Direct Simulation Based Model-Predictive Control of Flow Maldistribution in Parallel Microchannels

Mathieu Martin Chris Patton John Schmitt
Sourabh V. Apte*
School of Mechanical, Industrial and Manufacturing Engineering
Oregon State University
Corvallis, OR 97331

July 5, 2009

Abstract

Flow maldistribution, resulting from bubbles or other particulate matter, can lead to drastic performance degradation in devices that employ parallel microchannels for heat transfer. In this work, direct numerical simulations of fluid flow through a prescribed parallel microchannel geometry are performed and coupled with active control of actuated microvalves to effectively identify and reduce flow maldistribution. Accurate simulation of fluid flow through a set of three parallel microchannels is achieved utilizing a fictitious domain representation of immersed objects, such as microvalves and artificially introduced bubbles. Flow simulations are validated against experimental results obtained for flow through a single, high aspect ratio microchannel, flow around an oscillating cylinder, and flow with a bubble rising in an inclined channel. Results of these simulations compare very well to those obtained experimentally, and validate the use of the solver for the parallel microchannel configuration of this study. System identification techniques are employed on numerical simulations of fluid flow through the geometry to produce a lower dimensional model that captures the essential dynamics of the full nonlinear flow, in terms of a relationship between valve angles and the exit flow rate for each channel. A model predictive controller is developed which employs this reduced order model to identify flow maldistribution from exit flow velocities and prescribe actuation of channel valves to effectively redistribute the flow. Flow simulations with active control are subsequently conducted with artificially introduced bubbles. The model predictive control methodology is shown to adequately reduce flow maldistribution by quickly varying channel valves to remove bubbles and equalize flow rates in each channel.

*Corresponding Author: e-mail: `sva@engr.orst.edu`, phone: 541-737-7335, fax: 541-737-2600

1 Introduction

Microchannels are employed in a variety of devices, such as heat sinks and heat exchangers, to improve heat transfer effectiveness. For improved efficiency and cooling of high heat loads, two-phase flows involving convective boiling of high latent heat fluids are often used. Formation of vapor bubbles inside the microchannel geometry can lead to blocking effects, resulting in flow maldistribution with non-uniform spatial and temporal conditions. Such flow maldistribution reduces the effectiveness of the heat transfer, leading to decreased performance of the devices under consideration.

Previous investigations into multiphase flow in microchannels [1, 2] have shown that the vapor phase can take different forms, each yielding qualitatively different flow behavior. For two-phase flow in a square channel, Cubaud *et al.* [3] classify flow regimes by the size and occupation of the channel by the vapor phase. In bubbly flow, small bubbles comprise a vapor phase that is relatively small in comparison to the channel dimensions. In this flow regime, bubbles are mainly spherical and flow with the fluid at a similar velocity. Sharp & Adrian [4] investigated clogging or arching structures inside microchannels or in sharp corners due to interaction of rigid particles. Bubble-laden flows may also lead to similar blocking effects in microchannel geometries.

In microchannel geometries using a liquid as a coolant for removal of high heat fluxes, bubbles may form and grow at nucleation sites. These bubbles remain attached to channel walls due to surface tension effects [5]. Existence of the bubble in a channel affects the fluid flow not only in that channel, but also in all of the connected channels, leading to flow maldistribution. As the bubble grows, eventually the surface tension forces keeping the bubble attached to the wall are lowered; and the hydrodynamic forces acting on the bubble may overcome any resistive forces, detaching and moving the bubble [6, 7]. Formation and growth of bubbles inside parallel microchannel geometries can lead to flow instabilities, flow reversals and can affect the flow distribution in a network of channels [8]. To mitigate these flow instabilities, Mukherjee *et al.* [8] proposed use of a variable size microchannels with increased cross-sectional areas in the downstream direction. Increasing areas in the downstream directions result in higher pressures locally, which tend to reduce the flow reversal, and is an example of a *passive control* to mitigate flow distribution problems. Active control, based on model-predictive control algorithm, has been used by Bleris *et al.* [9], in order to improve mixing of chemical species in parallel microchannels. By regulating the mass-flow rates in each channel, they showed how active control can be used to improve mixing and chemical reaction processes.

In the present work, we investigate an active control strategy as applied to two-phase flow in parallel microchannels in order to mitigate flow maldistributions due to presence of single or multiple bubbles in one or more branches of the channels. We consider the bubbly flow regime with one or more bubbles present in a parallel microchannel configuration as shown in Fig. 1. As the first step, the bubbles are modeled as rigid spherical particles in a two-dimensional simulation. Bubble growth is not simulated; instead, bubbles are injected into the domain close to a microchannel wall. Bubbles are held fixed until the hydrodynamic force applied by the flow reaches a critical threshold, at which point the bubble is released into the flow. The threshold force that must be overcome by the hydrodynamic forces is estimated by approximating surface tension forces based on the analysis performed by Lee

et al. [6]. Flow maldistribution resulting from bubble injection is regulated through actuation of valves at the entrance of each microchannel.

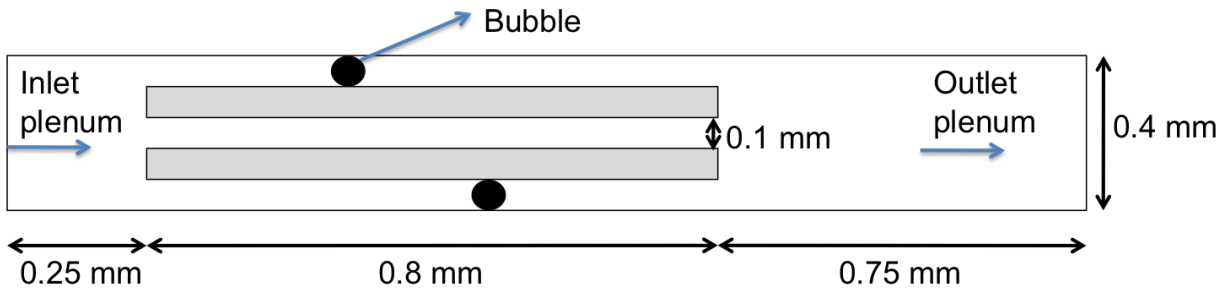


Figure 1: Schematic of parallel microchannels with bubbly flow.

The paper is structured as follows. Section 2 provides the mathematical foundation of the direct numerical simulations employed in this study. Details regarding implementation of the numerical algorithm employed are presented in Section 3. Control design methodology, including the means by which reduced order models are produced from direct numerical simulations and the use of such models in a model predictive control scheme, is detailed in Section 4. Section 6 presents results of validation cases for the flow solver. Controlled flow simulations of bubbly flow are examined in Section 7, with conclusions and avenues for future work presented in Section 8.

2 Mathematical formulation

The computations carried out in this work utilize direct numerical simulation (DNS) with fictitious domain representation of arbitrary shaped immersed objects such as the microvalves and bubbles. The fictitious domain approach Glowinski et al. [10], Patankar [11], Apte et al. [12] allows accurate representation of moving boundaries embedded in a fluid flow. Two types of moving boundaries are considered in this study: (i) specified motion of the immersed object and (ii) freely moving objects. The motion of the microvalves is a *specified* rigid body motion consisting of translation and rotational velocities. The bubbles or particles are allowed to move *freely*. Their motion is obtained by directly computing the forces acting on them. As the first step, we assume the bubbles as *rigid* objects immersed in a surrounding viscous fluid. As shown later, such an assumption is reasonable for low Reynolds numbers and low Weber numbers. For small Weber numbers, the inertial shearing forces acting on the bubble are much smaller than the surface tension forces. Under these conditions bubble deformation is minimal, the shape of the bubble is preserved. One consequence of this assumption is that modeling the motion of the bubble is much easier; the region occupied by the bubble is forced to undergo rigid body motion consisting of only translation and rotation. The bubble motion is then obtained directly by using a novel algorithm based on fictitious domain method for high-density ratios between the fluid and the immersed object. In this fully resolved simulation approach, models for drag, lift, or added mass forces on the bubble are not required, but such forces are directly computed. Below we describe in

detail the computational approach for freely moving rigid objects immersed in a viscous, incompressible fluid. Details of the numerical scheme and several verification and validation test cases are also presented to show good predictive capability of the numerical solver.

Let Γ be the computational domain which includes both the fluid ($\Gamma_F(t)$) and the particle ($\Gamma_P(t)$) domains. Let the fluid boundary not shared with the particle be denoted by $\mathcal{B}$ and have a Dirichlet condition (generalization of boundary conditions is possible). For simplicity, let there be a single rigid object in the domain and the body force be assumed constant so that there is no net torque acting on the object. The basis of fictitious-domain based approach [10] is to extend the Navier-Stokes equations for fluid motion over the entire domain Γ inclusive of immersed object. The natural choice is to assume that the immersed object region is filled with a Newtonian *fluid* of density equal to the object density (ρ_P) and some fluid viscosity (μ_F). Both the real and fictitious fluid regions will be assumed as incompressible and thus incompressibility constraint applies over the entire region. In addition, as the immersed objects are assumed rigid, the motion of the material inside the object is constrained to be a rigid body motion. Several ways of obtaining the rigidity constraint have been proposed [10], [13], [11]. We follow the formulation developed by Patankar [11] which was also applied by Sharma & Patankar [14] for freely moving rigid objects in laminar flows. Apte et al. [12] extended the formulation to co-located grid finite volume schemes with good conservative properties necessary for turbulent flows. Details of the numerical algorithm are described in detail by Apte et al. [12]. A brief description is given here for completeness.

The momentum equation for fluid motion applicable in the entire domain Γ is given by:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\,\mathbf{u}\right) = -\nabla p + \nabla\cdot\left(\mu_F\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right)\right) + \rho\mathbf{g} + \mathbf{f}, \tag{1}$$

where ρ is the density field, $\mathbf{u}$ the velocity vector, p the pressure, μ_F the fluid viscosity, $\mathbf{g}$ the gravitational acceleration, and $\mathbf{f}$ is an additional body force that enforces rigid body motion within the immersed object region Γ_P. The fluid velocity field is constrained by the conservation of mass which for an incompressible fluid simply becomes: $\nabla\cdot\mathbf{u} = 0$.

In order to enforce that the material inside the immersed object moves in a rigid fashion, a rigidity constraint is required that leads to a non-zero forcing function $\mathbf{f}$. Inside the particle region, the rigid body motion implies vanishing deformation rate tensor:

$$\left.\begin{array}{c} \frac{1}{2}\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right) = \mathbf{D}[\mathbf{u}] = 0, \\ \Rightarrow \mathbf{u} = \mathbf{u}^{RBM} = \mathbf{U} + \Omega\times\mathbf{r} \end{array}\right\} \text{ in } \Gamma_P, \tag{2}$$

where $\mathbf{U}$ and Ω are the translation and angular velocities of the object and $\mathbf{r}$ is the position vector of a point inside the object from its centroid.

The vanishing deformation rate tensor for rigidity constraint automatically ensures the incompressibility constraint inside the particle region. The incompressibility constraint gives rise to the scalar field (the pressure, p) in a fluid. Similarly, the tensor constraint $\mathbf{D}[\mathbf{u}] = 0$ for rigid motion gives rise to a tensor field inside the particle region. A fractional-step algorithm can be devised to solve the moving boundary problem [11, 12]. Knowing the solution at time level t^n the goal is to find $\mathbf{u}$ at time t^{n+1}.

1. In this first step, the rigidity constraint force $\mathbf{f}$ in equation 1 is set to zero and the equation together with the incompressibility constraint (equation 2) is solved by standard fractional-step schemes over the entire domain. Accordingly, a pressure Poisson

equation is derived and used to project the velocity field onto an incompressible solution. The obtained velocity field is denoted as $\mathbf{u}^{n+1}$ inside the fluid domain and $\hat{\mathbf{u}}$ inside the object.

2. The velocity field for a freely moving object is obtained in a second step by projecting the flow field onto a rigid body motion. Inside the object:

$$\rho_P \left(\frac{\mathbf{u}^{n+1} - \hat{\mathbf{u}}}{\Delta t} \right) = \mathbf{f}. \tag{3}$$

To solve for $\mathbf{u}^{n+1}$ inside the particle region we require $\mathbf{f}$. The constraint on the deformation rate tensor given by equation 2 can be reformulated to obtain:

$$\nabla \cdot \left(\mathbf{D}[\mathbf{u}^{n+1}] \right) = \nabla \cdot \left(\mathbf{D} \left[\hat{\mathbf{u}} + \frac{\mathbf{f}\Delta t}{\rho} \right] \right) \quad = \quad 0; \tag{4}$$

$$\mathbf{D}[\mathbf{u}^{n+1}] \cdot \mathbf{n} = \mathbf{D} \left[\hat{\mathbf{u}} + \frac{\mathbf{f}\Delta t}{\rho} \right] \cdot \mathbf{n} = 0. \tag{5}$$

The velocity field in the particle domain involves only translation and angular velocities. Thus $\hat{\mathbf{u}}$ is split into a rigid body motion ($\mathbf{u}^{RBM} = \mathbf{U} + \Omega \times \mathbf{r}$) and residual non-rigid motion ($\mathbf{u}'$). The translational and rotational components of the rigid body motion are obtained by conserving the linear and angular momenta and are given as:

$$M_P \mathbf{U} \quad = \quad \int_{\Gamma_P} \rho_P \hat{\mathbf{u}} d\mathbf{x}; \tag{6}$$

$$\mathcal{I}_P \Omega \quad = \quad \int_{\Gamma_P} \mathbf{r} \times \rho_P \hat{\mathbf{u}} d\mathbf{x}, \tag{7}$$

where M_P is the mass of the particle and $\mathcal{I}_P = \int_{\Gamma_P} \rho_P [(\mathbf{r} \cdot \mathbf{r})\mathbf{I} - \mathbf{r} \otimes \mathbf{r}] d\mathbf{x}$ is the moment of inertia tensor. Knowing $\mathbf{U}$ and Ω for each particle, the rigid body motion inside the particle region $\mathbf{u}^{RBM}$ can be calculated.

3. The rigidity constraint force is then simply obtained as $\mathbf{f} = \rho(\mathbf{u}^{RBM} - \hat{\mathbf{u}})/\Delta t$. This sets $\mathbf{u}^{n+1} = \mathbf{u}^{RBM}$ in the particle domain. Note that the rigidity constraint is non-zero only inside the particle domain and zero everywhere else. This constraint is then imposed in a third fractional step.

In practice, the fluid flow near the boundary of the particle (over a length scale on the order of the grid size) is altered by the above procedure owing to the smearing of the particle boundary. The key advantage of the above formulation is that the projection step only involves straightforward integrations in the particle domain.

The above formulation can be easily generalized to particles with *specified motion* (such as the microvalves) by directly setting $\mathbf{u}^{RBM}$ to the specified velocity. In this case, the integrations (equations 6) in the particle domain are not necessary.

3 Numerical Approach

The preceding mathematical formulation is implemented in a co-located, structured grid, three-dimensional flow solver based on a fractional-step scheme developed by Apte et al. [12]. Modifications to the original scheme for freely moving objects were made in order to handle large density ratios ($\mathcal{O}(1000)$) representative of water-to-air bubbles. Accordingly, in the present work the fluid-particle system is solved by a *three-level fractional step scheme*. First the momentum equations (without the pressure and the rigidity constraint terms) are solved. The incompressibility constraint is then imposed by solving a *variable-coefficient Poisson equation* for pressure. Finally, the rigid body motion is then enforced by constraining the flow inside the immersed object to translational and rotational motion. The main steps of the numerical approach are given below.

3.1 Immersed Object Representation

In the numerical implementation, we create small material volumes of cubic shape that completely occupy the immersed object (see Figure 2). Each material volume is assigned the properties of the immersed object (e.g. density etc.). The shape of the object can be reconstructed from these material volumes by computing an indicator or color function (with value of unity inside the object and zero outside) on a fixed background mesh used for flow solution. In this work, the material volumes are forced to undergo rigid motion, based on the translational and rotational velocities of the object, resulting in no relative motion among them. At each time-step the material volumes are advanced to new locations. In the

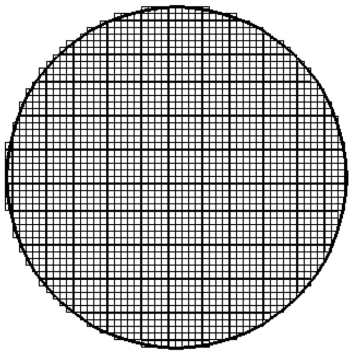


Figure 2: Schematic of material volumes for a circular object.

present approach, the boundary of the object is represented in a stair-stepped fashion and it is straightforward to create the material volumes using a bounding-box algorithm [14, 12]:

1. Determine the bounding box for the particle based on its surface representation.
2. Generate cubic grid within the bounding box.
3. Use distance searches to determine if the centroid of the control volume lies within the bounding surface of the particle.
4. Eliminate points outside the particle domain.

The total mass of the material volumes generated will be exactly equal to the mass of the particle if the surface of the particle aligns with the grid. The stair-stepped surface representation, however, results in an error in the total mass of the material volumes compared to the original shape. This error reduces with an increase in the total number of material volumes per object. A more complex grid generation process based on Delaunay triangulation can be used to accurately represent the surface of the object by using standard body-fitted grid generation tools. In the present work; however, we use sufficient number of material volumes to represent the object boundary and follow the stair-stepped approach owing to its simplicity.

3.2 Discretized Equations and Numerical Algorithm

Figure 3 shows the schematic of variable storage in time and space. All variables are stored at the control volume (cv) center with the exception of the face-normal velocity u_{N}, located at the face centers. The face-normal velocity is used to enforce continuity equation. Capital letters are used to denote particle fields. The time-staggering is done so that the variables are located most conveniently for the time-advancement scheme. We follow the collocated spatial arrangement for velocity and pressure field [15], [16]. Accordingly, the particle positions (X_i), density (ρ), volume fraction (Θ), viscosity (μ), and the pressure (p) are located at time level $t^{n-1/2}$ and $t^{n+1/2}$ whereas the velocity fields (u_i, u_{N}, and U_i) and the rigid body constraint force $f_{i,R}$, are located at time level t^n and t^{n+1}. This makes the discretization symmetric in time, a feature important to obtain good conservation properties.

The semi-discretization of the governing equations in each time-step is given below.

Step 1: Starting with a solution at t^n and $t^{n-1/2}$, the centroids of material volumes ($X_{i,M}$) representing immersed objects are first advanced explicitly.

$$X_{i,M}^{n+1/2} = X_{i,P}^{n-1/2} + \mathcal{R}_{ij}\left(X_{j,M}^{n-1/2} - X_{j,P}^{n-1/2}\right) + U_{i,M}^{n}\Delta t, \tag{8}$$

where $X_{i,M}$ is the position vector of the material volume center, $X_{i,P}$ is the position vector of the immersed object centroid, $U_{i,M}$ is the translation velocity, $\Omega_{i,M}$ is the angular velocity, and Δt is the time-step. Here $\mathcal{R}_{ij}$ is the rotation matrix evaluated using particle locations at $t^{n-1/2}$. The details of the particle update and the rotation matrix are given in Appendix 9.

Step 2: Knowing the new positions of the material volumes and particle centroid, an indicator function (color function) $\Theta^{n+1/2}$ is evaluated at the cv-center of the fixed background grid. We use a discrete delta-function [17] to compute the color function. The color function is unity inside the particle region and vanishes outside with smooth variation near the boundary. This thus allows identification of the particle on the background mesh. Details of the interpolation between the material volume centers and the cv center are given in Appendix 9. The density and the viscosity are then calculated over the entire domain as:

$$\rho_{cv}^{n+1/2} = \rho_P\Theta_{cv}^{n+1/2} + \rho_F\left(1-\Theta_{cv}^{n+1/2}\right) \tag{9}$$

$$\mu_{cv}^{n+1/2} = \mu_P\Theta_{cv}^{n+1/2} + \mu_F\left(1-\Theta_{cv}^{n+1/2}\right) \tag{10}$$

where ρ_P is the density of the immersed particle and ρ_F is the density of the surrounding fluid. Likewise μ_P is dynamic viscosity of the fictitious fluid inside the particle region, and

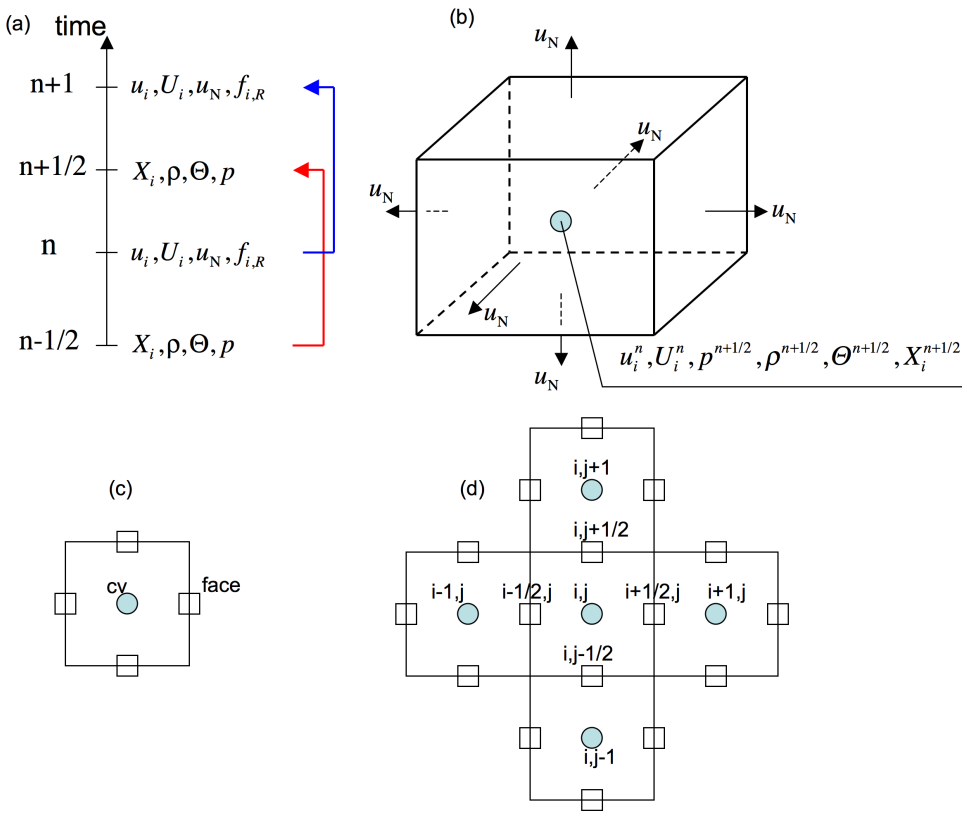


Figure 3: Schematic of the variable storage in time and space: (a) time-staggering, (b) three-dimensional variable storage, (c) cv and face notation, (d) index notation for a given k-index in the z direction. The velocity fields (u_i, u_N) are staggered in time with respect to the volume fraction (Θ), density (ρ), and particle position (X_i), the pressure field (p), and the rigid body force ($f_{i,R}$). All variables are collocated in space at the centroid of a control volume except the face-normal velocity u_N which is stored at the centroid of the faces of the control volume.

μ_F is the dynamic viscosity of the surrounding fluid. For particles with specified motion (microvalves) μ_P is assumed equal to the fluid viscosity (μ_F). For bubbles, appropriate viscosity of the air bubble is specified.

Step 3: Advance the momentum equations using the fractional step method [18]. First, obtain a predicted velocity field over the entire domain. We advance the velocity field from t^n to t^{n+1}. The predicted velocity fields may not satisfy the continuity or the rigidity constraints. These are enforced later.

$$\frac{u^*_{i,\mathrm{cv}} - u^n_{i,\mathrm{cv}}}{\Delta t} + \frac{1}{V_{\mathrm{cv}}} \sum_{\text{faces of cv}} u^{*n+1/2}_{i,\mathrm{face}} u^n_{\mathrm{N}} A_{\mathrm{face}} = \frac{1}{\rho^{n+1/2}_{cv}} \left(\frac{1}{V_{\mathrm{cv}}} \sum_{\text{faces of cv}} \tau^{*n+1/2}_{ij,\mathrm{face}} \mathrm{N}_{j,\mathrm{face}} A_{\mathrm{face}} \right) + g_i \quad (11)$$

where g_i is the gravitational acceleration, V_{cv} is the volume of the cv, A_{face} is the area of the face of a control volume, $\mathrm{N_{j,face}}$ is the face-normal vector and

$$\begin{aligned} u^{*n+1/2}_{i,\mathrm{face}} &= \frac{1}{2}\left(u^n_{i,\mathrm{face}} + u^*_{i,\mathrm{face}}\right); \\ \tau^{*n+1/2}_{ij,\mathrm{face}} &= \mu^{n+1/2}_{cv} \left[\frac{1}{2} \left(\frac{\partial u^n_i}{\partial x_j} + \frac{\partial u^*_i}{\partial x_j} \right) + \left(\frac{\partial u^n_j}{\partial x_i} \right) \right]_{\mathrm{face}} \end{aligned}$$

In the above expressions, the velocities at the 'face' are obtained by using arithmetic averages of the neighboring cvs attached to the face. For the viscous terms, the velocity gradients in the direction of the momentum component are obtained implicitly using Crank-Nicholson scheme. A centered discretization scheme is used for spatial gradients. Evaluation of the pressure gradients at the cv centers is explained below.

Step 4: Solve the variable coefficient Poisson equation for pressure:

$$\frac{1}{\Delta t} \sum_{\text{faces of cv}} u^*_N A_{\mathrm{face}} = \sum_{\text{faces of cv}} \frac{1}{\rho^{n+1/2}_{\mathrm{face}}} A_{\mathrm{face}} \frac{\delta p}{\delta N}^{n+1/2}, \quad (12)$$

where ρ_{face} is obtained using arithmetic averages of density in the neighboring cvs. The face-normal velocity u^*_N and the face-normal pressure gradient are obtained as:

$$\begin{aligned} u^*_{\mathrm{N}} &= \frac{1}{2}(u^*_{i,\mathrm{nbr}} + u^*_{i,\mathrm{cv}})\mathrm{N}_{i,\mathrm{face}} \\ \frac{\delta p}{\delta N}^{n+1/2} &= \frac{p^{n+1/2}_{\mathrm{nbr}} - p^{n+1/2}_{\mathrm{cv}}}{|\mathbf{s}_{\mathrm{cv,nbr}}|} \end{aligned}$$

where nbr represents neighboring cv associated with the $face$ of the cv, and $|\mathbf{s}_{\mathrm{cv,nbr}}|$ is the distance between the two cvs. The variable-coefficient pressure equation is solved using a Bi-Conjugate gradient algorithm [19].

Step 5: Reconstruct the pressure gradient at the cv centers using density and face-area weighting first proposed by Ham & Young [20]

$$\frac{1}{\rho^{n+1/2}_{cv}} \frac{\delta p}{\delta x_i}^{n+1/2} = \frac{\sum_{\text{faces of cv}} \frac{1}{\rho^{n+1/2}_{\mathrm{face}}} \frac{\delta p}{\delta N}^{n+1/2} \cdot \vec{i} |N_{i,\mathrm{face}} A_{\mathrm{face}}|}{\sum_{\text{faces of cv}} |N_{i,\mathrm{face}} A_{\mathrm{face}}|} \quad (13)$$

Step 6: Update the *cv*-center and face-normal velocities to satisfy the incompressibility constraint:

$$\widehat{u}_{i,\mathrm{cv}} = u^*_{i,\mathrm{cv}} - \Delta t \frac{\delta p^{n+1/2}_{\mathrm{cv}}}{\delta x_i} \tag{14}$$

$$\widehat{u}_{\mathrm{N}} = u^*_{\mathrm{N}} - \Delta t \frac{\delta p^{n+1/2}}{\delta \mathrm{N}} \tag{15}$$

The face-normal velocity field $\widehat{u}_{\mathrm{N}}$ will satisfy the incompressibility constraint, however, the *cv*-based velocity may not satisfy the rigid-body constraint inside the particle region. Note that in the absence of any rigid body, $\rho = \rho_F$ throughout the domain, and the algorithm reduces to the standard fractional step scheme for single-phase, incompressible flow. The above velocity field will then be denoted as $u^{n+1}_{i,\mathrm{cv}}$. In the presence of rigid bodies, the following steps are performed to enforce the rigidity constraint within the particle domain.

Step 7: First interpolate the velocity field $\widehat{u}_{i,\mathrm{cv}}$ from the grid cvs to the material volume centroids to obtain $\widehat{U}_{i,M}$ using the kernel interpolation outlined in the Appendix 9. Solve for the translational and rotational velocity fields

$$\mathcal{M}_P \mathbf{U}^T_P = \sum_{M=1}^{N} V_M \rho_M \mathbf{U}_M \tag{16}$$

$$\mathcal{I}_P \mathbf{\Omega}_P = \sum_{M=1}^{N} \rho_M V_M (\mathbf{r} \times \mathbf{U}_M), \tag{17}$$

where subscripts P and M denote the particle and the material volume centroids respectively, V_M is the volume and ρ_M the density of each material volume, $\mathcal{M}_P = \sum_{M=1}^{N} \rho_M V_M$ is the total mass of the particle, $\mathcal{I}_P$ is the moment of inertia of the particle about the coordinate axes fixed to the particle centroid, and $\mathbf{r}$ is the position vector of a point within the particle region with respect to the particle centroid.. The moment of inertia is given as

$$\mathcal{I}_P = \sum_{M=1}^{N} \rho_M V_M \left[(\mathbf{r} \cdot \mathbf{r}) \mathbf{I} - \mathbf{r} \otimes \mathbf{r} \right], \tag{18}$$

where $\mathbf{I}$ represents the identity matrix. The rigid body motion is then obtained as:

$$\mathbf{U}^{RBM}_M = \mathbf{U}^T_M + \mathbf{\Omega}_P \times (\mathbf{X}_M - \mathbf{X}_P). \tag{19}$$

Step 8: Compute the rigid-body constraint force and correct the velocity field to satisfy this constraint within the particle region.

$$\mathrm{F}^{n+1}_{i,M} = -\frac{(U_{i,M} - U^{RBM,n+1}_{i,M})}{\Delta t}. \tag{20}$$

The force on the grid control volumes ($f_{i,\mathrm{cv}}$) is obtained from $\mathrm{F}_{i,M}$ by using the interpolation scheme discussed in Appendix 9. The velocity field inside the particle region is then modified as:

$$u^{n+1}_{i,\mathrm{cv}} = \widehat{u}_{i,\mathrm{cv}} + \Delta t f^{n+1}_{i,\mathrm{cv}}. \tag{21}$$

4 Controller Design

Model-based control design requires a reduced order model of the flow dynamics that relates individual channel valve openings with the exit flow velocities for each channel. While direct numerical simulations of the flow field produce the most accurate relationship between these quantities, the computationally intensive nature of these simulations precludes their use in any real-time physical realization. However, many real-time control methodologies have been developed to control linear, multiple-input, multiple-output (MIMO) systems. The development of a reduced order, linear MIMO model is therefore motivated both by the ability of linear models to adequately represent nonlinear flow dynamics in certain flow regimes as well as the relative success of linear control methodologies in controlling nonlinear systems.

4.1 System Identification

As a result, standard system identification techniques [21] are employed to produce a linear model of the flow dynamics which relates channel valve openings (inputs) to channel exit flow velocities (outputs). Specifically, an auto-regressive exogenous (ARX) model [22] of the flow dynamics is developed from direct numerical simulations of the flow regime, in which channel valve openings are varied in a prescribed fashion and the resulting output flow velocities are recorded. While a linear, MIMO ARX model is developed, the relationship between inputs and outputs in the ARX formulation is most easily illustrated for the single-input, single-output case. In this instance a linear difference equation relates the input and output

$$\begin{aligned} y(t) &= -a_1 y(t-1) - a_2 y(t-2) - ... - a_{n_a} y(t-n_a) + \\ &\quad b_1 u(t-n_k) + b_2 u(t-n_k-1) + ... + b_{n_b} u(t-n_k-n_b+1) \end{aligned} \tag{22}$$

where y(t) is the output, u(t) is the input, n_k is the time delay, n_a is the number of poles, n_b is the number of zeros plus one, and a_i and b_j are constants to be determined via the identification process. The equation for the current output is therefore a function of both values of the output and the input at previous sampling instants. The choice of how many previous input and output values to retain is driven by the model validation procedure, in which the output prediction of the model is compared to the results obtained from direct numerical simulations of the flow for data not utilized in the identification process. In the multi-variable case, the coefficients a_i and b_i become $n_o \times n_o$ and $n_o \times n_i$ matrices, respectively, where n_o and n_i represent the number of model outputs and inputs.

System identification is conducted following the procedure presented in [22]. Uncontrolled simulations are utilized to determine the system settling time, which informs the choice of both the sampling interval and the duration of the identification tests. Identification tests are subsequently conducted via numerical simulations of the flow field, in which channel valve angles are randomly varied to excite all modes of the flow dynamics. Flow velocities at the exit of each channel are recorded and this output data, in conjunction with the recorded variation of the input valve orientations, is processed within the Matlab system identification toolbox to produce multiple linear ARX models of varying order. Models

are validated against numerical simulation data not used in the identification procedure. Channel exit flow velocities are generated by each model from the prescribed variation of the input valves used to produce the validation data set. These velocities are compared to those obtained by direct numerical simulation of the flow field. Model selection is governed by output accuracy, as balanced with model simplicity. The selected linear, MIMO system model is subsequently employed as a substitute for the actual flow dynamics in the model predictive controller design.

4.2 Model Predictive Control

A model predictive control (MPC) methodology [23] is employed to equalize flow velocities in a parallel microchannel configuration in the presence of bubble disturbances. Benefits of model predictive control include: real-time optimization of control outputs, direct incorporation of constraints on both manipulated and controlled variables, successful system operation closer to constraints, and robustness to model uncertainty and external disturbances. An overview of the functionality of a model predictive controller is illustrated in Figure 4 and summarized below.

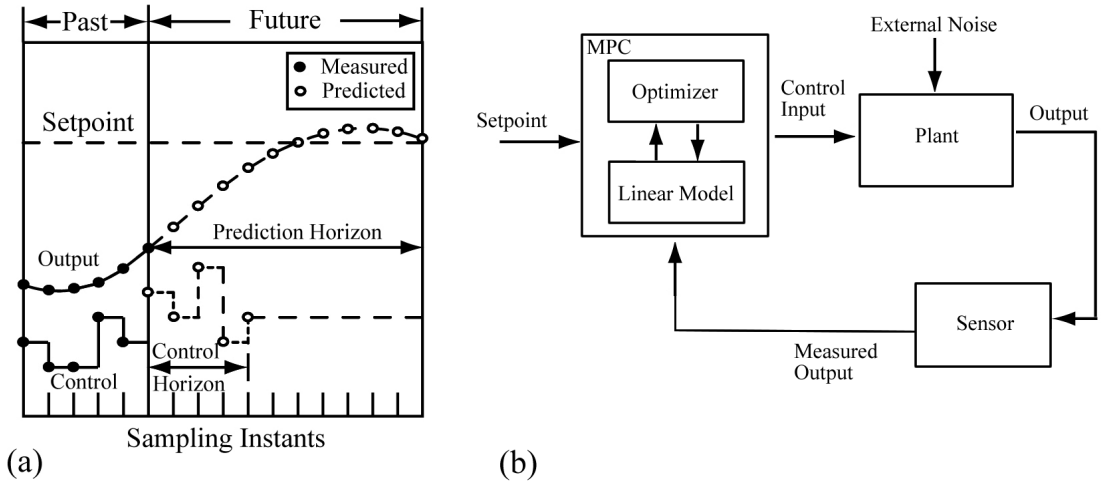


Figure 4: Model predictive control scheme: (a) inputs and outputs and their relation to the control and prediction horizons (b) control system diagram including the model predictive controller.

For the parallel microchannel configuration studied, the model predictive controller prescribes the valve openings for each channel (control inputs) from knowledge of the desired flow velocity at the exit of each channel (system setpoints) and the actual flow velocities (measured outputs). To determine the valve openings that will result in the measured flow velocities reaching the desired values in the future, the model predictive controller utilizes the linear system model produced by the system identification procedure. Predictions of the exit flow velocities are generated using the model for a user-specified duration into the future (prediction horizon) from a sequence of channel valve openings over a user-specified control horizon. Optimal values for the valve openings are determined, via solution of a quadratic

programming problem, over the control horizon such that a cost function involving the deviation from the desired setpoints is minimized over the prediction horizon. Once the optimal sequence of valve openings is determined, only the first set of openings are provided to the flow solver (plant). At the next sampling instant, the resulting exit flow velocities are measured and sent to the model predictive control scheme. Utilizing this new information regarding the actual flow velocities achieved as a result of the valve openings, as opposed to those predicted by the linear model, the process repeats.

Realization of the model predictive control design is achieved by through the Matlab model predictive control toolbox. The system model, control horizon, prediction horizon, cost function structure and associated weighting matrices, setpoint values and the measured outputs are input into the MPC toolbox. The quadratic programming problem is solved within the toolbox to produce the channel valve openings utilized by the flow solver. The resulting strategy is computationally lightweight, enabling future physical implementation with small microcontrollers.

5 CFD-Controller Interface

To perform the coupled CFD-controller simulations, we have developed a simple communication interface between the Matlab-based control algorithms and Fortran-based CFD solver. The controller algorithms are run on a single processor; whereas, the direct numerical simulations are performed on multiple processors. Figure 5 shows the schematic of the communication interface between the two packages.

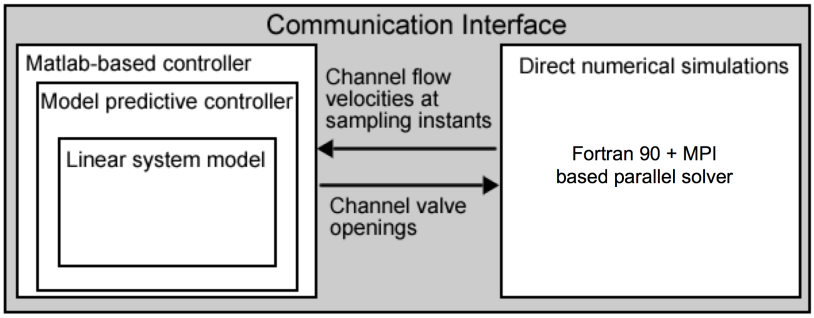


Figure 5: Schematic of CFD-controller interface.

An external communication interface, developed in Fortran, is created to manage execution of both the CFD simulation and the Matlab-based model predictive controller. Direct numerical simulations are executed as directed by the communication interface, and at each sampling instant, produce an ASCII text file containing the outlet flow velocities of each channel. This file is read by the Matlab-based control program, and system states are employed by the model predictive control algorithm to produce a new set of valve openings, written to an external file. Once written, the file of valve openings is read by the CFD solver, and the process repeats. Valve openings are not instantaneously changed in the direct numerical simulations; rather, a maximum rate of change is prescribed and the valves move from their prior position to the prescribed openings at this rate.

The direct numerical simulations mainly govern the total simulation time mainly due to the details of the flow captured. The Matlab-based controller takes negligible amount of time compared to CFD. This framework is efficient and is applicable to several other physics problems requiring coupling of CFD and model-predictive control.

6 Validation cases

The numerical formulation together with the control algorithm are first used to perform basic validation and verification studies. The numerical test cases were chosen to validate the basic incompressible flow algorithm applied to high-aspect ratio channel flows, flow developed by objects undergoing specified motion to test the capability of the solver to handle moving microvalves, motion of freely moving particles and bubbles, and finally testing of the coupled CFD-control algorithm. Accordingly, the following tests are performed: (i) single phase incompressible flow in a high-aspect ratio microchannel at different Reynolds numbers corresponding to the experimental data of Qu *et al.* [24], (ii) flow over a fixed cylinder at different Reynolds number, (iii) flow developed by an oscillating sphere corresponding to experiments by [25], (iv) rising of a small bubble in an inclined channel corresponding to experiments by [26], and (v) equalizing mass-flow rates in parallel microchannels using model-predictive control. After these extensive validation studies, the coupled solver is applied to mitigate flow maldistribution in parallel microchannels.

6.1 Microchannel channel case

Simple flow in a 3D channel, as studied by Qu *et al.* [24], serves as the initial test case. Validation of the flow solver for this test case is a critical step towards our simulation goal, since the parallel microchannel geometry of interest involves flow through microchannels with a high aspect ratio (see Fig. 6).

Table 1: Dimensions (in $[mm]$) for the single-microchannel [24].

Wch	Wp	Hch	Lp1	Lpch	Lp2
0.222	6.35	0.694	6.35	120	12.7

Channel dimensions are given in Table 1. As seen from the schematic, the ratio of plenum to microchannel height is very large. The computational grid consists of around 1.5 M grid points with roughly 30 cells inside the microchannel in the vertical direction. The grid cells are refined near the walls and near the entrance region to resolve the flow accurately. Fluid flow through the channel is simulated for different Reynolds numbers ($Re = 196$ and $Re = 1895$), and the resulting velocity profiles and pressure drops are compared with the experimental and numerical results presented in [24]. In these simulations, two large plenums are used at the inlet and outlet of the domain to ensure that the boundary conditions do not perturb the solution inside the channel. A fully developed velocity profile for a three-

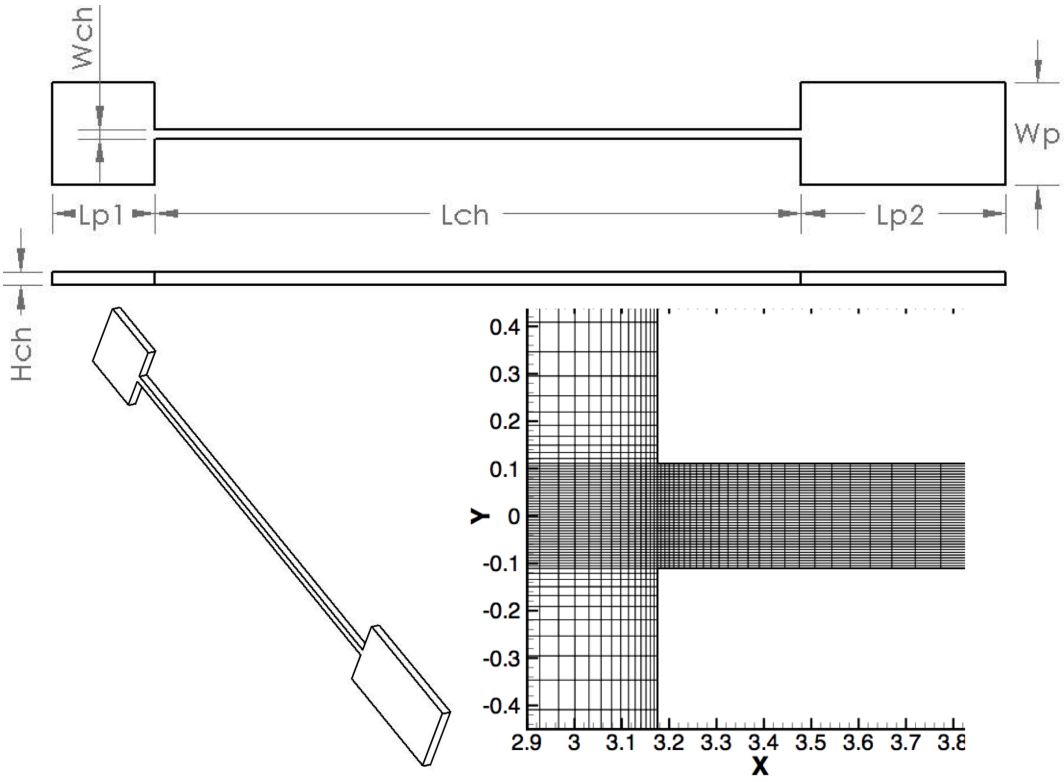
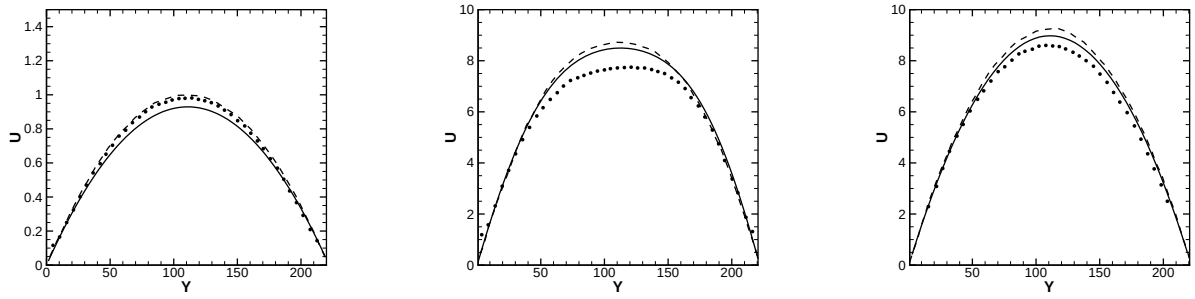


Figure 6: Schematic and grid for the single channel geometry. The grid used consists of around 1.5M grid elements. tOnly a small section of the grid is shown.

dimensional rectangular channel [24] is applied at the inlet of the domain and data is only collected after the flow in the channel reaches steady state.



(a) $Re = 196, x' = 1\,cm$ (b) $Re = 1895, x' = 1\,cm$ (c) $Re = 1895, x' = 10\,cm$

Figure 7: Velocity profiles in the center plane of the channel taken at $x' = 1cm$ and $x' = 10cm$ from the entrance of the channel. • shows the experimental data, $- - -$ the numerical simulation from [24] and — the present study. The velocity is expressed in $[m/s]$ and the y location in $[\mu m]$.

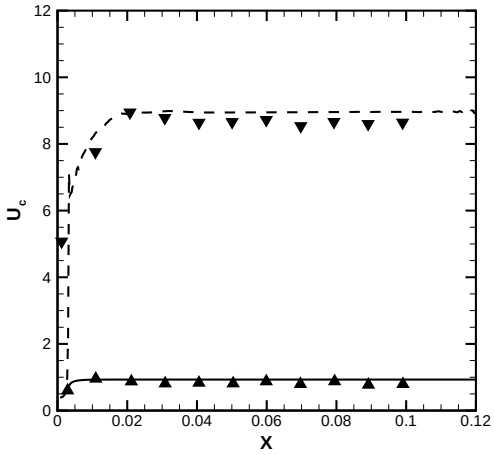


Figure 8: Velocity distribution along the central line of the channel. $Re = 196$: ▲ experimental data from [24], — present simulation; $Re = 1895$: ▼ experimental data, $- - -$ present simulation. Velocity is expressed in $[m/s]$ and the $[x]$ location in $[m]$.

Figure 7 illustrates the comparison of the simulated velocity profile over the cross-section at different locations within the channel to the numerical and experimental data of [24]. The simulations produce a parabolic profile and good agreement with the both the numerical and experimental results. For larger Reynolds numbers, the numerical velocity at the center of the channel is slightly larger than the experimental data; however, agrees with the simulations conducted by Qu *et al.* [24]. These deviations are within the uncertainties in maintaining

constant flow rates as well as velocity measurements. As illustrated in Figure 8, the velocity along the center line of the channel, for the length of the channel, shows very good agreement with the experimental data.

Table 2: Comparison between computed and theoretical pressure drops at different Reynolds numbers.

	Pressure Drop $[bar]$		
Re_{ch}	DNS (with plenum)	DNS (without plenum)	Theory (without plenum)
196	0.189	0.195	0.20
1021	1.09	1.06	1.04
1895	1.33	1.87	1.93

Table 2 presents the comparison of numerically obtained pressure drop inside the microchannel with the theoretical pressure drop in a square microchannel (excluding the plenums). A good agreement is achieved for low Reynolds number (up to $Re_{ch} = 1021$). For $Re_{ch} = 1895$ the pressure drop is under-predicted by the simulation. This is attributed to the effects of sudden changes in aspect ratios near the inlet and outlet plenums. Flow separations are possible near the entrance, modifying the flow evolution to fully developed velocity profile and thus affecting the overall pressure drop. To test the accuracy of the solver, we have also simulated flow through the microchannel duct (without the plenums) with uniform inlet on a grid similar to the grid used for the case with plenums. The pressure-drop values for the same mass-flow rates are shown in Table 2 and agree well with the theoretical estimates.

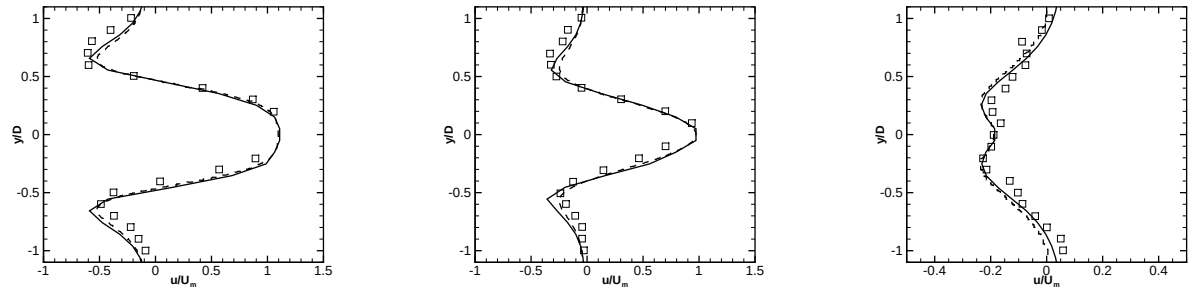
6.2 Flow induced by an oscillating cylinder

Simulation of a periodic oscillating cylinder in a fluid at rest has often been used to test the accuracy of immersed boundary techniques, and serves as our second test case. Numerical simulation results are validated against those of [27] as well as experimental data available from [25].

Two non-dimensional numbers characterize a periodic oscillating cylinder: the Reynolds number $Re = U_m d/\nu$ and the Keulegan-Carpenter number $KC = U_m/fd$, where $U_m = 0.01m/s$ is the maximum velocity of the cylinder, $f = 0.2Hz$ the frequency of the oscillations, $d = 0.01\,m$ is the diameter of the cylinder and ν the kinematic viscosity. The position of the cylinder in the x-direction is defined by the equation:

$$x_p(t) = -A_p \sin \omega t \tag{23}$$

where $x_p(t)$ is the location of the centroid of the cylinder in the x-direction and A_p is the maximum amplitude of the oscillations. Because the oscillation frequency is given by $\omega = 2\pi f$, the Keulegan-Carpenter number can be rewritten as $KC = 2\pi A_p/d$. Simulations are conducted with $Re_d = 100$ and $KC = 5$, as in [27]. The computational domain is



(a) $\omega t = 180°$ (b) $\omega t = 210°$ (c) $\omega t = 330°$

Figure 9: Normalized axial velocity (u/U_m) at three different phase position. The velocity is measured at a fixed x location ($x = -0.6d$) relative to the initial location of the particle center: □ Experimental data from [25]; — present simulation; − − − Numerical results from [27].

comprised of a square box $100d \times 100d \times d$ respectively in the axial, vertical and span-wise directions. The grid is refined and uniform in the region of motion of the cylinder and stretched towards the walls. Grid resolution in the cylinder region is refined so that 20 grid-points are always located along the cylinder diameter. Neumann boundary conditions are applied in the x and y directions. Simulations begin with the both the flow and cylinder at rest, with the cylinder positioned in the center of the domain. Simulation data regarding fluid flow is collected after the cylinder has gone through ten cycles.

The normalized velocity in the wake of the cylinder at three different phases of the cycle is compared to numerical and experimental results in Figure 9. As illustrated, our results show excellent agreement with both the experimental [25] and numerical simulation data [27].

Rising spherical bubbles

We test the numerical algorithm for bubbles rising in a stationary fluid. Two test cases are considered: (i) a single spherical bubble rising in an inclined channel corresponding to the study of Lumholt *et al.* [26] at low density ratio ($\rho_f/\rho_p =$), and (ii) a spherical bubble rising in a straight channel at high density ratio ($\rho_f/\rho_p = 1000$). The first case is important to show that the present approach correctly captures the hydrodynamics of the rising bubble and the thin layer between the bubble and the channel wall. The second case tests the stability of the solver at high density ratios.

6.2.1 Sphere rising in inclined channel

Flow solver performance is also examined for another case of particle motion: a bubble rising in an inclined channel. In the simulation, the bubble is modeled as a rigid spherical particle flowing in an inclined channel. The simulation is conducted with a fluid density of $\rho_f = 1115\, kg/m^3$, a bubble density of $\rho_p = 1081\, kg/m^3$, and a fluid viscosity of $\nu = 3.125\, mm^2/s$.

The Reynolds number Re_p^{Stokes} based on the Stokes settling velocity W is defined as :

$$Re_p^{Stokes} = \frac{2aW}{\nu} = \frac{4a^3}{9\nu^2}|\frac{\rho_p}{\rho_f} - 1|g \tag{24}$$

where $g = 9.82\, m/s^2$ is the gravitational acceleration, and $a = 2\, mm$ is the diameter of the particle. The channel is inclined at an angle of 8.23° with the vertical. This is simulated by adding components of gravitational forces in the horizontal and vertical directions.

The computational domain consists of a rectangular box with dimensions $10\, mm$ in the x direction, $80\, mm$ in the y direction and $40\, mm$ in the z direction. The grid is Cartesian and uniform over the domain with $40 \times 320 \times 160$ grid points, respectively in the x, y and z directions so that $\Delta = 0.25 \times 10^{-3}\, m$. The bubble is injected at $x = -1.4\, mm$, $y = -1.0 mm$ and $z = 20.0 mm$. Simulation results for a Reynolds number of $Re_p^{Stokes} = 13.6$ are compared with experimental and numerical data from [26]. As illustrated in Figure 10, the numerical simulation exhibits excellent agreement with both experimental and numerical results. Buoyancy forces cause the bubble to rise and travel alongside the right wall of the domain. Ultimately, the particle follows the right wall without touching it, keeping a very thin lubrication layer between the particle and the wall. This simulation validates that the numerical approach is capable of simulating freely moving rigid objects.

Rising sphere with high density ratio

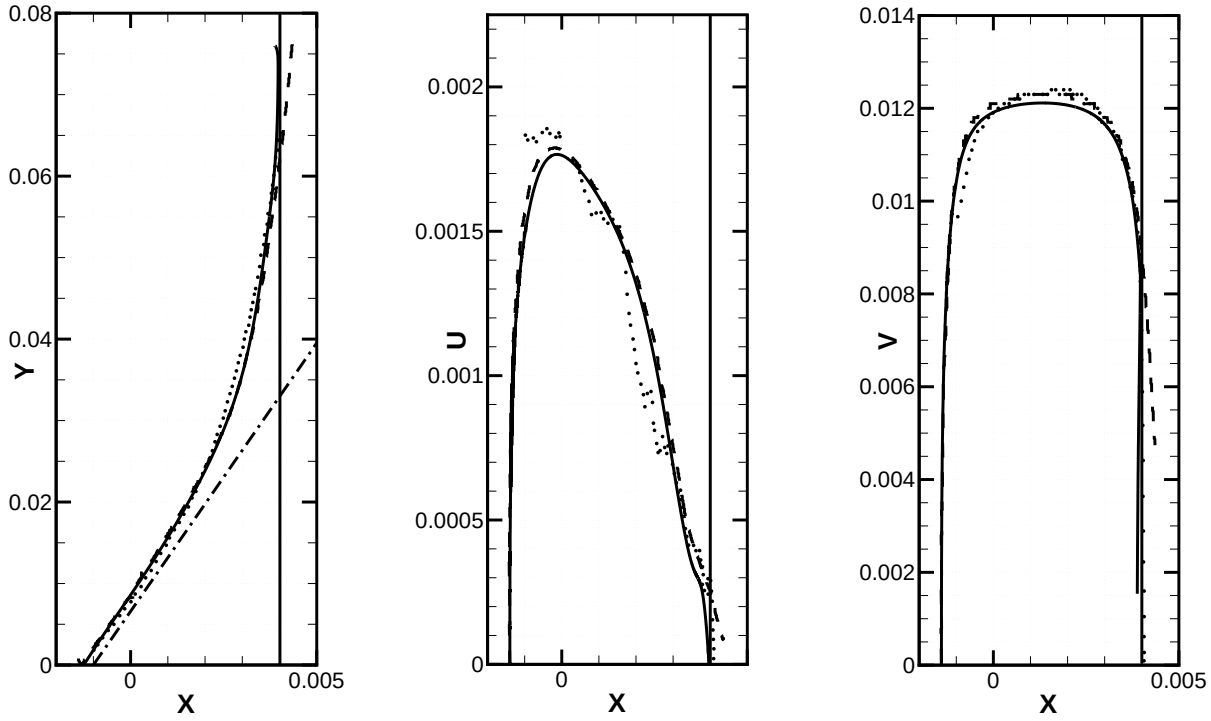
We consider a sphere (diameter 6.66 $[mm]$, $\rho_p = 1$ $[kg/m^3]$) rising in a quiescent fluid ($\rho_f = 1000$ $[kg/m^3]$, $\mu_f = 0.0422$ $[kg/m.s]$) in a vertical channel of size $0.02 \times 0.02 \times 0.09$ $[m]$. Initially the spherical bubble is placed at the center of the channel and a height of 0.01 $[m]$ from the bottom wall. The bubble starts rising due to buoyancy force ($g = 9.8$ $[m/s^2]$) and reaches a terminal velocity. The computation is performed on a uniform grid ($80 \times 80 \times 360$ giving approximately 26 points within the sphere) and the time step is 25 $[\mu s]$.

Mordant & Pinton [28] performed experiments on freely falling spherical particles in a large water tank for various density ratios (maximum density ratio considered was $\rho_p/\rho_f = 14.6$). They showed that for small particles falling in a large tank (that is, for small values of the ratio of particle diameter to tank width $D_p/L \sim 0.005$) the temporal evolution of the particle velocity can be well predicted by the the curve:

$$U^* = 1 - \exp\left(-\frac{3t}{\tau_{95}}\right), \tag{25}$$

where U^* is the velocity of the particle normalized by its terminal velocity, τ_{95} is the time it takes for the sphere to reach 95% of its terminal velocity, and t is time. We compare the temporal evolution of the rising spherical particle to this curve in Figure 11. The present simulations are performed for a much higher density ratio $\rho_f/\rho_p = 1000$. The domain size in simulations is small ($D_p/L = 0.333$) and thus wall effects become important. This test case confirms the stability of the numerical solver when applied to large-density ratio fluid-particle systems.

Additional test cases of the present approach for large-density ratio fluid-particle systems were also performed to show code stability and accuracy [29].



(a) (b) (c)

Figure 10: Comparison of experimental • and numerical simulations − − − [26] with the current simulation —. Figure 10a shows the particle trajectory inside the domain (the −.− line shows the initial trajectory due only to the effect of gravity), 10b the velocity of the particle in the lateral direction and 10c the velocity on the vertical direction. The particle position is expressed in $[m]$ and the velocities are expressed in $[m/s]$.

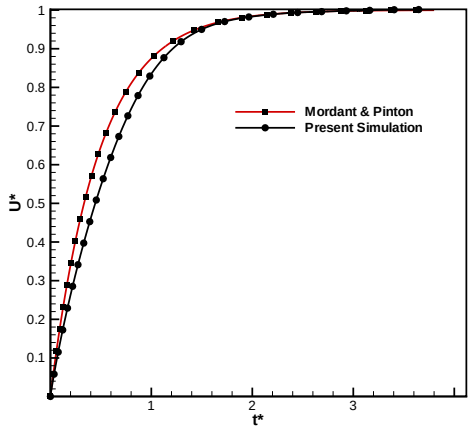


Figure 11: Temporal evolution of a spherical bubble rising in a water column: $U^* = U/U_{terminal}$, $t^* = t/\tau_{95}$. Square symbols, Mordant & Pinton [28] and circle, present simulation.

6.3 Equalizing flow rates in a parallel microchannel geometry

The coupling and coordination between the control system design and the flow solver is examined via flow equalization in a parallel microchannel geometry. In this test case, a fully-developed parabolic velocity profile is imposed at the inlet. This inflow condition combined with the non-symmetric geometry of the parallel microchannels produces different mass flow rates in each of the channels. Because the mass flow rate is imposed at the inlet, the amount of fluid entering the geometry is known. As a result, without any bubble in the geometry, the controller regulates the flow in each channel by varying channel valve openings to achieve equal mass flow rates in each channel. Simulation details, including the geometry, boundary conditions, grid, and controller design are detailed in the following section.

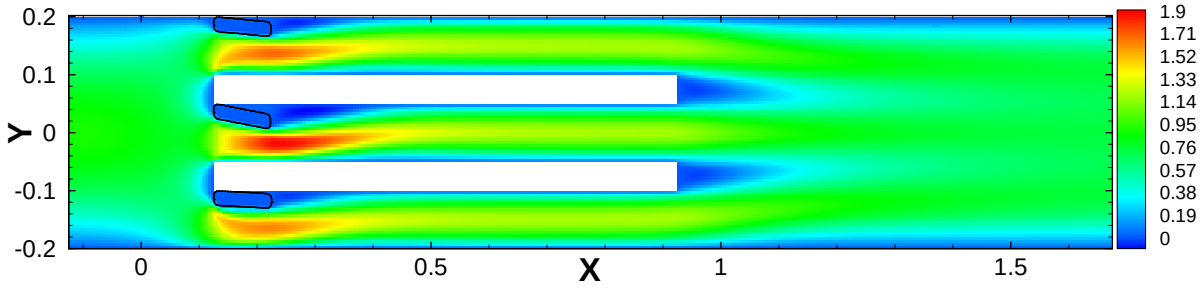


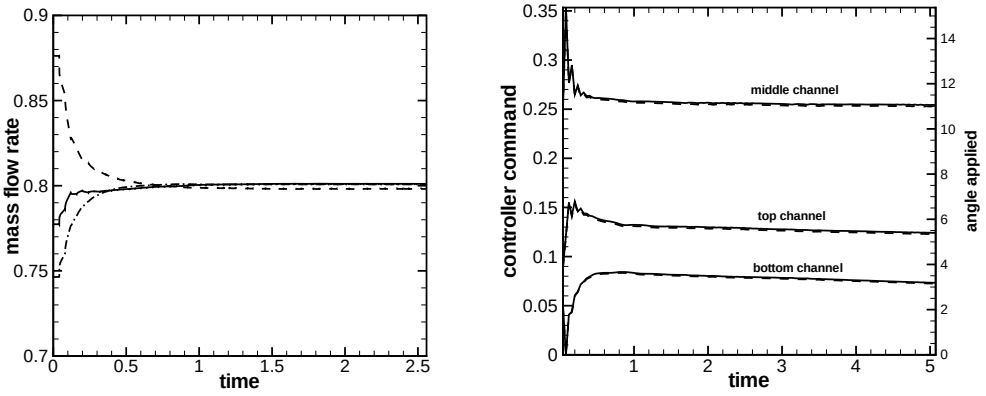
Figure 12: Steady state velocity contours in a parallel microchannel after flow rates in each channel are made equal. Also shown are the steady-state position of the microvalves. Lengths are expressed in $[mm]$ and the velocity contours are in $[m/s]$.

Figure 12 shows the velocity contour of the fluid inside a three-channel microchannel once the steady state has been achieved and the flow equalized by the controller. As illustrated, mass flow rates are equalized in each channel through partial closure of each valve. The maximum amount of closure occurs for the center channel valve, due to the increased flow velocity that occurs as a result of the inlet boundary condition and the asymmetry of the geometry.

Figure 13a shows the mass flow rate history in each channel, and Figure 13b illustrates the corresponding history of the controller actions. Initially, different mass flow rates exist in each channel, with the maximum flow rate occurring in the middle channel. Valve actuation, as prescribed by the controller, partially closes the valves in each of the channels, reducing the flow rate in the center channel while increasing it in both the upper and lower channels until equal flow rates are achieved. The maximum difference between the mass flow rates in the three channels is 0.12%, and the error in the steady state flow rates to the theoretically computed value for the inlet flow rate is 0.16%.

7 Flow Simulation Through Parallel Microchannels

In this section the results for a three-channel microchannel are presented. The schematic of the geometry and the model used for these simulations are presented in Fig. 1. A single



(a) Velocity history in each channel

(b) Controller output history correlated with angle applied to each valve.

Figure 13: Time evolution of the flow rates 13a and controller input 13b to the flow solver: − − − shows the data for the top channel; — middle channel; -.- bottom channel. Mass flow rate is expressed in $[kg/s]$, time in $[ms]$ and angles in degrees.

bubble is introduced in the top channel and the resultant flow distribution is first examined. A control algorithm is developed that senses the presence of the bubble and resultant flow maldistribution, activates microvalves and facilitates effective removal of the bubble. Details of these coupled CFD-control simulations are discussed here.

7.1 Geometry and model

The parallel microchannel geometry considered consists of three microchannels, and two plenums for the inlet and outlet (see Fig. 1). The dimensions of the domain are $0.8\,mm$ in the x direction, $0.4\,mm$ in the y direction and $0.025\,mm$ in the span-wise direction. The span-wise dimension remains constant for the entire geometry. The inlet plenum is $0.25\,mm$ and $0.4mm$ in the x and y directions, respectively. The outlet plenum is $0.75mm$ and $0.4mm$ in the x and y directions, respectively. The outlet plenum is longer than the inlet to prevent the outflow boundary from perturbing the flow inside the channels. Dimensions of all three channels are equal: $0.8\,mm$ in the x direction and $0.1\,mm$ in the y direction. The fluid used in this study is water with fluid properties at room temperature. The density ratio between the fluid and the rigid vapor bubble was held fixed at 10. Simulations with higher density ratios (on the order of 100) revealed similar results as presented below.

The grid is cubic and uniform in each direction so that $\Delta = 0.005mm$, therefore there are five grid points in the z direction all over the domain. Each channel contains 20 grid-points in the y direction and 160 grid-points in the x direction. Over the entire domain there are $128,000$ grid-points.

At the inlet boundary of the domain a 2D fully-developed parabolic profile is applied, the fluid used having the same properties as water. The maximum velocity at the center of the parabolic profile is defined by $\frac{3}{2}U_p$ where $U_p = 0.6\,m/s$ so that the global mass flow rate

entering the computational domain is $\dot{m} = 2.4\, mg/s$.

7.2 System identification and controller development

The system settling time in response to a step change in a valve position was numerically determined as approximately $0.8ms$, which was the time required for the channel flow velocities to reach 98% of their steady state values. A sampling interval of one twentieth of the settling time, or $.04ms$, was chosen for both the identification tests and the control design. A generalized binary noise (GBN) signal was generated to prescribe the closure of each of the valves between fully open and 50% closed for the system identification test. The GBN signal was created with a mean switching time of one third the settling time and a minimum switching time equal to the sampling rate. The duration of the test was twelve times the settling time, or $9.6ms$.

Input and output data generated by the numerical system identification test was processed in the MATLAB system identification toolbox to produce ARX models of varying complexity. Model validation was performed for each model, with model quality judged by a "best fit" criteria

$$Best\ Fit = \left(1 - \frac{\|y - \hat{y}\|}{\|y - \bar{y}\|}\right) \times 100 \tag{26}$$

where y is the measured output, $\hat{y}$ is the predicted output and $\bar{y}$ is mean of the measured output. A model has a perfect Best Fit value of 100% if $\hat{y}$ is equal to y. A best fit value of zero represents a fit which is no better than picking a constant value of $\bar{y}$ for the model output. A model with three poles, two zeros and one delay produced best fit values of 86.5%, 87.6% and 83.7%, for channels one, two and three, respectively. Models of lower order were significantly less accurate while higher order models were only marginally more accurate. The model identified by this process, and subsequently utilized in the controller design, was

$$\begin{aligned} I\mathbf{y}(t) \;=\; & -\begin{bmatrix} -2.09 & -0.91 & -0.74 \\ -1.70 & -3.00 & -1.87 \\ 3.13 & 3.21 & 1.96 \end{bmatrix} \mathbf{y}(t-1) - \begin{bmatrix} 2.15 & 1.92 & 1.69 \\ -1.00 & -0.72 & -0.76 \\ -0.99 & -1.04 & -0.77 \end{bmatrix} \mathbf{y}(t-2) - \\ & \begin{bmatrix} -1.82 & -1.85 & -1.79 \\ 1.99 & 2.04 & 1.92 \\ -0.03 & -0.05 & 0.01 \end{bmatrix} \mathbf{y}(t-3) + \begin{bmatrix} 0.05 & -0.02 & -0.02 \\ -0.02 & 0.05 & -0.02 \\ -0.03 & -0.03 & 0.04 \end{bmatrix} \mathbf{u}(t-1) + \\ & \begin{bmatrix} 0.01 & -0.03 & -0.01 \\ -0.01 & 0.03 & -0.02 \\ -0.01 & -0.00 & 0.02 \end{bmatrix} \mathbf{u}(t-2) + \begin{bmatrix} -0.02 & 0.02 & 0.01 \\ 0.01 & -0.04 & 0.01 \\ 0.01 & 0.02 & -0.02 \end{bmatrix} \mathbf{u}(t-3) \end{aligned} \tag{27}$$

where $\mathbf{y}(t) = [y_1(t)\ y_2(t)\ y_3(t)]^T$ represents the flow velocity in each channel and $\mathbf{u}(t) = [u_1(t)\ u_2(t)\ u_3(t)]^T$ represents the valve closure for each channel.

Controller design required specification of the control horizon, prediction horizon, as well as the cost function form and associated weighting parameters. In this study, a quadratic cost function was constructed in terms of a vector of valve positions, a vector comprised of the change in the valve positions in one sampling period, and a vector representing the difference between the exit flow rates and the desired values. Selection of the values in the weighting matrices of the cost function provided a means to balance the magnitude and rate

of the valve closures with the attainment of the desired exit flow rates. Initial values for the control horizon, prediction horizon and weighting matrices were computed via the tuning strategy detailed in [30]. These parameters were subsequently refined based upon simulation results of the controlled performance. Final values utilized in the simulation results are presented in Table 3.

Table 3: Control parameters used for controller design and system ID tests.

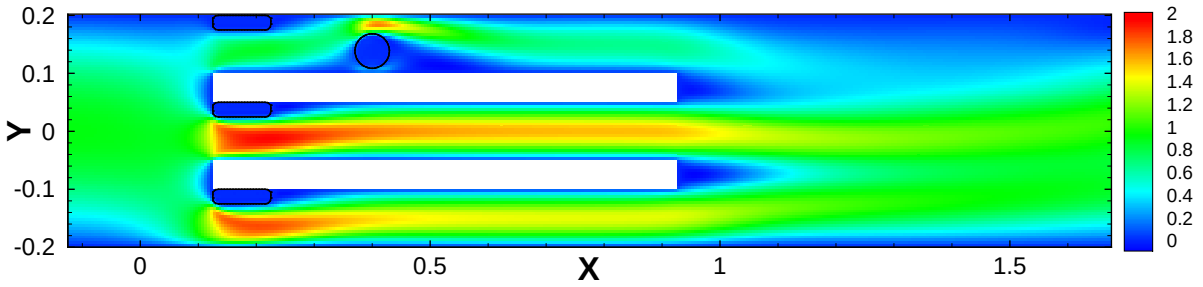
Control Parameters	
Control Horizon	5 sampling periods 0.2 ms
Prediction Horizon	10 sampling periods 0.4 ms
Valve Position Weighting Matrix	Diagonal 3 x 3 matrix, 0.001 on diagonal
Valve Rate Weighting Matrix	Diagonal 3 x 3 matrix, 1.0 on diagonal
Output Error Weighting Matrix	Diagonal 3 x 3 matrix, 10 on diagonal
Valve Opening Constraint	Constrained to remain between 0 and 1
Valve Rate Constraint	Constrained to remain within -0.15 and 0.15

7.3 Effect of a single bubble on the flow distribution

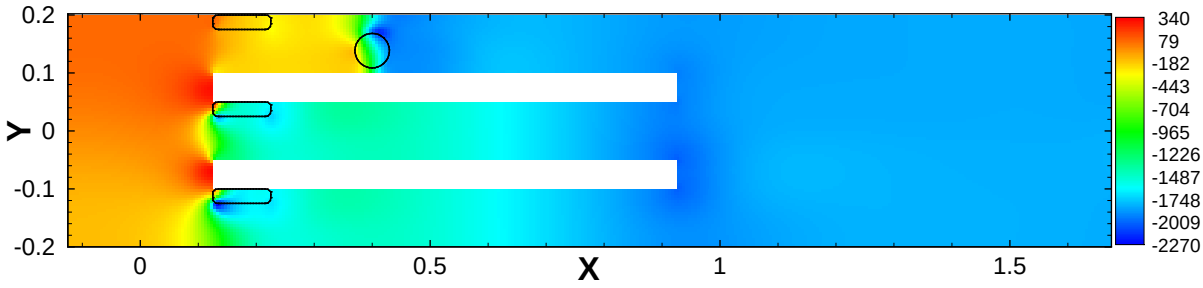
First, the effect of a single bubble, present in the top channel of the three-channel geometry, on the flow distribution is investigated. A 60 μm bubble is introduced in the top channel, close to the bottom side of the channel wall. The bubble plugs 60% of the channel height. The bubble is held fixed at this location and its effect on the flow distribution and pressure drop are presented below. In this work, we do not simulate the physics of bubble nucleation and bubble growth. Instead, a bubble is introduced initially and held fixed. This represents a practical situation wherein the surface tension and viscous forces on the bubble are greater than the hydrodynamic forces, and the bubble remains attached to one side of the channel wall.

Figures 14a-b show contours of axial flow velocity and pressure inside the parallel microchannel in the presence of a bubble in the top channel. For this configuration the microvalves are completely open and controller is not activated, and the flow maldistribution is clearly visible. Figure 15a shows the history of flow rates in each channel after the bubble is injected into the top channel, whereas Fig. 15b shows the steady state pressure drop in each branch of the microchannel in the presence of the bubble, indicating higher pressure drop in the top channel.

Next, we estimate the range of forces acting on the bubble held fixed in the top channel by varying the microvalve configurations. We consider two extreme cases: (a) all microvalves are completely open, and (b) the middle and bottom channel are completely closed by microvalves. In the first case, majority of the flow goes through the middle and bottom branches, whereas in the latter case all flow goes through the top channel. Figure 16 shows the time history of the forces on the bubble in these two extreme configurations. It is found that when the bottom and middle channels are completely closed, the entire inflow goes through the top channel, increasing the hydrodynamic forces on the bubble. The range of the forces applied to the bubble in these extreme configurations vary from 0.013 μN to

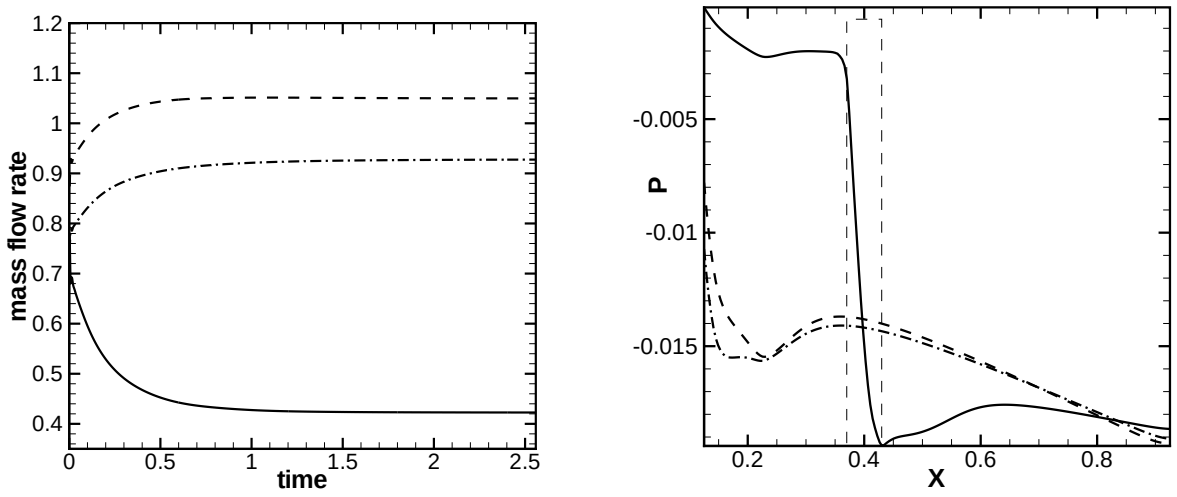


(a) Velocity contour



(b) Pressure contour

Figure 14: Velocity and pressure contours inside the parallel microchannel with presence of a stationary bubble in the top channel. The lengths are expressed in $[mm]$, the velocity in $[m/s]$ and the pressure in $[Pa]$.



(a) Mass Flow Rate

(b) Pressure Drop

Figure 15: Temporal evolution of mass flow rate (15a) in each channel and steady-state pressure drops (15b) in each branch: — the top channel; − − − middle channel; -.-.- bottom channel. The mass flow rates are expressed in $[mg/s]$, time in $[ms]$, length in $[mm]$, and pressure in $[bar]$. The dashed rectangle in right panel represents the location of the bubble.

0.19 μN. The magnitude of the force, when the bottom and middle channels are closed, is

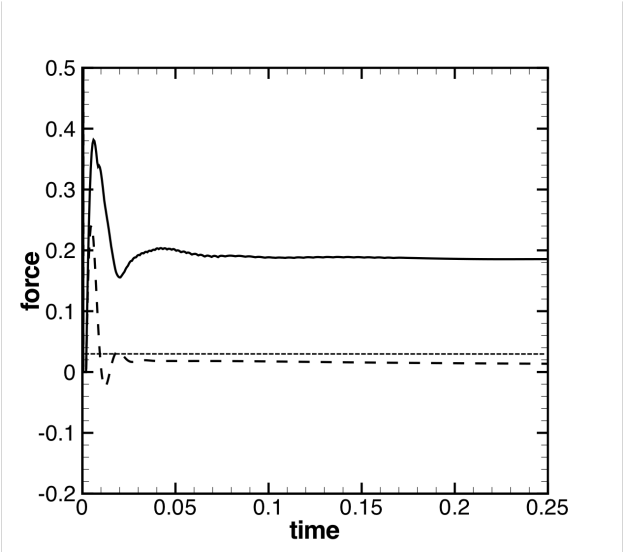


Figure 16: Temporal evolution of the total force acting on the bubble: − − − all channels are open, — the middle and bottom channel are completely closed. The forces are expressed in $[\mu N]$ and the time in $[ms]$. *The horizontal dashed line represents the threshold inertial force necessary to set the bubble in motion.*

larger than the forces necessary to hold the bubble fixed, estimated based on the surface tension forces, and thus the bubble can be removed by increasing mass flow rate in the top channel. Without any actuation (i.e. all valves are completely open), the hydrodynamic force on the bubble is insufficient to overcome the estimated surface tension forces, and thus the bubble will remain fixed inside the top channel. The goal of the controller then is to first detect flow maldistribution and then activate microvalves such that the flow rate inside the top channel is increased sufficiently to remove the bubble. This is achieved by slowly activating the microvalves in the middle and the bottom channel.

7.4 Coupled CFD-Controller Simulations

The model predictive control algorithm developed above is applied to mitigate flow maldistribution due to presence of bubbles in branches of parallel microchannels shown in Fig. 1 . Two cases are investigated to show the effectiveness of the controlled simulations: (i) presence of a single bubble in the top channel, (ii) presence of one bubble in the top and bottom channels. The history of the controller commands, evolution of the flow rates in each branch, and the trajectory of the bubble are presented below.

7.4.1 Single bubble simulation

We start with a steady-state solution for the three-channel geometry (see Fig. 1) with valve configurations set such that flow rates are equalized in branch of the channel. A single bubble of diameter 60 μm, blocking the upper channel 60% is then introduced in the upper channel. The controller sensor and actuator are activated to first sense the flow maldistribution and then provide inputs to the microvalves to regulate the flow rates in each channel such that the bubble is effectively removed.

As a sensor, we monitor the flow rates at the exit of each branch of the microchannel. Figure 17 shows the history of the mass flow rates in each branch of the microchannel starting with initial equal rates. With the presence of the bubble in the top branch, the mass-flow rate in the top channel decreases whereas those in the bottom and middle channels increase. This disparity in the flow rates is detected by the sensor, and the controller provides a series of commands to each microvalve such that equal flow rates are restored in each channel.

Figures 18a-c provide the temporal history of the controller command, the location and velocities of the bubble. As the sensor senses flow maldistribution, it starts to close the middle and bottom channels, thus increasing the flow rate in the top channel. This increase in flow increases the hydrodynamic forces on the bubble and the bubble is set in motion as soon as the hydrodynamic force exceeds a pre-calculated resistive force (due to surface tension forces) on the bubble. Once in motion, the bubble quickly acquires the velocity of the fluid flow as its Stokes number is very small. The bubble is slowly moved out of the top branch. The controller then starts to close the top channel such that the flow rates in all channels are equalized.

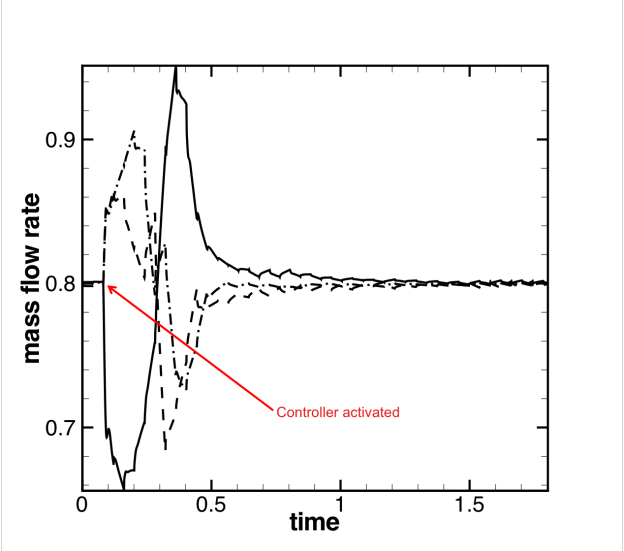
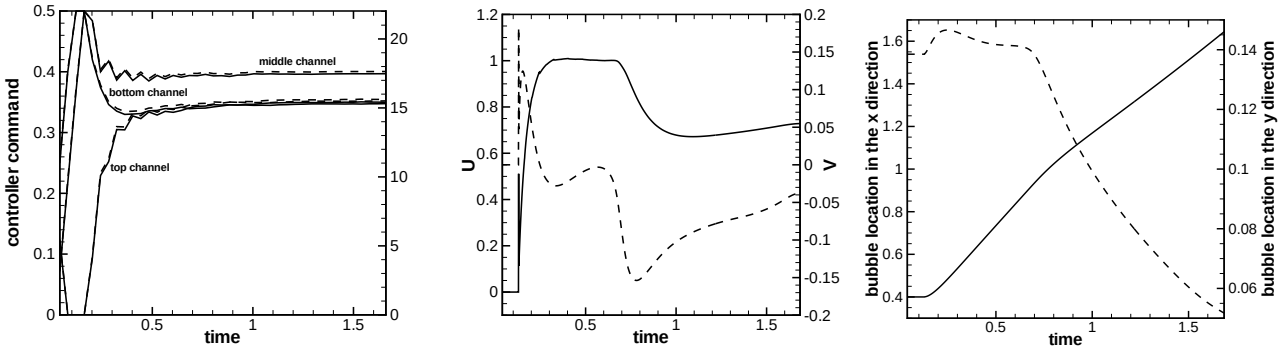


Figure 17: History of the flow in each channel with injection of the bubble at $t = 0.04\,ms$.— shows the data for the top channel; − − − middle channel; -.- bottom channel. The time is expressed in $[ms]$ and the mass flow rate in $[mg/s]$.



(a) Controller command (b) Bubble velocity (c) Bubble location

Figure 18: Temporal evolution of the bubble motion and the controller commands: 18a controller command —— history correlated with angular location −−−− for each microvalve, 18b : axial —- and vertical − − −− velocity of the bubble, and 18c axial —— and vertical − − −− location of the bubble. Time is expressed in $[ms]$, velocities are in $[m/s]$, bubble locations in $[mm]$, and angles in degrees.

7.4.2 Multiple bubble simulation

To test the controller on multiple bubbles, we introduce bubbles in the top and the bottom channels. The bubbles are placed close to the bottom walls of respective branches. The bubble in the bottom channel is further downstream as compared to the bubble present in the top channel. Figure 19 shows the pressure drop distribution inside the microchannel after the bubbles are introduced and before the controller is activated. It clearly shows large pressure drops in branches with bubbles and correspondingly the flow rates are also reduced in those channels.

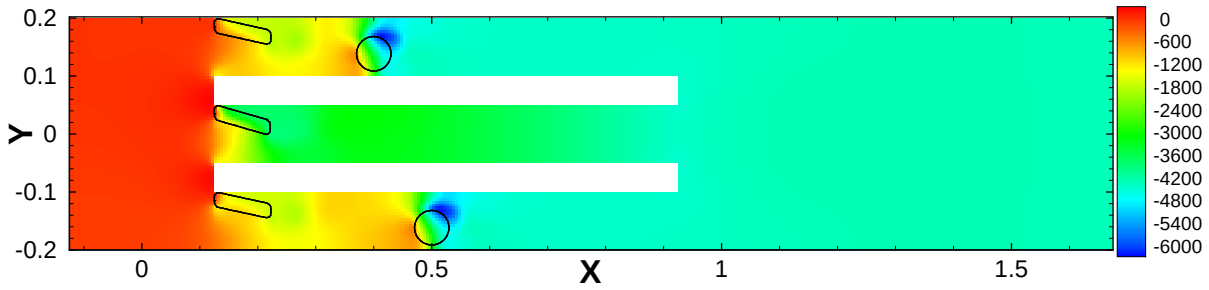
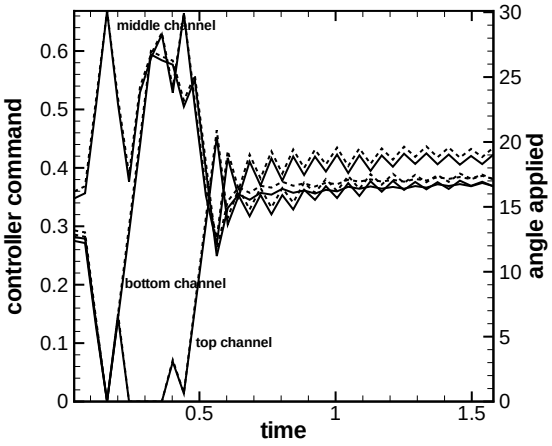


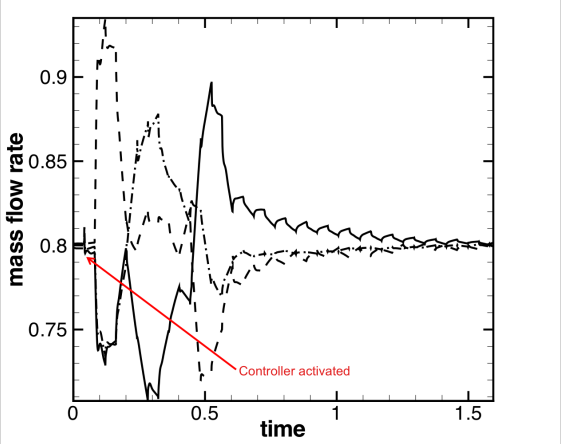
Figure 19: Pressure contours in the presence of two bubbles before the controller is activated. The pressure is expressed in $[Pa]$ and lengths are in $[mm]$.

Once the controller is activated, the sensors sense flow maldistributions; with lower flow rates in the bottom and the top channels and high flow rate in the middle channel. The controller starts closing the middle valve while keeping the others fully open. This diverts the flow in the other two channels. Figure 20a shows temporal history of the controller command correlated with the angular positions of the microvalves. Large variations in the microvalve positions are observed initially, during the initial transient after introduction of the bubbles. A time history of the flow rates in each branch of the channel is shown in Fig. 20b. With increase in flow rates in the top and bottom channels, the net force on the bubbles in respective branches also increases. Figure 20c shows the history of the force acting on the bubbles in the top and bottom channels. For this simulation, the force on the bottom bubble exceeds the threshold required to obtain an attached bubble, releasing the bubble in the bottom channel first. As soon as the bubble starts moving, it quickly reaches its terminal velocity (owing to small Stokes number) and the total force on the bubble reduces. Once the hydrodynamic forces on the bubble in the top channel overcome the estimated resistive forces, the bubble starts moving downstream. Finally, the bubbles move out of the channel branches to the exit plenum and the controller acts on all microvalves in order to equate the flow rates in each branch of the parallel microchannel.

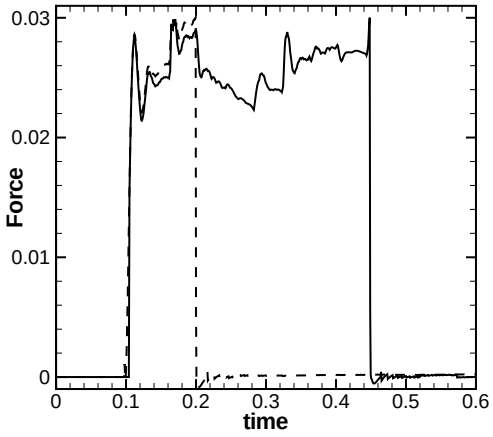
Figure 20a-c shows the velocity contours at three different stages during this coupled simulation: (i) before activation of the controller indicating flow maldistribution, (ii) after the controller is activated, the time at which the bubble in the bottom channel starts moving, and (iii) the time at which the bubble in the top channel is released. Before the controller is activated, the flow rate in the middle channel is much larger compared to the other two



(a) Controller command

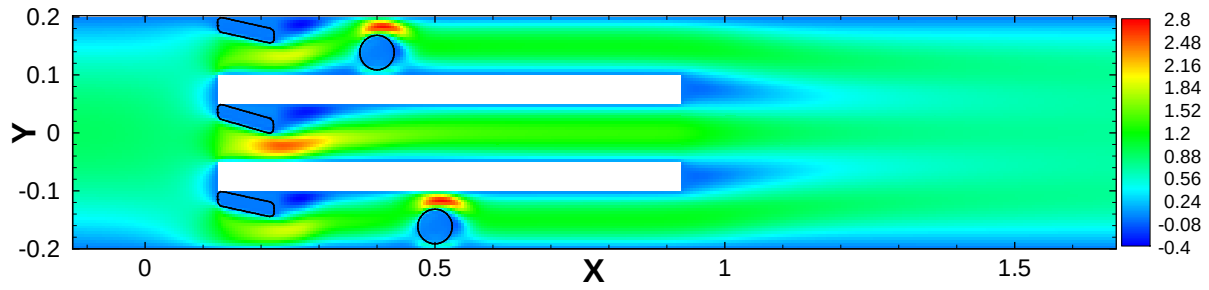


(b) Flow rates

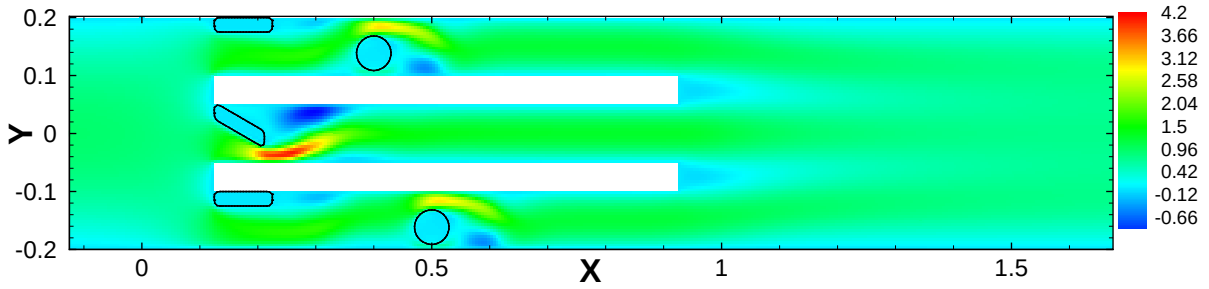


(c) Force hisotry

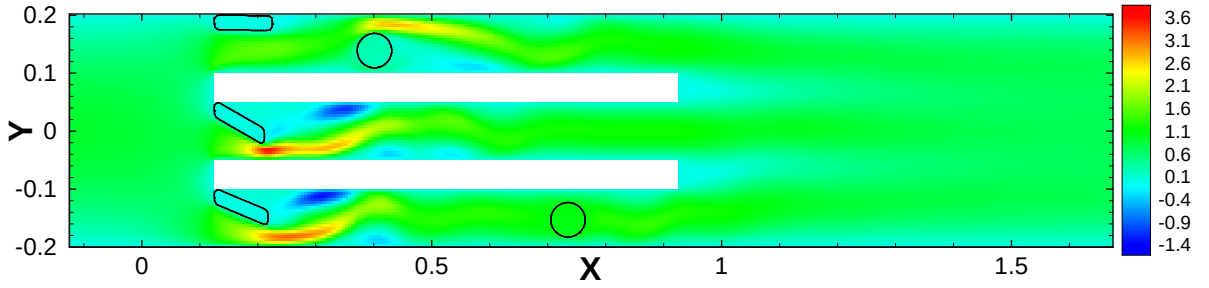
Figure 20: Temporal histories of controller command, mass flow rates and forces on the bubbles: 20a history of controller commands — correlated with angular positions of microvalves − − −,20b mass flow rates measured at the end of each channel, — top channel; − − − middle channel; -.-.- bottom channel, 20c history of total forces acting on the bubbles, — bubble initially in the top channel; − − − bubble initially in the bottom channel, Time is expressed in $[ms]$, angles in degrees, force is expressed in $[\mu N]$, and mass flow rates are expressed in $[kg/s]$.



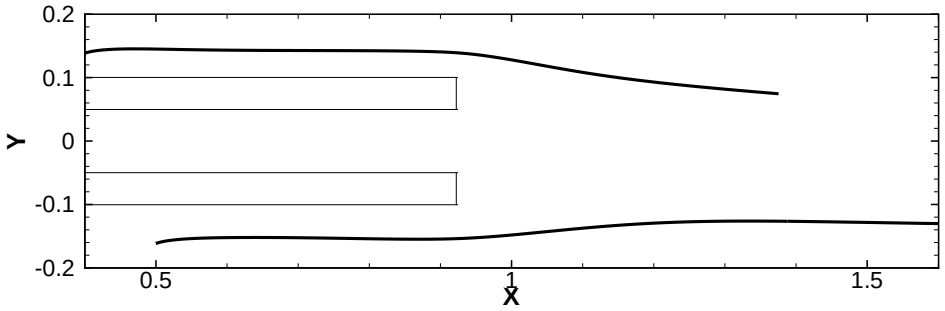
(a) Before action of the controller.



(b) Before the bottom bubble has been released.



(c) Before the top bubble has been released.



(d) Bubble trajectories

Figure 20: Temporal evolution of velocity contours and bubble trajectories in the coupled CFD-cotroller simulation for two bubbles.The lengths are expressed in $[mm]$, the velocity in $[m/s]$.

branches. Once the controller is activated, it closes the microvalve for the middle channel and fully opens the valves in the top and bottom channels. The bubbles are then set in motion and finally escape into the exit plenum, reducing the flow maldistribution. Figure 20d shows the trajectory of each bubble inside the parallel configuration.

8 Conclusion

We investigated flow maldistribution occurring in parallel microchannels due to presence of vapor bubbles in certain branches of the channel. In the present work, the bubbly flow regime consisting of small bubbles inside the microchannels was simulated at low Reynolds numbers. Neglecting any deformation of the vapor bubble, the bubbles were treated as rigid particles of density lower than the surrounding fluid. This assumption is reasonable for low Weber and Reynolds numbers. A novel fictitious-domain based direct numerical simulation approach was developed to simulate, from first principles, motion of freely moving rigid objects with large density ratios between the object and the surrounding fluid. The approach is also applicable to model flow around rigid microvalves with specified rotational motion. The numerical approach was thoroughly validated over a range of standard numerical tests for freely moving and specified motion of particles to show good agreements with available experimental data. The parallel numerical solver was integrated with a MATLAB-based model-predictive control algorithm to perform coupled CFD-controller simulations wherein the flow rates in different branches of the microchannels are controlled by activating microvalves at the entrance to the channels.

The goal of these coupled simulations was to first detect flow maldistribution in the presence of bubbles, and then eliminate any disparity in the flow rates through microvalve actuation. Accordingly, the flow rates in each branch of the microchannel were used as sensor input. System identification techniques were first employed on numerical simulations of fluid flow through the parallel microchannel in the absence of any bubbles. These studies produced a lower dimensional model that captures the essential dynamics of the full nonlinear flow, in

terms of a relationship between valve angles and the exit flow rate for each channel. A model predictive controller was then developed by utilizing this reduced order model to identify flow maldistribution from exit flow velocities and prescribe actuation of channel valves to effectively redistribute the flow. Coupled simulations were first applied to single-phase flow in three-channel geometry that equates the flow rates in each branch of the channel. The approach was then applied to two-phase flow, with artificially introduced bubbles in certain branches. The model predictive control methodology was shown to adequately reduce flow maldistribution by quickly varying channel valves to remove bubbles and equalize flow rates in each channel. The approach developed is general in the sense that it is also applicable to fouling and clogging of microchannels to presence of rigid particulates. The numerical model is efficient and simulations involving particle clusters are feasible due to its parallel implementation.

Acknowledgment

This work was supported under the US Army's Tactical Energy Systems program at Ft. Belvoir. The funding was received through the Oregon Nano and Microtechnology Institute (ONAMI) at Oregon State University. All simulations were conducted at the in-house high performance computing cluster. SVA also acknowledges Dr. Ki-Han Kim of the Office of Naval Research for the support provided under the ONR grant number N000140610697. This resulted in successful implementation of the DNS algorithm for freely moving particles.

9 Appendix

9.1 Interphase Interpolations

Any property defined at the material volumes within the particle can be projected onto the background grid by using interpolation functions. Use of simple linear interpolations may give rise to unphysical values within the particle domain (e.g. volume fractions greater than unity) [14] and may give rise to numerical oscillations in the particle velocity. In order to overcome this, a smooth approximation of the quantity can be constructed from the material volumes using interpolation kernels typically used in particle methods [31]:

$$\Phi^{\Delta}(\mathbf{x}) = \int \Phi(\mathbf{y})\xi^{\Delta}(\mathbf{x}-\mathbf{y})d\mathbf{y} \tag{28}$$

where Δ denotes grid resolution. The interpolation operator can be discretized using the material volume centroids as the quadrature points to give

$$\Phi^{\Delta}(\mathbf{x}) = \sum_{M=1}^{N} V_M \Phi(\mathbf{X}_M)\xi^{\Delta}(\mathbf{x}-\mathbf{X}_M) \tag{29}$$

where $\mathbf{X}_M$ and V_M denote the coordinates and volume of the material volumes respectively and the summation is over all material volumes for a particle. For example, in order to compute particle volume fraction, $\Phi(\mathbf{X}_M)$ will be unity at all material points. This gives unity volume fraction within the particle domain and zero outside the particle. In order to conserve the total volume of the particle as well as the total force/torque exerted by the particle on the fluid, the interpolation kernel should at least satisfy

$$\sum_{M=1}^{N} V_M \xi^{\Delta}(\mathbf{x}-\mathbf{X}_M) = 1 \tag{30}$$

$$\sum_{M=1}^{N} V_M (\mathbf{x}-\mathbf{X}_M)\xi^{\Delta}(\mathbf{x}-\mathbf{X}_M) = 0 \tag{31}$$

Several kernels with second-order accuracy include Gaussian, quartic splines etc. A kernel with compact support requiring only the immediate neighbors of a control volume has been designed and used in immersed boundary methods [17]. For uniform meshes with resolution Δ it utilizes only three points in one dimension and gives the sharpest representation of the particle onto the background mesh:

$$\xi^{\Delta}(\mathbf{x}-\mathbf{X}_M) = \frac{1}{\Delta^3}\delta\left(\frac{x-X_M}{\Delta}\right)\delta\left(\frac{y-Y_M}{\Delta}\right)\delta\left(\frac{z-Z_M}{\Delta}\right), \tag{32}$$

where

$$\delta(r) = \begin{cases} \frac{1}{6}(5-3|r|-\sqrt{-3(1-|r|)^2+1}, & 0.5 \le |r| \le 1.5, r = \frac{(x-x_0)}{\Delta} \\ \frac{1}{3}(1+\sqrt{-3r^2+1}, & |r| \le 0.5 \\ 0, & \text{otherwise.} \end{cases} \tag{33}$$

The same interpolation kernel can be used to interpolate an Eulerian quantity defined at the grid centroids to the material volume centroids. The interpolation kernel is second order accurate for smoothly varying fields [32]. The effect of these interpolations is that the surface of the particle is smoothed over the scale proportional to the kernel length. Note that in order to reduce the spreading of the interfacial region, it is necessary to use compact support as well as finer background grids and material volumes.

9.2 Updating the Particle Position

The rigid body motion (RBM) of a particle can be decomposed into translational ($\mathbf{U}^T$) and rotational ($\mathbf{U}^R$) components. The total velocity field at each point within the particle is given as

$$\mathbf{U}^{RBM} = \mathbf{U}^T + \mathbf{\Omega} \times \mathbf{r} \tag{34}$$

where $\mathbf{U}^T$ is the translational velocity, $\mathbf{\Omega}$ the angular velocity, and $\mathbf{r}$ the position vector of the material volume centroid with respect to the particle centroid. All the material volumes have the same translational velocity as the particle centroid ($\mathbf{U}^T = \mathbf{U}_P$).

Given a velocity field and the positions ($\mathbf{X}_M^0$) of the material volume centroids and the particle centroid ($\mathbf{X}_P$) at $t = t_0$, the new positions ($\mathbf{X}_M^t$) at $t = t_0 + \Delta t$ are obtained by linear superposition of the rotational and translational components of the velocity. The axis of rotation passing through the rigid body centroid $\mathbf{X}_P$ is given as $\hat{\sigma} = \mathbf{\Omega}/|\mathbf{\Omega}|$. The new coordinates due to rotation around $\hat{\sigma}$ are given as

$$\mathbf{X}' = \mathcal{R}(\mathbf{X}_M^0 - \mathbf{X}_P) + \mathbf{X}_P \tag{35}$$

where the rotation matrix is

$$\mathcal{R} = \begin{bmatrix} t\hat{\sigma}_x\hat{\sigma}_x + c & t\hat{\sigma}_x\hat{\sigma}_y - s\hat{\sigma}_z & t\hat{\sigma}_x\hat{\sigma}_z + s\hat{\sigma}_y \\ t\hat{\sigma}_x\hat{\sigma}_y + s\hat{\sigma}_z & t\hat{\sigma}_y\hat{\sigma}_y + c & t\hat{\sigma}_y\hat{\sigma}_z - s\hat{\sigma}_x \\ t\hat{\sigma}_x\hat{\sigma}_z - s\hat{\sigma}_y & t\hat{\sigma}_y\hat{\sigma}_z + s\hat{\sigma}_x & t\hat{\sigma}_z\hat{\sigma}_z + c \end{bmatrix}. \tag{36}$$

Here $c = \cos(\alpha)$, $s = \sin(\alpha)$, $t = 1 - \cos(\alpha)$, and $\alpha = |\mathbf{\Omega}|dt$. The material volume centroids are all uniformly translated to give the final positions,

$$\mathbf{X}_M^t = \mathbf{X}' + \mathbf{U}^T dt. \tag{37}$$

References

[1] Qu, W. and Mudawar, I., 2003, "Flow boiling heat transfer in two-phase micro-channel heat sinks—-i. experimental investigation and assessment of correlation methods," International Journal of Heat and Mass Transfer, **46**(15), pp. 2755–2771.

[2] Triplett, K., Ghiaasiaan, S., Abdel-Khalik, S., and Sadowski, D., 1999, "Gas–liquid two-phase flow in microchannels part i: two-phase flow patterns," International Journal of Multiphase Flow, **25**(3), pp. 377–394.

[3] Cubaud, T., 2004, "Transport of bubbles in square microchannels," Physics of Fluids, **16**(12), p. 4575.

[4] Sharp, K. and Adrian, R., 2005, "On flow-blocking particle structures in microtubes," Microfluidics and Nanofluidics, **1**(4), pp. 376–380.

[5] Wang, E., Shankar, D., Hidroo, D., CH Fogg, Koo, J., Santiago, J., Goodson, K., and Kenny, T., 2004, "Liquid velocity field measurements in two-phase microchannel convection," *3rd International symposium on Two-phase Flow Modeling and Experimentation*.

[6] Lee, P., Tseng, F., and Pan, C., 2004, "Bubble dynamics in microchannels. part i: single microchannel," International Journal of Heat and Mass Transfer, **47**(25), pp. 5575–5589.

[7] Li, H., Tseng, F., and Pan, C., 2004, "Bubble dynamics in microchannels. part ii: two parallel microchannels," International Journal of Heat and Mass Transfer, **47**(25), pp. 5591–5601.

[8] Mukherjee, A. and Kandlikar, S., 2005, "Numerical study of the effect of inlet constriction on bubble growth during flow boiling in microchannels," *3rd International Conference on Microchannels and Minichannels, ICMM2005*, vol. 75143.

[9] Bleris, L., Garcia, J., Arnold, M., and Kothare, M., 2006, "Model predictive hydrodynamic regulation of microflows," Journal of micromechanics and microengineering, **16**(9), p. 1792.

[10] Glowinski, R., Pan, T., Hesla, T., Joseph, D., and Periaux, J., 2001, "A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies- application to particulate flow," Journal of Computational Physics, **169**(2), pp. 363–426.

[11] Patankar, N., 2001, "A formulation for fast computations of rigid particulate flows," Center for Turbulence Research Annual Research Briefs 2001, pp. 185–196.

[12] Apte, S. V., Martin, M., and Patankar, N., 2008, "A numerical method for fully resolved simulation (frs) of rigid particle-flow interactions in complex flows," Journal of Computational Physics.

[13] Patankar, N., Singh, P., Joseph, D., Glowinski, R., and Pan, T., 2000, "A new formulation of the distributed Lagrange multiplier/fictitious domain method for particulate flows," International Journal of Multiphase Flow, **26**(9), pp. 1509–1524.

[14] Sharma, N. and Patankar, N., 2005, "A fast computation technique for the direct numerical simulation of rigid particulate flows," Journal of Computational Physics, **205**(2), pp. 439–457.

[15] Kim, D. and Choi, H., 2000, "A second-order time-accurate finite volume method for unsteady incompressible flow on hybrid unstructured grids," Journal of Computational Physics, **162**(2), pp. 411–428.

[16] Mahesh, K., Constantinescu, G., Apte, S., Iaccarino, G., Ham, F., and Moin, P., 2006, "Large-Eddy Simulation of Reacting Turbulent Flows in Complex Geometries," Journal of Applied Mechanics, **73**, p. 374.

[17] Roma, A., Peskin, C., and Berger, M., 1999, "An Adaptive Version of the Immersed Boundary Method," Journal of Computational Physics, **153**(2), pp. 509–534.

[18] Ferziger, J. and Perić, M., 2002, *Computational methods for fluid dynamics*, Springer New York.

[19] van der Vorst, H., 2003, *Iterative Krylov Methods for Large Linear Systems*, Cambridge University Press.

[20] Ham, F. and Young, Y., 2003, "A Cartesian Adaptive Level Set Method for Two-Phase Flows." Annu. Research Briefs, 2003: Center for Turbulence Research.

[21] Leung, L., 1999, *System Identification - Theory for the user*, Prentice Hall.

[22] Zhu, Y., 2001, *Multivariable System Identification for Process Control*, Elsevier Science.

[23] Camacho, E. and Bordons, C., 2004, *Model Predictive Control*, Springer-Verlag.

[24] Qu, W., Mudawar, I., Lee, S., and Wereley, S., 2006, "Experimental and computational investigation of flow development and pressure drop in a rectangular micro-channel," Journal of Electronic Packaging, **128**, p. 1.

[25] Dütsch, H., Durst, F., Becker, S., and Lienhart, H., 1998, "Low-reynolds-number flow around an oscillating circular cylinder at low keulegan–carpenter numbers," Journal of Fluid Mechanics, **360**, pp. 249–271.

[26] Lomholt, S., Stenum, B., and Maxey, M., 2002, "Experimental verification of the force coupling method for particulate flows," International Journal of Multiphase Flow, **28**(2), pp. 225–246.

[27] Kim, D. and Choi, H., 2006, "Immersed boundary method for flow around an arbitrarily moving body," Journal of Computational Physics, **212**(2), pp. 662–680.

[28] Mordant, N. and Pinton, J., 2000, "Velocity measurement of a settling sphere," The European Physical Journal B, **18**(2), pp. 343–352.

[29] Finn, J., 2009, "A multiscale modeling approach for bubble-vortex interactions in hydropropulsion systems," *M.S. Thesis, Oregon State University, Corvallis, OR.*

[30] Shridhar, R. and Cooper, D., 1998, "A Tuning Strategy for Unconstrained Multivariable Model Predictive Control," Industrial and Engineering chemistry research, **37**, pp. 4003–4016.

[31] Koumoutsakos, P., 2005, "Multiscale flow simulations using particles," Annual Review of Fluid Mechanics, **37**(1), pp. 457–487.

[32] Peskin, C., 2003, "The immersed boundary method," Acta Numerica, **11**, pp. 479–517.