.doi.org/10.1016/S0006-3495(02)75296-X

Poux, A., Glockner, S., Ahusborde, E., & Azaïez, M. (2012). Open boundary conditions for the velocity-correction scheme of the Navier–Stokes equations. *Computers & Fluids*, 70, 29–43. doi:10.1016/j.compfluid.2012.08.028

Poux, A., Glockner, S., & Azaïez, M. (2011). Improvements on open and traction boundary conditions for Navier–Stokes time-splitting methods. *Journal of Computational Physics*, 10, 4011–4027. doi:10.1016/j.jcp.2011.02.024

Regulagadda, P., Naterer, G. F., & Dincer, I. (2011). Transient heat exchanger response with pressure regulated outflow. *Journal of Thermal Science and Engineering Applications*, 3, 021008. doi:10.1115/1.4004009.

Rudy, D. H., & Strikwerda, J. C. (1980). A nonreflecting outflow boundary condition for subsonic Navier–Stokes calculations. *Journal of Computational Physics*, 36, 55–70. doi:10.1016/0021-9991(80)90174-6

Sani, R. L., & Gresho, P. M. (1994). Résumé and remarks on the open boundary condition minisymposium. *Journal for Numerical Methods in Fluids*, 18, 983–1008. doi:10.1002/fld.1650181006

Sani, R. L., Shen, J., Pironneau, O., & Gresho, P. M. (2006). Pressure boundary condition for the time-dependent

incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Fluids*, *50*, 673–682. doi:10.1002/fld.1062

Sohankar, A., Norberg, C., & Davidson, L. (1998). Low-Reynolds number flow around a square cylinder at incidence: Study of blockage, onset of vortex shedding, and open boundary conditions. *International Journal for Numerical Methods in Fluids*, *26*, 39–56. doi:10.1002/(SICI)1097-0363 (19980115)26:1$%$3C39::AID-FLD623$$3E3.0.CO;2-P

Thompson, K. W. (1987). Time dependent boundary conditions for hyperbolic systems. *Journal of Computational Physics*, *68*, 1–24. doi:10.1016/0021-9991(87)90041-6

Tseng, Y. H., & Huang, H. (2014). An immersed boundary method for endocytosis. *Journal of Computational Physics*, *273*, 143–159. doi:10.1016/j.jcp.2014.05.009

Wang, M. M., Tu, E., Raymond, D. E., Yang, J. M., Zhang, H., Hagen, N., & Butler, W. F. (2005). Microfluidic sorting of mammalian cells by optical force switching. *Nature Biotechnology*, *23*, 83–87. doi:10.1038/nbt1050

Whitesides, G. M. (2006). The origins and the future of microfluidics. *Nature*, *442*, 368–373. doi:10.1038/nature05058

Wu, C. L., & Chau, K. W. (2006). Mathematical model of water quality rehabilitation with rainwater utilisation: A case study at Haigang. *International Journal of Environment and Pollution*, *28*, 534–545.

Yun, H., Bang, H., Min, J., Chung, C., Chang, J. K., & Han, D. C. (2010). Simultaneous counting of two subsets of leukocytes using fluorescent silica nanoparticles in a sheathless microchip flow cytometer. *Lab on a Chip*, *10*, 3243–3254. doi:10.1039/c0lc00041h