INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2005; **48**:691–722
Published online 11 March 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.937

Large eddy simulation of turbulent flows in complex and moving rigid geometries using the immersed boundary method

Mayank Tyagi[1,‡,§] and Sumanta Acharya[2,*,†,¶]

[1]*Center for Computation and Technology, Louisiana State University, Baton Rouge, LA 70803, U.S.A.*
[2]*Mechanical Engineering Department, Louisiana State University, Baton Rouge, LA 70803-6413, U.S.A.*

SUMMARY

A large eddy simulation (LES) methodology for turbulent flows in complex rigid geometries is developed using the immersed boundary method (IBM). In the IBM body force terms are added to the momentum equations to represent a complex rigid geometry on a fixed Cartesian mesh. IBM combines the efficiency inherent in using a fixed Cartesian grid and the ease of tracking the immersed boundary at a set of moving Lagrangian points. Specific implementation strategies for the IBM are described in this paper. A two-sided forcing scheme is presented and shown to work well for moving rigid boundary problems. Turbulence and flow unsteadiness are addressed by LES using higher order numerical schemes with an accurate and robust subgrid scale (SGS) stress model. The combined LES–IBM methodology is computationally cost-effective for turbulent flows in moving geometries with prescribed surface trajectories.

Several example problems are solved to illustrate the capability of the IBM and LES methodologies. The IBM is validated for the laminar flow past a heated cylinder in a channel and the combined LES–IBM methodology is validated for turbulent film-cooling flows involving heat transfer. In both cases predictions are in good agreement with measurements. LES–IBM is then used to study turbulent fluid mixing inside the complex geometry of a trapped vortex combustor. Finally, to demonstrate the full potential of LES–IBM, a complex moving geometry problem of stator–rotor interaction is solved.

KEY WORDS: large eddy simulations; immersed boundary method; moving rigid geometry; dynamic mixed model; jet-in-crossflow; stator-rotor; trapped vortex combustor

*Correspondence to: Sumanta Acharya, Mechanical Engineering Department, Louisiana State University, Baton Rouge, LA 70803-6413, U.S.A.
† E-mail: acharya@me.lsu.edu
‡ E-mail: mtyagi@lsu.edu
§ Research Associate.
¶ Professor.

Contract/grant sponsor: AFOSR
Contract/grant sponsor: LEQSF

Received 15 January 2004
Revised 14 December 2004
Accepted 16 December 2004

INTRODUCTION

The solution of time-dependent three-dimensional Navier–Stokes equations in complex geometries is a formidable task due to the resolution requirements associated with resolving the spatial and temporal fluctuations in a turbulent flow and the difficulties in grid generation for non-Cartesian geometries. Direct numerical simulation (DNS) of turbulence is computationally intensive and is impractical for high Reynolds number flows of practical interests. Reynolds-averaged simulations have used many ingenious modelling ideas over the past several decades to simulate turbulence in a statistical sense, but their success has been limited and general models have not emerged. Large eddy simulation (LES) is a cost-effective approach to turbulence simulation in which the governing equations are spatially filtered to resolve the dynamics of the large scales, and modelling is done only for the 'universal' small scales. However LES in complex geometries introduce additional challenges due to the computational effort needed for grid generation and commutation errors introduced due to spatial filtering on non-uniform curvilinear grids. In this paper, a LES methodology for complex geometries is presented with a Cartesian-grid approach which eliminates the effort needed for curvilinear grid generation. Implementation strategies that enable the accurate resolution of moving boundaries in a cost-effective manner are described and demonstrated for selected problems.

Three key elements required for accurate and cost-effective simulation of turbulent flows in complex geometries are:

(i) High order of accuracy of the computed solution.
(ii) Accurate representation and models for turbulence.
(iii) Cost-effective grid generation for complex moving geometries.

The first requirement is addressed here by using higher order accurate finite difference schemes to solve the unsteady, filtered Navier–Stokes equations on a staggered grid. The second requirement is satisfied through LES and by employing a dynamic mixed model (DMM) for the subgrid scale (SGS) stress terms. This model can be considered to be the least common denominator to all the mathematical constraints and the physical requirements on the SGS tensor. It can represent large-scale anisotropy and back-scatter of energy from small-to-large scale through a scale-similar term and maintain the energy drain through an eddy viscosity term whose coefficient is allowed to change within the computational domain. The third requirement is satisfied by using the immersed boundary method (IBM). For complex moving rigid geometries, IBM combines the efficiency inherent in using a fixed Cartesian grid, along with the ease of tracking the immersed boundary at a set of moving Lagrangian points. Therefore, in this paper, all three requirements for cost-effective accurate computation of turbulent flows are addressed by integrating LES with IBM.

A key issue for complex geometries is the grid generation process and the effort needed in setting up a suitable grid. Although body-fitted block-grids are widely used, they generally require considerable effort, and can lead to a loss of accuracy unless special care is taken to avoid grid stretching and skewness, and to compute the metrics accurately. The grid generation effort becomes very time-consuming for moving or deforming geometries. The advantage of structured Cartesian grids over curvilinear or unstructured grids is the relative ease of the grid generation process, the lower computational effort, and the lower book-keeping requirements. Moreover, structured Cartesian grids can be used along with the IBM to simulate complex geometries. Therefore, in the present work, a Cartesian structured grid combined with the

IBM is utilized to minimize the effort needed for grid generation in complex and moving rigid geometries.

The main goal of the paper is to develop a combined LES–IBM methodology that can provide accurate and cost-effective simulation of turbulent flows in complex rigid geometries with stationary or moving boundaries. Details of the computational procedure and specific strategies developed for implementing the IBM are summarized in this paper, and described in greater detail in the dissertation by Tyagi [1]. While the advantages of the IBM for laminar flows in complex geometries has been widely reported in the literature [2–4], its use for turbulent flows has been rather limited [5–12]. In particular, the integration of LES and IBM has not been extensively studied [1, 6, 8–12]. In this paper, several forcing strategies for turbulent flow computations are explored, and in lieu of the traditional one-sided forcing strategies used for representing rigid boundaries with the IBM [5] a two-sided forcing strategy is introduced and shown to be robust for moving boundary problems. Several example problems are solved to validate the LES methodology, the implementation strategies for the IBM, and the combined LES–IBM method.

COMPUTATIONAL ISSUES

Solution algorithm

A fractional step scheme [13] is used for solving the unsteady incompressible Navier–Stokes equations. This is a time-splitting scheme where the momentum equations are split as shown in Equation (1). In the first step, an intermediate velocity field is obtained which includes the effects of the convection and diffusion terms only (Equation (1a)). In the second step, a pressure Poisson equation is solved to obtain the pressure field subject to the constraint of satisfying mass continuity (Equation (2)). As the last step of the algorithm (projection step), the velocity field is updated using this pressure field and the intermediate velocity field (Equation (1b)). Equations (1) and (2) are given by

$$\frac{\tilde{u}-u^n}{\Delta t}=\frac{3}{2}(C^n+D^n)-\frac{1}{2}(C^{n-1}+D^{n-1}) \tag{1a}$$

$$\frac{u^{n+1}-\tilde{u}}{\Delta t}=-\nabla p^{n+1} \tag{1b}$$

where the convective terms are represented by C and the diffusion terms are represented by D.

$$C=-(u\cdot\nabla)u, \quad D=\frac{1}{Re}\nabla^2 u$$

The pressure Poisson equation (Equation (2)) is obtained by taking the divergence of the second step (Equation (1b)) and enforcing the continuity condition for the velocity field at

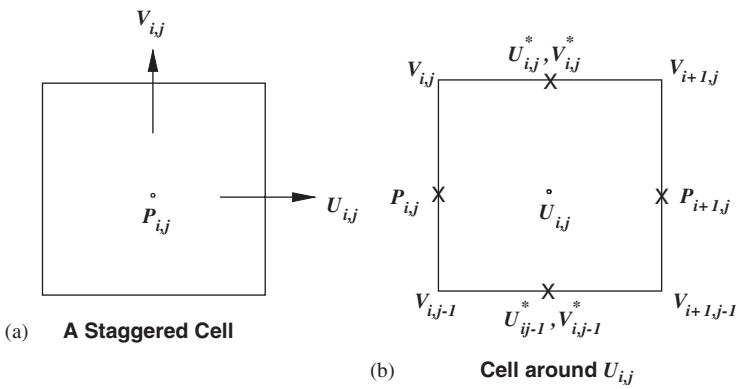


Figure 1. (a) Schematic of a staggered cell in two dimensions; and (b) computational cell and arrangement of variables in x-momentum equation.

the next time step ($\nabla \cdot u^{n+1} = 0$):

$$\nabla^2 p = \frac{\nabla \cdot \tilde{u}}{\Delta t} - \frac{\nabla \cdot u^{n+1}}{\Delta t} = \frac{\nabla \cdot \tilde{u}}{\Delta t} \tag{2}$$

The boundary conditions for the pressure, for given inflow and outflow conditions, are specified by a Neumann condition. However, solution of the Poisson equation subject to Neumann boundary conditions lacks existence and uniqueness. It has a solution only if the compatibility condition is satisfied [14]. The discrete operators for the Laplacian in the Poisson equation are therefore subject to such a constraint.

Differencing schemes

In the present paper, a staggered grid arrangement (Figure 1(a)) is used to avoid grid level pressure oscillations. Higher order accurate finite difference schemes are used for the convection and diffusion terms to maintain the numerical accuracy required for DNS and LES [15–17]. As shown in Equation (1a), the temporal discretization is based on a second-order accurate explicit Adams–Bashforth scheme.

The spatial discretization of the viscous terms in the momentum equations is done using a fourth-order accurate central difference approximation for the second derivative [18].

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j,k} = \frac{-u_{i-2,j,k} + 16u_{i-1,j,k} - 30u_{i,j,k} + 16u_{i+1,j,k} - u_{i+2,j,k}}{12\Delta x^2} \tag{3}$$

The order of accuracy is reduced to second order near the boundaries. The discretization of the convective terms in the x-momentum equation is done using a conservative formulation and a fourth-order central difference scheme for the cross-stream derivatives using interpolated velocity components (u^*, v^* and w^*) that are evaluated using fifth-order accurate upwinding schemes (Figure 1(b)). The resulting expression for the discretized convective term using the

fourth-order accurate centred approximation is given by

$$\left(\frac{\partial uv}{\partial y}\right)_{i,j,k} = \frac{u^*_{i,j+1,k}v^*_{i,j+1,k} - 27u^*_{i,j,k}v^*_{i,j,k} + 27u^*_{i,j-1,k}\,v^*_{i,j-1,k} - u^*_{i,j-2,k}\,v^*_{i,j-2,k}}{24\Delta x} \tag{4}$$

The streamwise convective term in the x-momentum equation is discretized in its non-conservative form with a third-order upwinding scheme using the computed velocities at the staggered u-grid points. Thus, for all interior points, there are two nodes upstream and one node downstream of the grid point at which the derivative is evaluated.

$$\begin{aligned}\left(\frac{\partial uu}{\partial x}\right)_{i,j,k} &= 2u_{i,j,k}\left(\frac{\partial u}{\partial x}\right)_{i,j,k} \\ &= \begin{cases} 2u_{i,j,k}\,\dfrac{u_{i-2,j,k} - 6u_{i-1,j,k} + 3u_{i,j,k} + 2u_{i+1,j,k}}{\Delta x} & (u_{i,j,k} > 0) \\ 2u_{i,j,k}\,\dfrac{-2u_{i-1,j,k} - 3u_{i,j,k} + 6u_{i+1,j,k} - u_{i+2,j,k}}{\Delta x} & (u_{i,j,k} < 0) \end{cases}\end{aligned} \tag{5}$$

The stencil size is decreased near the boundary points to retain second-order accuracy. However, the stencil weights can be generated on any arbitrary grid spacing for the desired derivative to any order of accuracy following Fornberg [18].

Additional details of the differencing schemes including the details of time-step restriction due to the CFL stability criterion can be found in Reference [19].

The pressure Poisson solver

The numerical solution of the Poisson equation is the most computationally demanding step of the algorithm. It would be highly desirable to have a fast, efficient and robust solver for such a system of equations. For flows having one homogeneous or periodic direction, a spectral decomposition in that direction reduces the three-dimensional Poisson equation into a set of two-dimensional Poisson equations which can be directly solved using a matrix diagonalization method. Such an approach is adopted in the present work.

The Poisson equation can be expressed as

$$\left(\frac{\partial^2 p}{\partial x^2}\right)_{i,j,k} + \left(\frac{\partial^2 p}{\partial y^2}\right)_{i,j,k} + \left(\frac{\partial^2 p}{\partial z^2}\right)_{i,j,k} = g_{i,j,k} \tag{6}$$

Without loss of generality, treating the z direction as the homogeneous direction, and taking the Fourier transform in that direction, the discrete Poisson equation can be written as [19]

$$\left(\frac{\partial^2 \hat{p}}{\partial x^2}\right)_{i,j,k} + \left(\frac{\partial^2 \hat{p}}{\partial y^2}\right)_{i,j,k} + f(k)\hat{p}_{i,j,k} = \hat{g}_{i,j,k} \tag{7}$$

where the function $f(k)$ depends on the wavenumber k. If the discrete operators $\partial^2/\partial x^2$ and $\partial^2/\partial y^2$ are represented by the matrices X and Y, the discretized equations can be written in a matrix form as

$$\begin{aligned} X\hat{P}(k) + \hat{P}(k)Y + f(k)I\hat{P}(k) &= \hat{G}(k) \\ X\hat{P}(k) + \hat{P}(k)Y'(k) &= \hat{G}(k) \end{aligned} \tag{8}$$

where $Y'(k) = Y + f(k)I$. The matrix diagonalization of X and $Y'(k)$ leads to

$$X = P_x D_x P_x^{-1}, \quad Y'(k) = P_y(k) D_y(k) P_y^{-1}(k) \tag{9}$$

Multiplying the Poisson equation (8) with P_x^{-1} and $P_y(k)$ results in

$$\begin{aligned} D_x \hat{P}'(k) + \hat{P}'(k) D_y(k) &= \hat{G}'(k) \\ \hat{P}'(k) &= P_x^{-1} \hat{P}(k) P_y(k) \\ \hat{G}'(k) &= P_x^{-1} \hat{G}(k) P_y(k) \end{aligned} \tag{10}$$

Now, the eigenvalues of matrices X and $Y'(k)$ can be used to determine the pressure field as follows:

$$\begin{aligned} \hat{P}'_{i,j}(k) &= \frac{\hat{G}'_{i,j}(k)}{\lambda_{xi} + \lambda_{yj}(k)} \\ \hat{P}(k) &= P_x \hat{P}'(k) P_y^{-1}(k) \end{aligned} \tag{11}$$

As the last step, the inverse Fourier transform is applied to get the pressure field. In the current implementation of the discrete operator in the Poisson equation ($\nabla \cdot \nabla p$), the 4–2 formulation is used, i.e. the gradient operator is discretized with a fourth-order accurate centred approximation and the divergence operator is represented by a second-order accurate centred scheme [14]. For calculation of the FFTs, the Compaq extended math library (CXML) subroutines are used. For the calculation of eigenvalues of operators and the inverses of various matrices, subroutines available at an internet repository are used (http://www.netlib.org).

LES METHODOLOGY

In LES, the governing equations are spatially filtered, with the filter width (proportional to the size of each grid element) representing the scales in the flow field that are resolved [1, 9–12, 16]. The non-dimensional filtered governing equations for the conservation of mass and momentum for an incompressible Newtonian fluid are given as

$$\begin{aligned} \frac{\partial U_j}{\partial x_j} &= 0 \\ \frac{\partial U_i}{\partial t} + \frac{\partial U_i U_j}{\partial x_j} &= -\frac{\partial p}{\partial x_i} + \frac{1}{Re} \frac{\partial^2 U_i}{\partial x_j^2} + \frac{\partial \tau_{ij}}{\partial x_j} + f_i \end{aligned} \tag{12}$$

where U_i is the filtered velocity field and f_i is the body force term. With the IBM, grid points internal to a solid surface (or in its vicinity depending on the forcing strategy) have body force terms added such that the no-slip boundary condition at the immersed surface is satisfied [2, 5].

The SGS stress tensor is represented by τ_{ij} in the governing equation. In this study, a DMM is used to model the SGS stress tensor [20, 21]. The DMM can represent the backscatter of energy through the scale-similar part while it can drain the energy from the large

scales to the small scales using an eddy viscosity part. This model is generally considered to be the simplest model that satisfies both the physical and mathematical requirements for SGS models. Box filters are used in the Germano identity for the calculation of the dynamic coefficient and for the calculation of Leonard stresses appearing in the subgrid stress term. There are several approaches for the calculation and smoothing of dynamic coefficient. The assumption of locally constant model coefficient is not accurate and can lead to large variations in space as well as time for the model coefficient. Alternatively, retaining the dynamic coefficient inside the filters in Germano identity leads to a Fredholm integral equation (dynamic localization model [22]) that is computationally expensive to solve. Pomraning and Rutland [23] studied the solvability of such Fredholm integral equations and proposed new one-equation model. Further, Piomelli and Liu [24] have proposed an iterative scheme for the computation of dynamic coefficient without relying on the locally constant assumption. Meneveau *et al*. [25] proposed Lagrangian averaging along the pathlines in complex geometries. Even with all the mathematically consistent rationales, there is inevitably the need for numerical clipping procedures with all the above-stated coefficient calculation procedures. In this study, the dynamic coefficient is test filtered to avoid numerical instabilities. It should be considered as the cheap alternative for all other mathematically sophisticated procedures. The use of high-order accurate schemes in conjunction with explicit filtering and a robust SGS model provides a mathematically consistent and physically realizable methodology that is the basis for accurate representation of turbulent flows in the present work. Additional details of the procedure are given in Reference [1].

IBM

The IBM concept

While LES can formally alleviate the issue of cost-effectively resolving the turbulent flow dynamics in a high Reynolds number flow, complex geometries with structured boundary-fitted grids introduce the problem of commutation errors. Moreover, the representation of moving geometries using either sliding meshes or regenerating the mesh at each time step requires significant computational effort. IBM relies upon utilizing the body force terms added in the momentum equations to represent the geometry (rigid or flexible) on a fixed Cartesian mesh [2–7]. This formulation is simple and ideally suited for the moving geometries involving no-slip walls with prescribed trajectories and locations. However, as noted earlier, the IBM method has been adopted primarily for laminar flow problems, and its use to resolve turbulent flows in complex geometries has been rather limited. Mathematical details of the IBM are presented succinctly by Peskin [26]. Beyer and Leveque [27] presented analysis of a one-dimensional model for the IBM. Lee and Leveque [28] developed an immersed interface method for sharp resolution of pressure across moving interface and better volume conservation than traditional IBM. Also, a review of immersed boundary technique for turbulent flows is presented by Iaccarino and Verzicco [11]. Grigoriadis *et al*. [10] and Balaras [9] used the IBM in conjunction with LESs. Lima E Silva *et al*. [29] presented an approach to evaluate the Navier–Stokes equation for force calculation at Lagrangian immersed points followed by the redistribution of forces on the Eulerian mesh. Cortez and Minion [30] extended the concept of IBM using blob projection method. Cottet and Poncet [31] used the

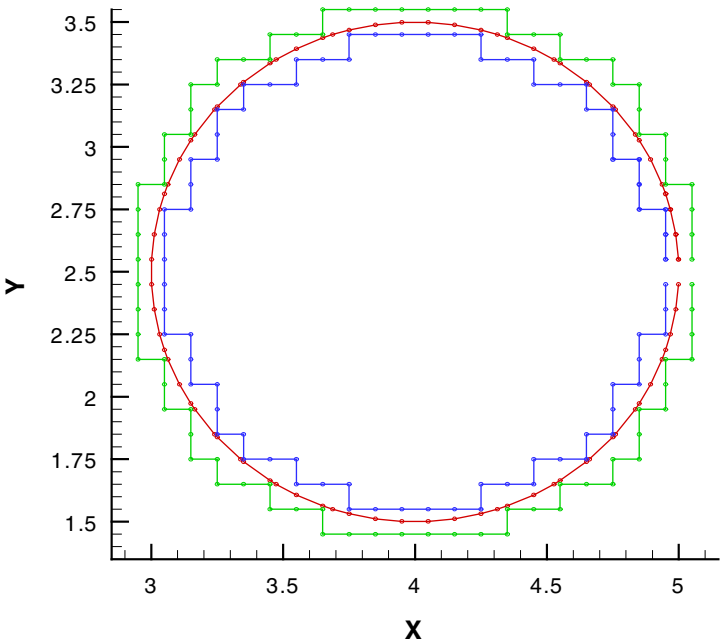


Figure 2. Identification of the circular boundary on uniform 2-D Cartesian mesh (red points are the surface points, green points are solved using governing equations and blue points are forced using appropriate interpolation stencils).

IBM with vortex methods to simulate the impingement of a vortex ring on a circular cylinder while maintaining no-slip and no-penetration conditions on immersed boundaries. Kirkpatrick *et al*. [32] detailed a second-order accurate representation of curved geometries on staggered grids. Tseng and Ferziger [12] used a similar approach to present a ghost fluid formulation of the IBM. Vikhansky [33] modified the interpolation technique using the D'Alembert principle to model no-slip boundaries without explicit calculation of forces. Russell and Wang [34] used the streamfunction–vorticity formulation for modelling moving objects in 2-D incompressible viscous flows.

In the IBM, the complex geometrical features are incorporated by adding a forcing function in the governing equations. The forcing function is zero everywhere except in the vicinity of the surface where the influence of the solid boundaries is assigned (subscript Γ). To explain the concept of the IBM, a circle on a two-dimensional uniform grid is used as an example here (Figure 2). The grid points interior and exterior to the circle are identified and then paired. For internal (external) flows, the boundary conditions (forcing) are generally applied to the exterior (interior) points. Appropriate interpolation stencils can be formed using the points solved for in the flow field, the point inside the immersed object (forced) and the point at the immersed interface Γ. Several ways to achieve this forcing are described later and their implications in different scenarios are discussed. Also, for moving geometries, these forcing terms are prescribed in a time-dependent fashion. On a fixed Cartesian mesh with geometry defined on moving Lagrangian points (location of forcing), the need for time-consuming grid

generation methods is therefore not required. However, the intersection points of the immersed boundary with the underlying Cartesian mesh need to be computed in an efficient fashion.

The computed velocity field needs to be consistent with the no-slip requirement at the geometric features of the immersed solid object. As a first step, the exact location of the intersection points between the geometry and the Cartesian grid need to be solved. In general, these locations will not coincide with the computational grid nodes. The weights can now be evaluated and the velocity at the forced point determined such that when the forced and the computed velocities are interpolated to the surface the no-slip condition is satisfied. In the IBM, a body force term containing the velocity at the forced point appears in the momentum balance equation for the forced points (Equation (13)). Thus, the influence of the complex geometric features is distributed on the computational mesh through these body force terms obtained through appropriate interpolation. Also, such an influence is limited to a single mesh spacing Δ to preserve the sharp geometric features. Thus, in the fractional step scheme, the momentum equations are re-written as

$$\begin{aligned}
\frac{\tilde{u}-u^n}{\Delta t} &= \frac{3}{2}(C^n+D^n)-\frac{1}{2}(C^{n-1}+D^{n-1})+f \\
f &= \left[\frac{u_\Gamma^{n+1}-u^n}{\Delta t}-\frac{3}{2}(C^n+D^n)+\frac{1}{2}(C^{n-1}+D^{n-1})\right]\delta(\mathbf{x}-\mathbf{x}_\Gamma) \qquad (13) \\
\frac{u^{n+1}-\tilde{u}}{\Delta t} &= -\nabla p^{n+1}
\end{aligned}$$

where f represents the forcing term and u_Γ is the interpolated velocity at the forced point. The delta distribution function in Equation (13) denotes that the immersed boundary contribution arise only at the specified boundary and is zero everywhere else. It is noted here that for a general class of deformable boundary problems, a fluid–structure interaction problem must be solved [2]. Taking the divergence of the projection step and enforcing the continuity condition for the velocity field at the next time step, we obtain the following modified pressure Poisson equation:

$$\nabla^2 p = \frac{\nabla\cdot\tilde{u}}{\Delta t}-\nabla\cdot f \qquad (14)$$

Thus a key step in the IBM is the determination of the forcing function f which, in turn, depends on the interpolated velocity u_Γ at the forced point.

The forcing strategies

Forcing at the immersed points can be achieved in different ways, and the behaviour of the forcing scheme (interpolation) varies as the immersed point is located closer and closer to the solid interface. In turbulent flows, for a well-resolved simulation, the nearest grid point from the immersed surface should be in the viscous sublayer. The use of linear interpolation schemes for the tangential components of velocity can therefore be justified (law of the wall). However, for physical consistency near the surface, the normal component of velocity to the immersed surface should satisfy the continuity condition. As explained below, this can be achieved with a quadratic interpolation scheme which also provides greater accuracy.

Four different forcing/interpolation strategies are described below and discussed. Case A corresponds to one-sided forcing, and represents the common practice for representation of

rigid geometries using IBM in the literature. Case B represents two-sided forcing and is proposed here for moving boundary problems and for problems with complex geometrical features. As discussed below, the Case A approach introduces large errors when the immersed surface is located close to the neighbouring point in the flow field (a likely possibility for moving boundaries or in complex geometries), while the proposed Case B approach remains numerically robust for all configurations tested including with moving boundaries. Case C provides a strategy for higher order accuracy (quadratic interpolation instead of linear) and for ensuring that the normal velocity component near the surface is chosen such that mass conservation is better satisfied near the immersed surface. Case D provides the details of a strategy based on inverse-distance weighted interpolation recently advocated by several researchers [11, 12].

Case A: Forcing on only one side (inside the solid) of the immersed boundary. Let Δ be the mesh spacing and δ be the distance of the forced point from the immersed surface (Figure 3(a)). A linear extrapolation utilizing the point on the immersed surface (with specified boundary velocity) and the point just outside the solid (with computed velocity) is then used to obtain the velocity at the forced point inside the solid. Let V_d be the desired velocity at the point on the immersed surface and V_c be the computed flow velocity outside the solid surface. Therefore, the velocity at the forcing point V_{im} is given by

$$V_{im} = V_d \Delta/(\Delta - \delta) - V_c \delta/(\Delta - \delta) \tag{15}$$

Clearly, in the limit δ going to zero, i.e. the forcing point approaching the point on the immersed surface, we retrieve the limit V_{im} approaching V_d. However, in the limit δ approaching mesh spacing Δ (i.e. the immersed surface is located close to the neighbouring node in the flow), we have V_c approaching V_d, and the difference $(\Delta - \delta)$ becomes extremely small. Under these conditions, V_{im} is ill-defined because of very large coefficients, and leads to significant round-off errors and inaccurate representation of V_{im}. For a static geometry and given mesh resolution, it is possible to minimize such ill-conditioned forced immersed points through proper grid-point distribution. However, for complex geometries and moving boundaries δ will change with the surface motion and the above-noted ill-conditioned situations cannot be avoided with Equation (15). Thus, a new forcing strategy is needed.

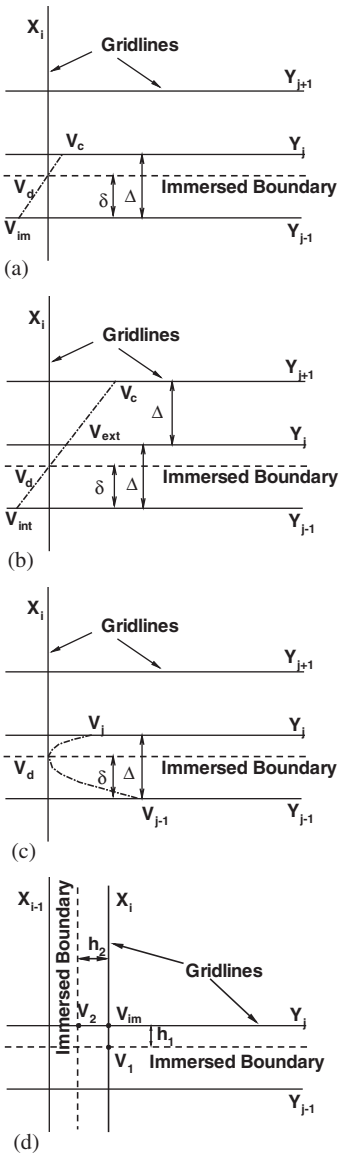
Case B: Forcing on both sides of the immersed boundary. To avoid the problem of ill-conditioned weights with one-sided forcing, the immersed boundary forcing can be applied to both the interior grid node (δ from the surface) and the first exterior grid node that bracket the point on the immersed surface (Figure 3(b)). Thus the linear interpolation/extrapolation is done using the specified surface velocity (V_d) and the computed flow velocity at the second grid node from the surface (V_c). The resulting velocity at the internal forcing point V_{int} is given by

$$V_{int} = V_d 2\Delta/(2\Delta - \delta) - V_c \delta/(2\Delta - \delta) \tag{16}$$

Similarily, the velocity at the external forcing point V_{ext} is given by

$$V_{ext} = V_d \Delta/(2\Delta - \delta) + V_c(\Delta - \delta)/(2\Delta - \delta) \tag{17}$$

Clearly, in the limit δ going to zero, i.e. the internal forcing point approaching the point on the immersed surface, we retrieve the limit V_{int} approaching V_d and V_{ext} approaching



$V_{im} = V_d\,\Delta/(\Delta - \delta) - V_c\,\delta/(\Delta - \delta)$

V_c : Computed Velocity
V_d : Desired Velocity
V_{im} : Immersed Velocity
Δ: Grid Spacing
δ: Location of Immersed surface from Forced point

$V_{int} = V_d\,2\Delta/(2\Delta - \delta) - V_c\,\delta/(2\Delta - \delta)$
$V_{ext} = V_d\,\Delta/(2\Delta - \delta) + V_c\,(\Delta - \delta)/(2\Delta - \delta)$

$V_{J-1} = V_J\,((\Delta-\delta)^2/\delta^2) + V_d\,(\Delta(2\Delta-\delta)/\delta^2)$

$V_{im} = (V_1/h_1+V_2/h_2)/(1/h_1+1/h_2)$

V_1,V_2 : Desired Velocities at the immersed points
V_{im} : Forced Velocity using interpolation
h_1,h_2 : Distances between immersed surfaces and forced point

Figure 3. Interpolation strategies for immersed boundary forcing.

$(V_d + V_c)/2$. In the limit δ approaching mesh spacing Δ, we obtain V_{ext} approaching V_d, and V_{int} becoming $(2V_d - V_c)$ as it should be by reflection condition. Thus, the forcing remains physical for all positions of the immersed surface between the grid interfaces. For moving boundary implementation, this is particularly important and avoids ill-conditioned interpolation/extrapolation. As the moving boundary crosses the grid interfaces, these extreme limits will be approached and are always well-defined as compared to one-sided interpolation strategies that may become ill-conditioned.

Case C: Forcing on both sides of the immersed boundary with asymptotically consistent near-wall behaviour. The forcing at the immersed point (Case A or B above) is designed to satisfy the no-slip boundary condition. However, it does not guarantee mass conservation in the fluid region adjacent to the immersed surface. The no-slip condition implies that $(\partial u/\partial x)=0$ on the immersed surface (assuming coordinate system aligned with the surface). Therefore, the normal component satisfies a Neumann condition for mass conservation to be satisfied, i.e. $(\partial v/\partial y)=0$ on the surface. The key modification required to enforce this condition is to modify the forcing of the velocity component normal to the surface at the forced fluid point. All other velocities can be forced using the approach described for either Case A or B above.

Consider an immersed boundary between grid points J (fluid point) and $J-1$ (point inside the solid surface). The distance of point $J-1$ from the immersed point is δ and the grid resolution is Δ (Figure 3(c)). A quadratic profile is generated between the J, $J-1$ and the immersed surface with the conditions:

- $y=0,\ V=V_J$
- $y=\Delta-\delta,\ V=V_S$ and $\partial V/\partial y=0$ (simplified continuity equation on the wall)
- $y=\Delta,\ V=V_{J-1}$

Solving for V_{J-1}, we get

$$V_{J-1}=V_J((\Delta-\delta)^2/\delta^2)+V_S(\Delta(2\delta-\Delta)/\delta^2) \tag{18}$$

Checking the limiting behaviour for different locations of the immersed surface between the grid points, we get:

I. The forced point approaches the immersed surface ($\delta\to\Delta$)

$$V_{J-1}\to V_S$$

II. Immersed point is in the centre of the grid cell ($\delta\to\Delta/2$)

$$V_{J-1}\to V_J \text{ (symmetry condition is satisfied naturally)}$$

III. Immersed point approaches the solved grid point ($\delta\to 0$, thus $V_J\to V_S$)

$$V_{J-1}\to V_J \text{ (or } V_S)$$

Thus, Equation (18) satisfies all the expected limiting behaviour for different immersed point locations between the grid points. In addition, the simplified form of the continuity equation is satisfied. A similar treatment of satisfying the continuity conditions around cells containing immersed boundaries is presented by Kellog [7]. Kim *et al.* [8] add source terms in the continuity equation for a finite volume method to eliminate any sources or sinks created by the imposition of forcing at immersed boundary points in the momentum equations.

Case D: Forcing using the inverse-distance weighting schemes. Several researchers [11, 12] have used the inverse-distance weighted interpolation to evaluate the body forces on the immersed grid points. A general interpolation scheme can be written as

$$V_{\text{im}}=\sum_{m=1}^{n} w_m V_m/q,\quad w_m=\left(\frac{R-h_m}{Rh_m}\right)^p,\quad q=\sum_{i=1}^{n}\left(\frac{R-h_i}{Rh_i}\right)^p$$

where V_m are the computed solution around immersed point, w_m is the weight, p is the power exponent (usually set to 2), h_m is the distance of corresponding grid points from the forced point, R is the maximum of h_m. Figure 3(d) presents a scenario of forcing around corners using another inverse-distance weighted scheme.

Use of schemes presented as Cases C and D is deferred for the future research work.

VALIDATION STUDIES

IBM validation

Case 1: Laminar flow past a heated cylinder in a plane channel. A uniform Cartesian grid of $172 \times 132 \times 41$ points is used for a domain of the size $8.5D \times 6.5D \times 4.0D$, where D is the diameter of the cylinder. The IBM forcing was done with one-sided interpolation (Case A) due to the relative simplicity of the geometry. Several global parameters (drag coefficient, Strouhal number and Nusselt number) were calculated and shown to compare well (within 5%) with published experimental or theoretical results (Table I). This excellent agreement with published data validates the implementation of the IBM. The separation points at (80–82°) are also observed (Figure 4). The flowfield inside the cylindrical surface see in Figure 5 is a reflection of the IBM where calculations are done for both interior and exterior points. However, the key interior region of interest is represented by the interior points adjacent to the boundary that are forced with the Case A approach in order to satisfy the no-slip condition on the curved surface.

A sequence of snapshots is presented for a full vortex-shedding cycle corresponding to the dominant Strouhal frequency of 0.283 (with time period T) in Figure 5(a)–(e). The non-dimensional temperature contours range from zero (blue) to one (red). At time instant t_0, the upper vortex is beginning to shed into the wake. The lower vortex is in the beginning stages of roll-up. At time instant $t_0 + \tau$ (where $\tau = T/4$), the lower vortex has convected further down into the wake region, and the temperature of the upper vortex has diffused. At $t_0 + 2\tau$, the pattern looks like the mirror image of the first instant and reveals that the time instance is around the half-period of the complete shedding cycle. At later time instants, these vertical structures convect down into the wake region, and the decrease in their temperature levels is primarily due to diffusion. The shedding cycle is complete at $t_0 + 4\tau$ (frames 1 and 5 are nearly identical).

Table I. Comparison of experimental and computed values of drag coefficients, Nusselt number and Strouhal number.

	Computed	Theoretical or experimental
C_{Dp} pressure drag C_{Df} friction drag	0.620 0.593 } 1.213	1.2 (total drag)*
Nusselt number	5.45	5.21 ($\pm$ 20%)†
Strouhal number	0.283	0.281–0.287*

*Reference [35].
†Reference [36].

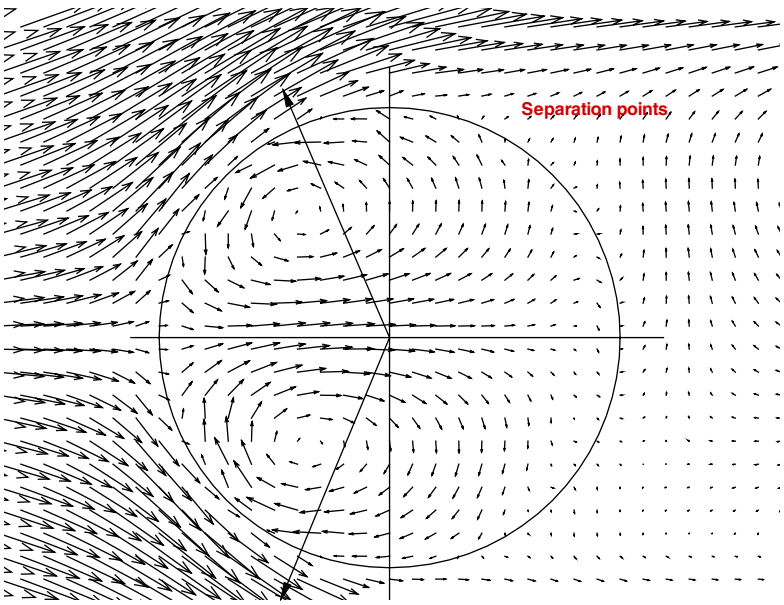


Figure 4. Separation points in the flow over circular cylinder (80–82°).

Case 2: Laminar flow across tube bundle. Laminar flow in a staggered arrangement of 95 cylinders (Figure 6) is solved to demonstrate the advantage of IBM over conventional body-fitted grids. A body-fitted grid generation procedure would take a considerable level of effort to produce a corresponding grid. In contrast, generating a Cartesian grid is trivial. The only complexity is that of defining the intersection points between the individual cylinders and the Cartesian grid points. This calculation and the determination of the appropriate interpolation stencils was done in a preprocessor step. The forcing schemes took a very small fraction of the total computational time. The generation of body fitted grids, and the calculation of the appropriate metrics would take considerably greater time.

In the present problem and all subsequent problems, the two-sided forcing approach (Case B) is used unless otherwise specified. If the 95 cylinders were arranged in a complex non-uniform configuration, it would be impossible to ensure the grid distribution near each cylinder to be such that the ill-conditioned situations discussed earlier would not arise. While generating a Cartesian grid (potentially non-uniform) to avoid the ill-conditioned interpolation is possible, it would require some trial and error calculations, and the goal of the present work is to minimize computational effort pertaining to grid generation. Thus, the two-sided forcing is henceforth adopted.

For the problem considered, the Reynolds number based upon the average flow inlet velocity and the cylinder diameter is 20. A grid resolution of 250×250 was chosen to discretize the domain under consideration. Pressure contours and the close-up of velocity field around the cylinders in the middle of the tube bundle are presented in Figure 6(a) and 6(b), respectively. The computed pressure drop across the tube bundle is in agreement with the numerical study of Ye *et al.* [37].

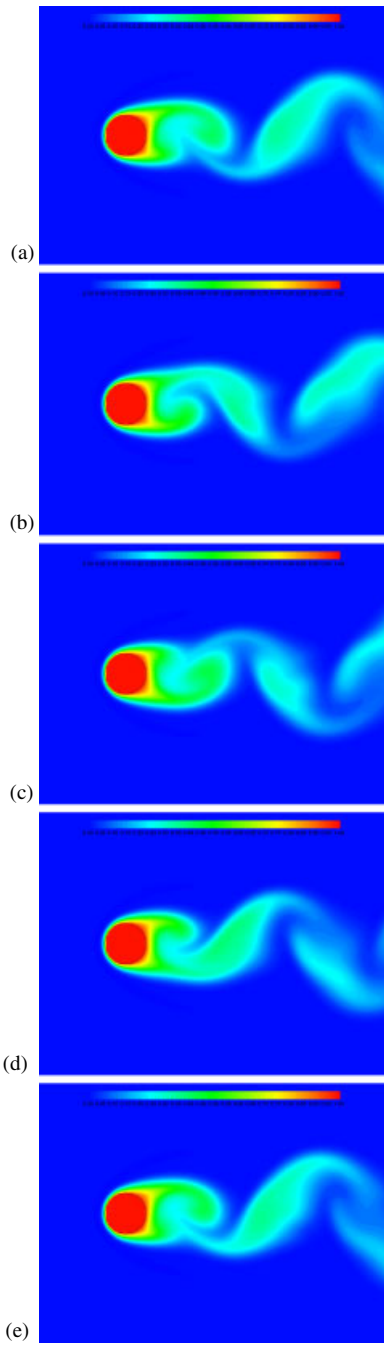


Figure 5. Five frames from vortex shedding cycle of flow past a heated cylinder in crossflow at $Re = 100$: (a) $t = t_0$; (b) $t = t_0 + \tau$; (c) $t = t_0 + 2\tau$; (d) $t = t_0 + 3\tau$; and (e) $t = t_0 + 4\tau$.

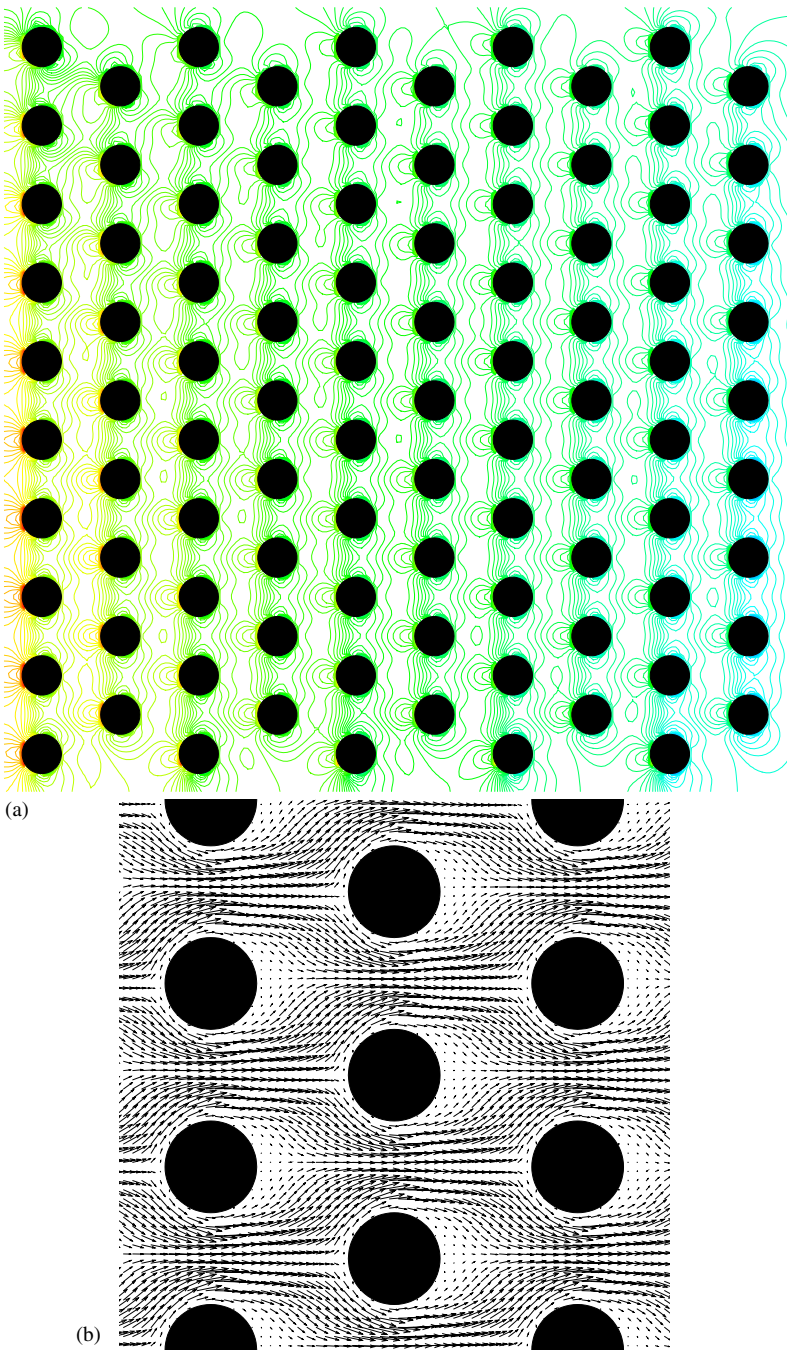


Figure 6. (a) Pressure contours around 95 staggered arrangement tube bundle; and (b) close-up of the velocity vectors around a few cylinders in the centre of tube bundle.

LES validation

Case 3: Lid-driven cavity flow. In a lid-driven cavity with spanwise aspect ratio of 0.5 at a Reynolds number of 10 000, the flow is partly laminar in the cavity and strongly turbulent along the downstream and bottom walls of the cavity. Therefore, it forms a good benchmark case to validate the LES procedure as well as the SGS model. Deshpande and Milton [38] and Leriche and Gavrilakis [39] performed DNS of lid-driven cavity flows. Zang [40] performed LES of the same configuration. A uniform Cartesian grid of $64 \times 64 \times 32$ points is used for a domain of the size $1D \times 1D \times 0.5D$, where D is the height of the cavity. Computations (Figure 7) show good agreement with the experimental data of Prasad and Koseff [41] as well as other numerical simulations [40].

A grid convergence study is performed by using mesh sizes $32 \times 32 \times 32$ and $64 \times 64 \times 32$ for the same domain. Both the L_2 as well as L_∞ measures of the error are presented (Table II) for velocity components on the mid-cavity plane. The error is calculated from the computed solution and the experimental data at each mesh point.

LES–IBM validation

Case 4: Film-cooling flow. The computational domain of interest is shown in Figure 8, and represents a spanwise periodic module containing a single coolant delivery tube inclined at 35° to the streamwise direction. For this geometry, detailed flow measurements have been provided by Lavrich and Chiapetta [42] with a coolant delivery tube length of $6D$ (where D is the diameter of the coolant jet delivery tube), and film cooling effectiveness data is provided by Sinha *et al.* [43] with a coolant delivery tube length of $1.75D$. The LES results of the present study are compared with these reported measurements. A uniform grid of $172 \times 102 \times 62$ is used to model the computational domain of size $17D \times 5D \times 6D$. The film-cooled surface is placed at $1.0D$ from the bottom of the computational domain with the length of the coolant delivery tube being equal to $1.75D$. The centre of the jet injection hole at the film-cooled surface is $5D$ downstream from the inlet plane. The axis origin is placed at the centre of hole on the film-cooled surface. Therefore, the domain is $[-5D, 12D] \times [-1D, 4D] \times [-3D, 3D]$. The jet delivery tube is simulated as an inclined cylindrical surface, and the IBM (with the Case A approach) is utilized to enforce the no-slip conditions on the delivery tube surface.

At the inlet of the coolant delivery tube, flow boundary conditions must be provided that are consistent with the measurements. In most experimental studies, detailed velocity measurements at the inlet to the coolant delivery tube or the jet-exit are not provided. Thus judicious choices must be made in this respect. It is generally accepted that since the crossflow influences the flow development in the delivery tube, the computational domain must include all or a portion of the coolant delivery tube. The present computations, with a $1.75D$ long delivery tube, are being compared with two sets of measurements with delivery tube lengths of $6D$ and $1.75D$. When comparing with the cooling effectiveness measurements of Sinha *et al.* [43] with a $1.75D$ long delivery tube and blowing ratio $M = 0.5$, the correct delivery tube length was used in the calculations, and a large stagnation-type plenum was assumed in the computations upstream of the $1.75D$ delivery tube, with the plenum fed by air-streams flowing parallel to the main crossflow. When comparing with the $6D$ delivery tube measurements of Lavrich and Chiappetta [42], instead of extending the delivery tube to be $6D$ long (due to the associated need for increasing the number of grid points significantly,

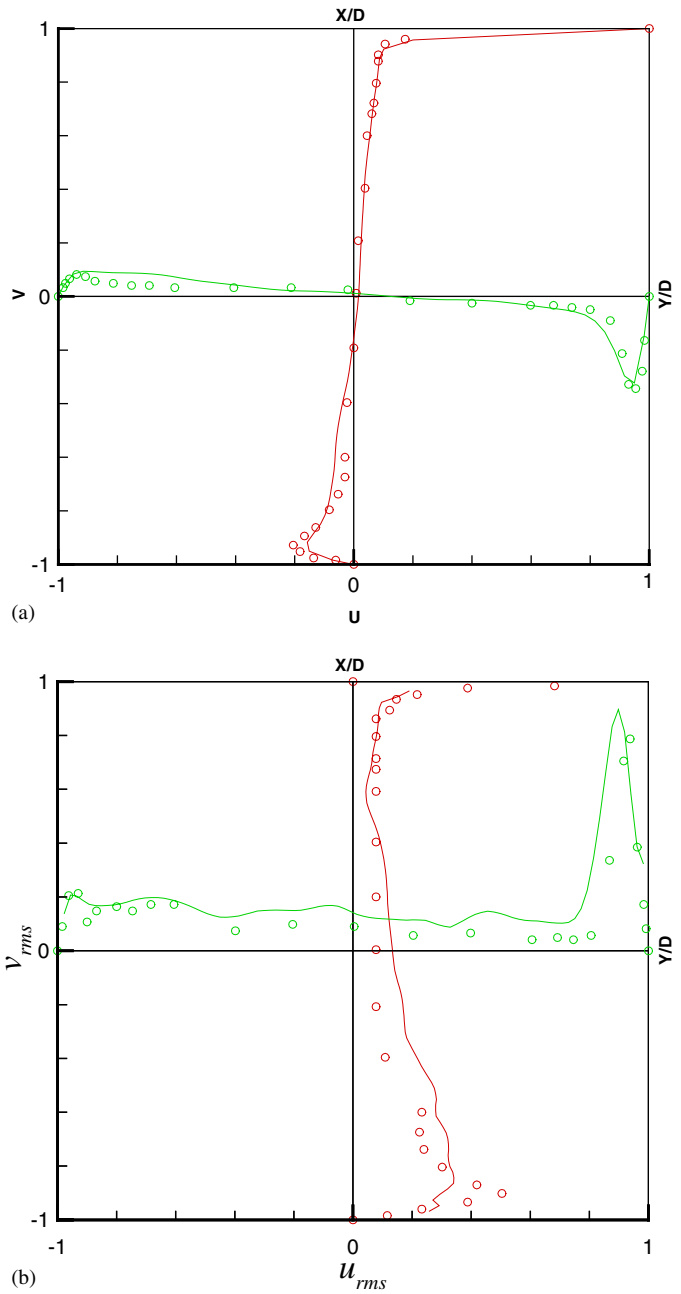


Figure 7. Comparison of LES results (lines) with experimental data (symbols) of Prasad and Koseff [41] on the centreplane of lid-driven cavity with spanwise aspect ratio of 0.5 at Reynolds number 10 000: (a) Mean velocity components; and (b) rms components of the fluctuating velocity field along X and Y centrelines.

Table II. Grid convergence test: Fine mesh is $64 \times 64 \times 32$ while coarse mesh is $32 \times 32 \times 32$.

	L_2	L_∞
U	2.24E-2 (6.23E-2)	6.82E-2 (2.51E-1)
V	3.34E-2 (6.44E-2)	1.09E-1 (1.64E-1)

Coarse mesh results are in parentheses.

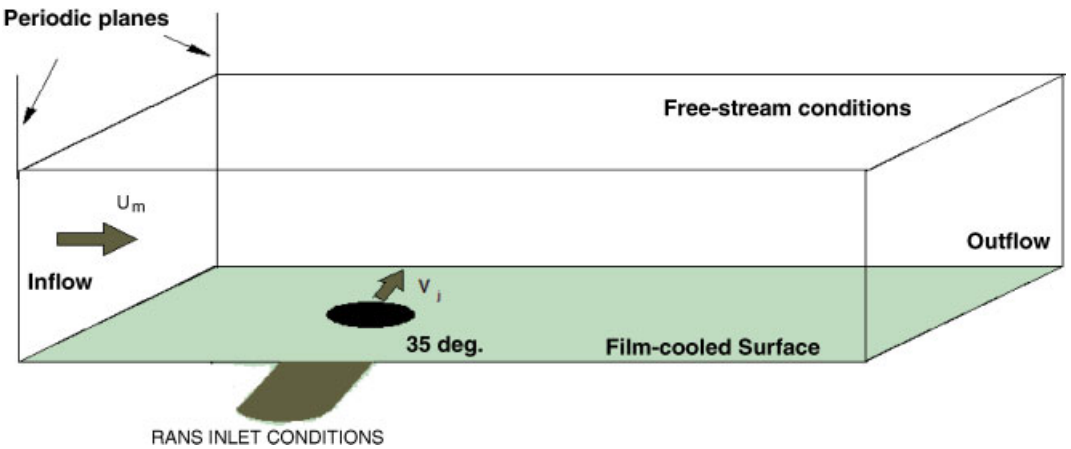


Figure 8. Schematic of the computational domain and boundary conditions.

and the limitations in the available computing resources), the delivery tube was maintained at $1.75D$, and mean velocities at the tube inlet were specified from a RANS calculation where a $6D$ long delivery tube fed by a large cylindrical plenum (as in the experiments) was used. Since LES requires the specification of instantaneous velocities, turbulent velocity fluctuations were added to the mean velocities. The velocity fluctuations were generated using a Gaussian random number generator (Box–Muller algorithm) with a variance corresponding to the RANS computed turbulent kinetic energy.

The top boundary of the computational domain (located $4D$ from the surface) is treated as freestream boundary. At the inlet, a fully developed turbulent profile (1/7th law, boundary layer thickness $\sim 1D$) is specified from the experimental data. At the outflow, a convective boundary condition is used where the convection speed is obtained from the mass flux balance. The spanwise direction (Z) is assumed to be periodic.

To validate the LES–IBM calculation procedure, the time-averaged LES results are compared with the velocity measurements of Lavrich and Chiappetta [42] and the film cooling effectiveness data of Sinha *et al.* [43]. Figures 9 and 10 present the velocity comparisons for blowing ratios (M) of 0.5 and 1.0, respectively. The streamwise (U) and vertical (V) component of velocity are shown along the spanwise centreplane ($Z/D = 0$) at three axial locations ($X/D = 0$, 5 and 1.0), while the spanwise (W) component of velocity is presented at $Z/D = 0.5$ which corresponds to the spanwise edge of the hole. The time-averaged statistics are obtained

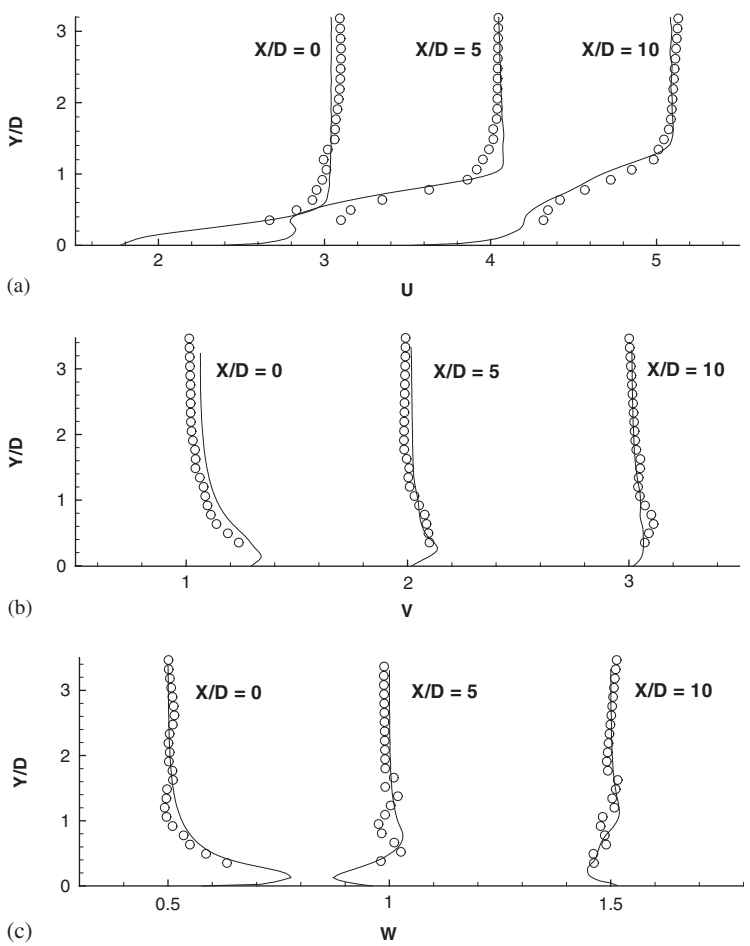


Figure 9. Comparison of predicted (lines) and measured (symbols) [42] velocities at a blowing ratio $M = 0.5$: (a) Streamwise component of velocity at $Z/D = 0$; (b) vertical component of velocity at $Z/D = 0$; and (c) spanwise component of velocity at $Z/D = 0.5$.

as the run-time average from the computations over approximately 10 flow-through time periods (flow-through time is the time taken by the crossflow to sweep the computational domain from the inlet plane to the exit plane). The velocity predictions at both blowing ratios are generally in good agreement with the experimental data. At $X/D = 5$, the streamwise velocity does underpredict the measured data particularly for $M = 1.0$. However the spanwise and vertical velocity components are in excellent agreement with the data. These velocity components play a critical role in the spanwise and vertical penetration of the jet, respectively, and the good agreement with the data at $M = 0.5$ and 1.0 provides a measure of validation for the LES calculations.

Predictions of the centreline film cooling effectiveness at a blowing ratio of 0.5 are shown in Figure 11. Also shown are the data of Sinha *et al.* [43] at the same blowing ratio.

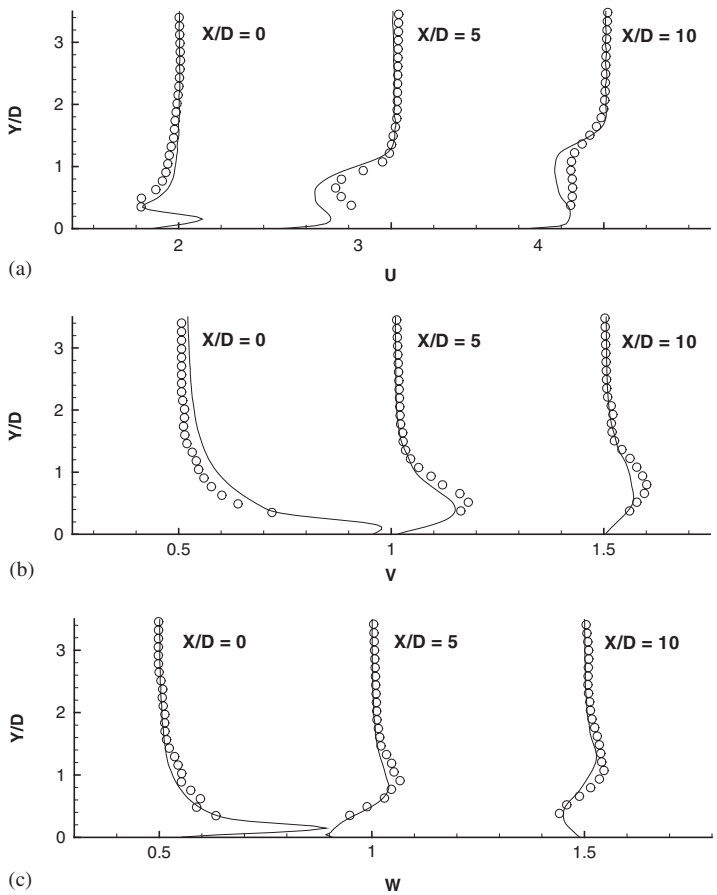


Figure 10. Comparison of predicted (lines) and measured (symbols) [42] velocities at a blowing ratio $M = 1.0$: (a) Streamwise component of velocity at $Z/D = 0$; (b) vertical component of velocity at $Z/D = 0$; and (c) spanwise component of velocity at $Z/D = 0.5$.

In both the experiments and the computations the coolant delivery tube length was the same and equal to $1.75D$. The agreement between the predictions and the data is excellent with the predicted profile lying virtually on top of the data. The good agreement between the cooling effectiveness predictions and the data again confirms the predictive accuracy of LES–IBM.

LES–IBM for complex geometries

To illustrate the full potential of LES–IBM, turbulent flow calculations are presented for (i) a trapped vortex combustor configuration with a cylindrical geometry and several cylindrical fuel–air-injection ports, and (ii) flow past a single row of stator–rotor blades with the rotor blades moving at a prescribed velocity relative to the stator row.

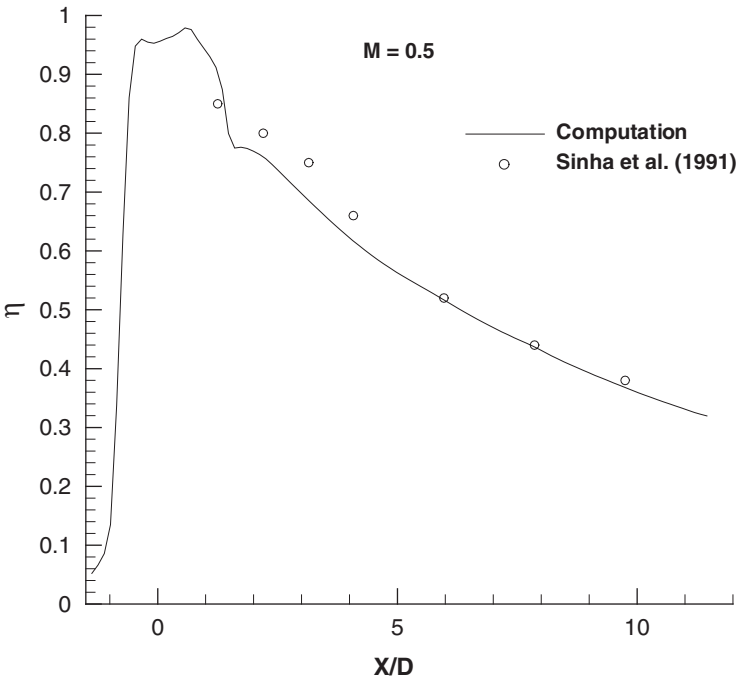


Figure 11. Comparison of predicted centreline film cooling effectiveness (lines) with experimental data (symbols) of Sinha *et al.* [43] at $M = 0.5$.

Case 5: Trapped vortex combustor (TVC). The TVC is a unique turbine engine combustor [44–49] concept with improved flame stabilization and emissions performance. In a trapped vortex (TV) combustor, a properly sized cavity is used to trap a vortex, which is used to provide a stable pilot for a wide range of operating conditions. A simple schematic presented in Figure 8 shows the concept of the TVC investigated by Mancilla [45], and consists of an asymmetric dumbbell-shaped flame holder, with fuel and secondary (coaxial) air supplied into the cavity from the afterbody. Thus the geometrical challenge is to represent the cylindrical combustor, the annular array of four fuel-jets supplied with co-axial air, and the array of four inner and eight outer air-jets as shown in Figure 12. It is evident that representing such a geometry with the large array of fuel and air-injectors is a lot simpler in Cartesian coordinates, and using the IBM to ensure that the boundary conditions are satisfied on the curved solid surfaces.

In a recent study, Stone and Menon [49] used 2-D LES to simulate the fuel–air mixing and combustion in a TVC. It should be noted that in two-dimensional simulations, the energy cascade via vortex stretching and folding or tilting mechanisms are absent. This leads to large energy retaining eddies. However, in real situations, these vortices supply the energy to smaller scales, which dissipate the energy away due to viscosity. The present 3-D LES simulations correctly represent this energy exchange, and details of the flow physics are provided in Reference [1].

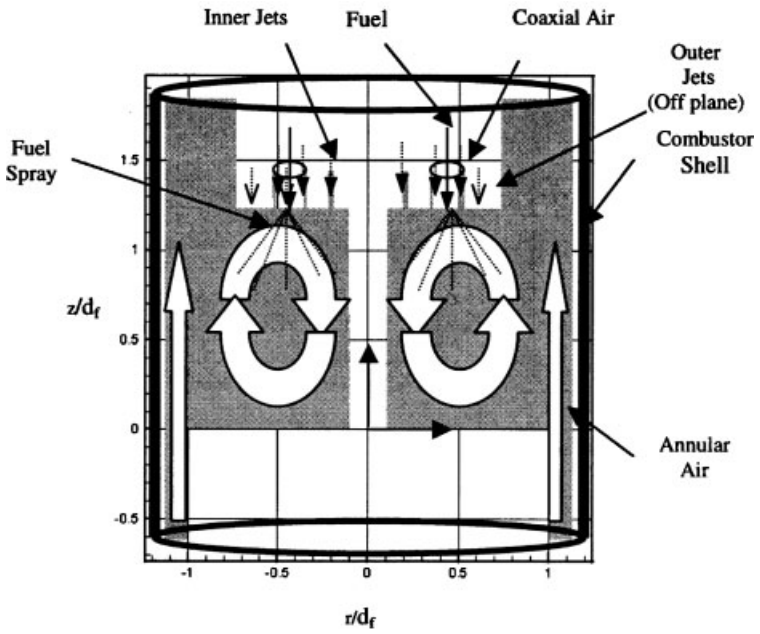
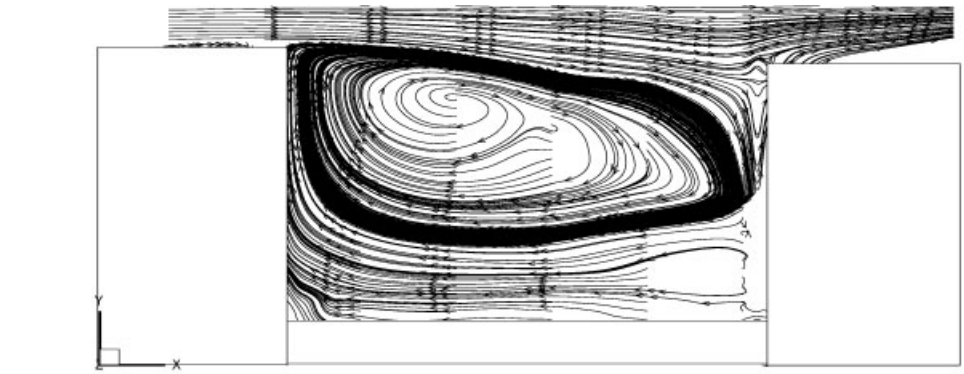


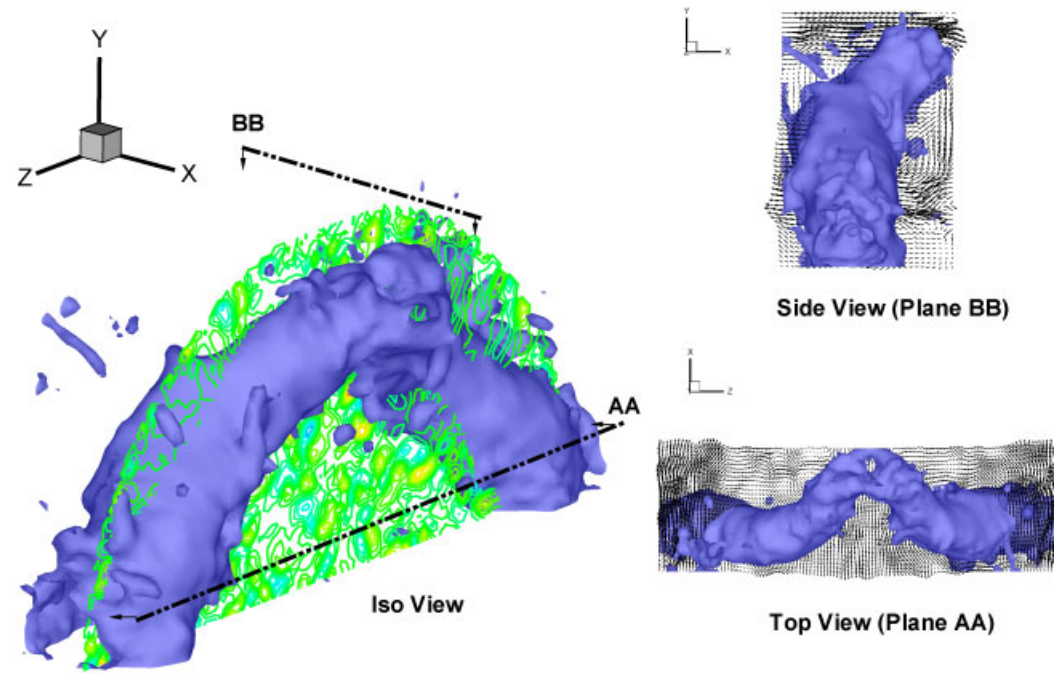
Figure 12. Schematic and concept of the TVC (from Reference [45]).

A uniform Cartesian grid of $92 \times 57 \times 117$ points is used for the computational domain. All the dimensions are selected to approximate the experimental set-up of Mancilla [45] (Figure 12). The ratio of air injection velocity to the mainflow velocity is 2.2. The Reynolds number based on the annular mainflow velocity and air hole dimension (D) is 3400 for these simulations. The radii of the forebody, the connecting tube, the afterbody and the outer shell are $24.5D$, $3.7D$, $23D$ and $27.5D$, respectively. The lengths of the forebody, the connecting tube and the afterbody are $12D$, $30D$ and $12D$, respectively. At the inflow, a fully developed velocity profile along with Gaussian velocity fluctuations is prescribed. At the walls, no slip boundary conditions are imposed using the IBM (Case A interpolation). Uniform injection of air and fuel are specified at the respective injection locations on the afterbody. At the outflow, a convective scheme is applied to convect away the flow structures out of the computational domain without any spurious reflections. The wave speed is calculated to maintain the mass flux balance in the whole domain. However, reaction and passive scalar mixing issues are not addressed in these simulations and hence, the fuel is treated with the same material properties as that of air.

The three-dimensional LES–IBM simulations show that the vorticity magnitude of the TV in the cavity changes due to vortex stretching mechanism that is absent in the two-dimensional simulations done by other researchers [46, 47, 49]. The time-averaged streamtraces are presented at the meridional plane $\theta = 90^\circ$ (Figure 13(a)), where a large recirculation region is formed between the annular mainflow and the fuel injection locations. The ingestion of annular mainflow in front of the afterbody is the main mechanism of the flow entrainment inside the cavity (Figure 13(a)). The instantaneous location of the TV is visualized by a low-pressure



(a)



(b)

Figure 13. (a) Streamtraces at $\theta = 90°$; and (b) iso-surface of pressure revealing doughnut shape of trapped vortex.

iso-surface (Figure 13(b)). The TV is a doughnut-shaped structure inside the cavity. The azimuthal motion of the streamtraces observed along the surface of the TV is absent in all the previous 2-D studies. However, the core of the TV is mostly irrotational. The top and side view of TV show that the core of the TV is closer to the afterbody near $\theta = 90°$. The velocity vectors in these projected views clearly show that mixing is achieved along the edges of this TV. Most of the vorticity is confined along the edges of fuel/air injections from the afterbody.

Case 6. Unsteady stator–rotor interactions. To illustrate the potential of LES–IBM on a moving complex geometry, an unsteady stator–rotor interaction problem is solved. Inherent unsteadiness of such a flow field is created by the relative motion between the stationary blades (stator) and the rotating blades (rotor), and requires the designer to account for three-dimensional as well as unsteady effects. The unsteadiness is caused by (a) the interaction of the rotor airfoils with the wakes and passage vortices generated by upstream airfoils, (b) the relative motion of the rotors with respect to the stators (potential effect), and (c) the shedding of vortices by the airfoils because of the blunt trailing edges [50, 51]. Computation of such flows is complicated by relative motion between rotor and stator airfoils and the periodic transition of the flow from laminar to turbulent. Unsteady simulations have been performed using different approximations such as the 'mixing-plane' approach, the 'average passage' approach and unsteady RANS [52–54]. In the 'mixing-plane' approach, flow through each airfoil row in the machine is calculated for a specified circumferentially uniform inlet and average exit boundary conditions. The effect of periodic unsteadiness is not accounted for in this approach. In the 'average passage' approach, the effects of adjacent airfoil rows are accounted for through the use of body forces and 'apparent stresses'. Reliable models are not yet available to account for circumferential variation of 'apparent stresses' [55]. Unsteady RANS has some potential to resolve periodic unsteadiness and can yield significantly better results. However, the modelling of the energy spectrum is generally inaccurate, particularly if there is significant turbulent energy in the secondary flows through the airfoil blade row. A phase-lagging method [56] is generally used to model blade rows with unequal airfoil counts. With this procedure, the solution domain for a given row need only span one pitch rather than multiple pitches as is required for spatial periodicity. These calculations have been performed invariably using 'sliding mesh' techniques [57] requiring further constraints on matching the interface conditions on different fluxes (all of these are not usually satisfied).

In the present study, we utilize LES with moving IBM to simulate unsteady stator–rotor interactions. Although the calculation is performed for an incompressible fluid at a low Reynolds number, it demonstrates the strength of the method by avoiding all *ad hoc* assumptions pertaining to RANS modelling and the computational complexities of sliding meshes.

The geometry of the airfoils is taken from the numerical study of Kelecy *et al.* [57]. The airfoil profile is approximated by cubic spline surfaces. The airfoil is divided into a leading edge, a trailing edge, a pressure surface and a suction surface to ensure that immersed boundary conditions are enforced on enough grid points to accurately realize the geometry. A uniform Cartesian grid of $302 \times 202 \times 11$ points is used for a domain of the size $3D \times 1D \times 0.1D$, where D is the chord length of the rotor airfoil. The choice of a small spanwise dimension ($0.1D$) may not allow larger physical scales in the flow, and hence may not be desirable. A uniform flow field is specified at the inlet. The Reynolds number based on the inflow velocity

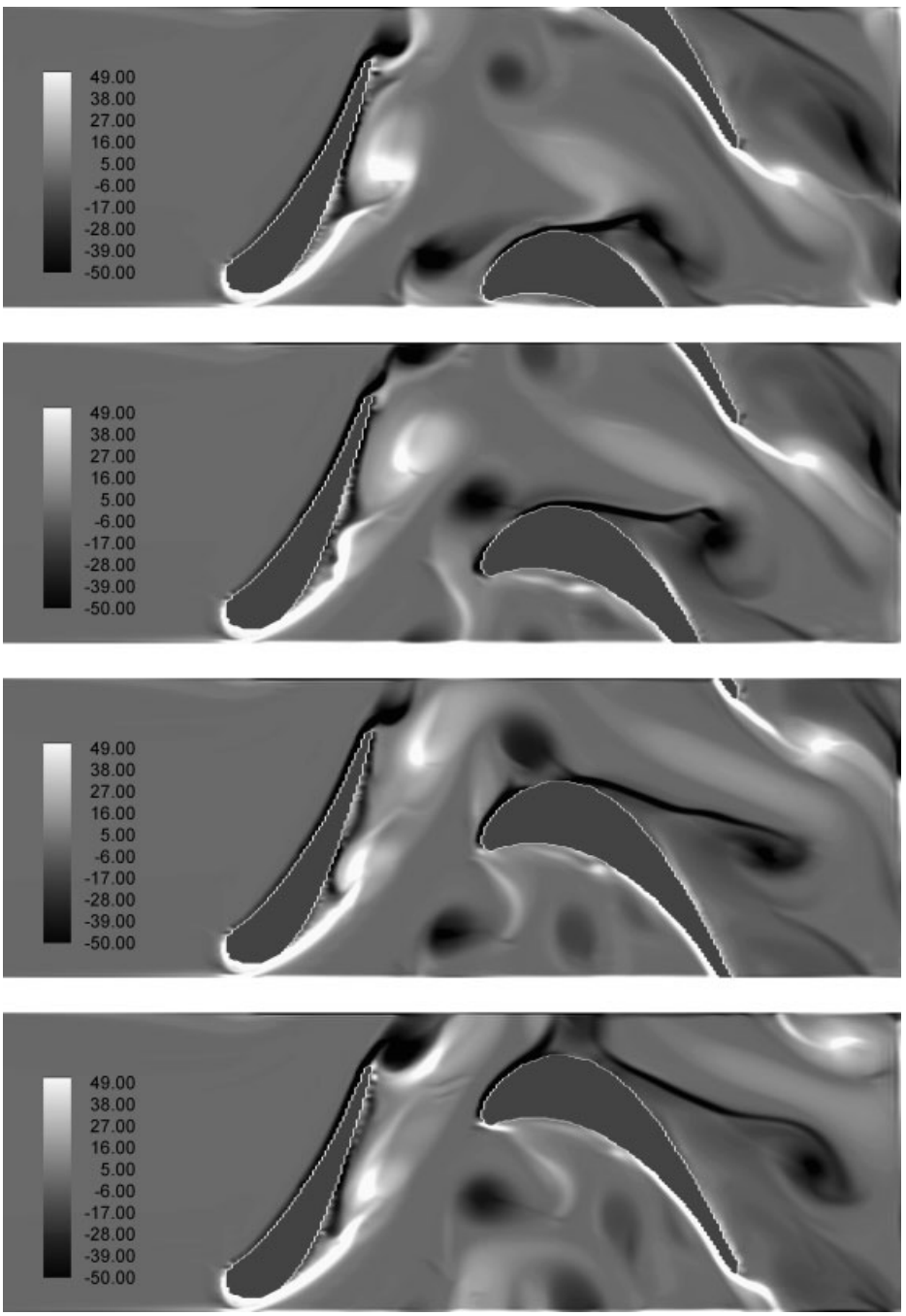


Figure 14. Four frames from the rotor cycle behind the stator. Contours are instantaneous vorticity field (component normal to the plane of flow field).

and rotor chord length is 5000. Periodic boundary conditions are applied in the pitch direction (y) and the spanwise (z) direction. At the outflow, a non-reflective convective scheme is applied to convect away the flow structures out of the computational domain without any spurious reflections. The wave speed is calculated to maintain the mass flux balance in the whole domain. It must be kept in mind that this numerical simulation is performed to demonstrate the capability of LES–IBM for a very complex problem and parameters chosen for this study may not be representative of a true physical problem.

Two-sided forcing on both sides of the immersed surface (Case B) is used here. The use of one-sided forcing (Case A) led to ill-conditioned weights in the interpolation stencils for certain locations of the immersed surfaces, and correct solutions could not be realized. In such problems, with moving boundaries, it is therefore important to adopt the two-sided forcing scheme advocated in this paper. The specific details of computational stencils and interpolation weights are shown for the suction and pressure immersed surfaces in Appendix A.

The snapshots of instantaneous vorticity field (component normal to the plane of flow field) are shown in Figure 14. The development of boundary layer vorticity on the solid surfaces and its subsequent shedding into the main crossflow near the trailing edge of the stator blade produces a mixing layer-type wake. Evidence of flow separation can be seen on the suction sides of the stator and rotor, and is a reflection of the low Reynolds number (off-design condition) used in the present calculation. Note that such a recirculation produces boundary layer vorticity with opposite sense and is captured in the simulations here. The trailing edge vortices of the stator blade impact on the suction side of the rotor blade near its leading edge. The trailing edge vortices of the rotor and the vortices formed due to the interaction of stator wake and suction-side boundary layer are shed into the passage flow and convected out of the domain. Two such vortices can be seen in these frames moving through the rotor blade passage.

CONCLUSION

A large eddy simulation (LES) methodology for turbulent flows in complex geometries with stationary and moving boundaries is presented in this paper. It advocates the use of high-order accurate schemes with robust subgrid scale (SGS) stress model in the LES calculations, and the immersed boundary method (IBM) for resolving the geometrically complex stationary or moving boundaries. A two-sided forcing strategy is proposed for the IBM, and advantages of this approach over the traditional one-sided approach is presented. The IBM implementation is validated for flow past a heated cylinder and demonstrated for flow past an array of 95 cylinders arranged in a staggered configuration. The LES procedure is first validated for the lid-driven cavity problem. Next, the combined LES–IBM procedure is validated for a turbulent film-cooling flow configuration, with inclined cylindrical cooling holes, and predictions are shown to be in good agreement with measurements. The merit and simplicity of the LES–IBM approach for simulating turbulent flows in complex geometries on a Cartesian mesh are further demonstrated for trapped vortex combustor geometry with complex geometrical features, and unsteady stator–rotor interactions with moving rotor blades.

APPENDIX A

Suction side

Let Φ be the indicator function to locate the immersed boundary between the grid lines. The shaded region defines the actual blade. Since it is separated from the actual fluid region by an immersed boundary, the interior (shaded) domain is referred to as virtual solid.

$$\begin{aligned} \Phi &= (y_j - y_s) \times (y_s - y_{j-1}) \\ \delta_1 &= \frac{(y_j - y_s)}{(y_{j+1} - y_s)}, \quad \delta_2 = \frac{(y_s - y_{j-1})}{(y_{j+1} - y_s)} \\ V_j &= (1 - \delta_1)V_s + \delta_1 V_{j+1} \\ V_{j-1} &= (1 + \delta_2)V_s - \delta_2 V_{j+1} \end{aligned} \tag{A1}$$

In Equation (A1), Φ is positive if and only if the immersed boundary (y_s) is between the grid lines y_j and y_{j-1} (Figure A1). Using linear interpolation between y_{j+1} and y_s and expressing the velocity at the forced locations (y_j and y_{j-1}) in terms of solved velocity (V_{j+1}) and solid wall velocity (V_s), we obtain the last expression in Equation (A1).

Pressure side

In Equation (A2), Φ is positive if and only if the immersed boundary (y_p) is between the grid lines y_j and y_{j+1} (Figure A2). Using linear interpolation between y_{j-1} and y_p and expressing the velocity at the forced locations (y_j and y_{j+1}) in terms of solved velocity (V_{j-1}) and solid

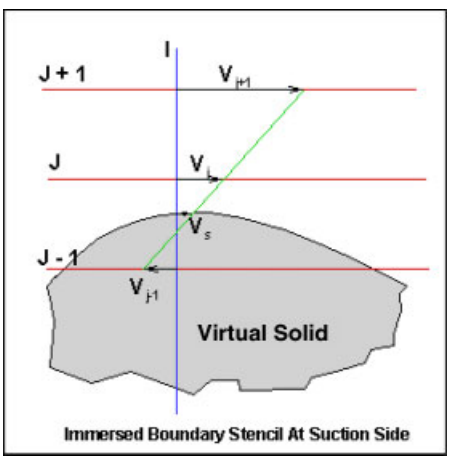


Figure A1. Computational stencils near the immersed boundary (suction-side surface) of moving blade.

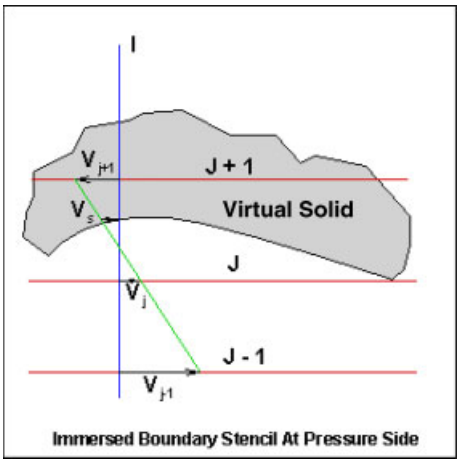


Figure A2. Computational stencils near the immersed boundary (pressure-side surface) of moving blade.

wall velocity (V_p), we obtain the last expression in Equation (A2).

$$\begin{aligned} \Phi &= (y_{j+1} - y_p) \times (y_p - y_j) \\ \delta_1 &= \frac{(y_p - y_j)}{(y_p - y_{j-1})}, \quad \delta_2 = \frac{(y_{j+1} - y_p)}{(y_p - y_{j-1})} \\ V_j &= (1 - \delta_1) V_p + \delta_1 V_{j-1} \\ V_{j+1} &= (1 + \delta_2) V_p - \delta_2 V_{j-1} \end{aligned} \tag{A2}$$

Again, it is noted that in moving geometries, two-sided forcing (Case B) should be preferred over one-sided forcing (Case A) because the linear interpolation weights can become large. Also, the problem of completely immersed cells changing into flow region of interest or *vice versa* is handled without any problem. Fadlun *et al.* [6] considered only single-sided forcing to describe the motion of immersed solid objects. Care must be taken in moving different sections of a single geometry.

NOMENCLATURE

U_i	filtered velocity field
p	pressure field divided by constant density
τ_{ij}	subgrid scale (SGS) stress tensor
f_i	body force terms arising due to immersed boundary
Re	Reynolds number
g_{ijk}	RHS of discrete pressure Poisson equation
$G_{ij}(k)$	Fourier transform of g_{ijk}

X	discrete operator for second derivative in X direction
Y	discrete operator for second derivative in Y direction
D_x	eigenvalues of X
D_y	eigenvalues of Y
P_x	eigenvectors of X
P_y	eigenvectors of Y
x	streamwise direction
y	vertical direction
z	spanwise direction (periodic)
Δ	mesh spacing
δ	distance of forcing point from the immersed surface
$\mathbf{x}_\Gamma$	location of immersed boundary

ACKNOWLEDGEMENTS

This work was supported by a grant from the AFOSR-Directorate of Materials and Aerospace Sciences with Dr Julian Tishkoff serving as the contract monitor, and the Louisiana Board of Regents under the Louisiana Educational Quality Support Fund (LEQSF) with Mr Jim Gershey serving as the program manager. Their support is gratefully acknowledged. Insightful comments by the reviewers are also acknowledged.

REFERENCES

1. Tyagi M. Large eddy simulations of complex turbulent flows. *Ph.D. Dissertation*, Louisiana State University, 2003.
2. Peskin CS. Numerical analysis of blood flow in the heart. *Journal of Computational Physics* 1977; **25**: 220–252.
3. Glowinski R, Pan T-W, Periaux J. A ficitious domain method for unsteady incompressible viscous flow modeled by Navier–Stokes equations. *Contemporary Mathematics* 1994; **157**:421–431.
4. Stockie JM. Analysis and computation of immersed boundaries, with application to pulp fibres. *Ph.D. Dissertation*, University of British Columbia, 1997.
5. Yusof JM. Interaction of massive particles with turbulence. *Ph.D. Dissertation*, Cornell University, 1996.
6. Fadlun E, Verzicco R, Orlandi P, Yusof, JM. Combined immersed boundary finite difference methods for complex flow simulations. *Journal of Computational Physics* 2000; **161**:35–60.
7. Kellogg S. Immersed boundary methods with applications to flow control. *M.S. Thesis*, Rice University, TX, 2000.
8. Kim J, Kim D, Choi H. An immersed-boundary finite-volume method for simulations of flow in complex geometries. *Journal of Computational Physics* 2001; **171**:132–150.
9. Balaras E. Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations. *Computers and Fluids* 2004; **33**:375–404.
10. Grigoriadis DGE, Bartzis JG, Goulas A. Efficient treatment of complex geometries for large eddy simulations of turbulent flows. *Computers and Fluids* 2004; **33**:201–222.
11. Iaccarino G, Verzicco R. Immersed boundary technique for turbulent flow simulations. *Applied Mechanics Reviews* 2003; **56**:331–347.
12. Tseng Y-H, Ferziger JH. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics* 2003; **192**:593–623.
13. Chorin AJ. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics* 1967; **2**:12–26.
14. Tafti D. Alternate formulations for the pressure equation Laplacian on a collocated grid for solving the unsteady incompressible Navier–Stokes equations. *Journal of Computational Physics* 1995; **116**:143–153.
15. Rai MM, Moin P. Direct simulations of turbulent flow using finite-difference schemes. *Journal of Computational Physics* 1991; **96**:15–53.
16. Ghosal S. An analysis of numerical errors in large-eddy simulations of turbulence. *Journal of Computational Physics* 1996; **125**:187–206.

17. Strikwerda JC. *Finite Difference Schemes and Partial Differential Equations*. Wadsworth & Brooks/Cole: Belmont, CA, 1989.
18. Fornberg B. Generation of finite difference formulas on arbitrarily spaced grids. *Mathematics of Computation* 1988; **51**:699–706.
19. Pointel G. Direct numerical simulations of turbulent channel flow on massively parallel computers. *M.S. Thesis*, Louisiana State University, 1995.
20. Vreman B, Geurts BJ, Kuerten H. On the formulation of the dynamic mixed subgrid scale model. *Physics of Fluids* 1994; **6**:4057–4059.
21. Zang Y, Street RL, Koseff JR. A dynamic mixed subgrid scale model and its application to turbulent recirculating flows. *Physics of Fluids* 1993; **5**:3186–3196.
22. Ghosal S, Lund TS, Moin P, Akselvoll K. A dynamic localization model for large eddy simulation of turbulent flows. *Journal of Fluid Mechanics* 1995; **286**:229–255.
23. Pomraning E, Rutland CJ. A dynamic one-equation non-viscosity LES model. *AIAA Journal* 2002; **40**:689–701.
24. Piomelli U, Liu J. Large eddy simulation of rotating channel flows using a localized dynamic model. *Physics of Fluids* 1995; **7**:839–848.
25. Meneveau C, Lund TS, Cabot WH. A Lagrangian dynamic subgrid scale model of turbulence. *Journal of Fluid Mechanics* 1996; **319**:353–385.
26. Peskin CS. The immersed boundary method. *Acta Numerica* 2002; 479–517.
27. Beyer RP, Leveque RJ. Analysis of a one-dimensional model for the immersed boundary method. *SIAM Journal on Numerical Analysis* 1992; **29**:332–364.
28. Lee L, Leveque RJ. An immersed interface method for incompressible Navier–Stokes equations. *SIAM Journal on Scientific Computing* 2003; **25**:832–856.
29. Lima E, Silva ALF, Silveira-Neto A, Damasceno JJR. Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method. *Journal of Computational Physics* 2003; **189**:351–370.
30. Cortez R, Minion M. The blob projection method for immersed boundary problems. *Journal of Computational Physics* 2000; **161**:428–453.
31. Cottet G-H, Poncet P. Particle methods for direct numerical simulations of three-dimensional wakes. *Journal of Turbulence* 2002; **3**(38):1–9.
32. Kirkpatrick MP, Armfield SW, Kent JH. A representation of curved boundaries for the solution of the Navier–Stokes equations on a staggered three-dimensional Cartesian grid. *Journal of Computational Physics* 2003; **184**:1–36.
33. Vikhansky A. A new modification of the immersed boundaries method for fluid-solid flows: moderate Reynolds numbers. *Journal of Computational Physics* 2003; **191**:328–339.
34. Russell D, Wang ZJ. A Cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow. *Journal of Computational Physics* 2003; **191**:177–205.
35. White FM. *Fluid Mechanics*. Prentice-Hall: New York, 1990.
36. Incropera FP, DeWitt DP. *Fundamentals of Heat and Mass Transfer*. Wiley: New York, 1990.
37. Ye T, Shyy W, Chung JN. A fixed-grid, sharp-interface method for bubble dynamics and phase change. *Journal of Computational Physics* 2001; **174**:781–815.
38. Deshpande MD, Milton SG. Kolmogorov scales in a driven cavity flow. *Fluid Dynamics Research* 1998; **22**: 359–381.
39. Leriche E, Gavrilakis S. Direct numerical simulation of the flow in a lid-driven cubical cavity. *Physics of Fluids* 2000; **12**:1363–1376.
40. Zang Y. On the development of tools for the simulation of geophysical flows. *Ph.D. Dissertation*, Stanford University, 1993.
41. Prasad AK, Koseff JR. Reynolds number and end-wall effects on a lid-driven cavity flow. *Physics of Fluids* 1989; **1**:208–218.
42. Lavrich PL, Chiappetta LM. An investigation of jet in a cross flow for turbine film cooling applications. *UTRC Report No. 90-04*, United Technologies Research Center, CT, 1990.
43. Sinha AK, Bogard DG, Crawford ME. Film-cooling effectiveness downstream of a single row of holes with variable density ratio. *Journal of Turbomachinery-Transactions of the ASME* 1991; **113**:442–449.
44. Hsu K-Y, Goss LP, Trump DD, Roquemore WM. Performance of a trapped-vortex combustor. *AIAA 95-0810*, 1995.
45. Mancilla PC. Flame stability in a trapped-vortex spray-combustor. *M.S. Thesis*, Louisiana State University, 2001.
46. Katta VR, Roquemore WM. Numerical studies on trapped-vortex combustor. *AIAA 96-2660*, 1996.
47. Katta VR, Roquemore WM. Study on trapped-vortex combustor—effect of injection on flow dynamics. *Journal of Propulsion and Power* 1998; **14**(3):273–281.
48. Hosokawa S, Ikeda Y, Nakajima T. Effect of flame holder shape on vortex shedding. *AIAA 96-3130*, 1996.
49. Stone C, Menon S. Simulation of fuel–air mixing and combustion in a trapped-vortex combustor. *AIAA Paper No. 00-0478*, *38th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, 2000.

50. Rai MM, Madavan NK. Multi-airfoil Navier–Stokes simulations of turbine rotor–stator interaction. *Journal of Turbomachinery-Transactions of the ASME* 1990; **112**:377–384.
51. Saxer AP, Giles MB. Predictions of three-dimensional steady and unsteady inviscid transonic stator/rotor interaction with inlet radial temperature nonuniformity. *Journal of Turbomachinery-Transactions of the ASME* 1994; **116**:347–357.
52. Denton J. Calculation of three-dimensional viscous flows through multistage turbine. *ASME Paper 90-GT-19*, 1990.
53. Adamczyk JJ. Model equation for simulating flows in multistage turbomachinery. *ASME Paper 85-GT-226*, 1985.
54. Sharma OP, Ni RH, Tanrikut S. Unsteady flows in turbines—impact on design procedure. *AGARD-LS-195*, *Paper # 5*, 1994.
55. Lakshminarayana B. *Fluid Dynamics and Heat Transfer of Turbomachinery*. Wiley: New York, 1995.
56. Erdos JL, Alzner E, McNally W. Numerical solution of periodic transonic flow through a fan stage. *AIAA Journal* 1977; **15**:1559–1568.
57. Kelecy FJ, Griffin JW, Delaney RA. The effect of vane-blade spacing on transonic turbine stage performance. *AGARD-CP-571*, *Paper # 5*, 1995.

