Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 224 (2007) 757–784
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

An immersed boundary method for complex incompressible flows

Jung-Il Choi [a], Roshan C. Oberoi [a], Jack R. Edwards [a,*], Jacky A. Rosati [b]

[a] Department of Mechanical and Aerospace Engineering, North Carolina State University, Raleigh, NC 27695-7910, USA
[b] US Environmental Protection Agency, Office of Research and Development, E343-06, 109 T.W. Alexander Drive, Research Triangle Park, NC 27711, USA

Received 9 February 2006; received in revised form 25 August 2006; accepted 18 October 2006
Available online 12 December 2006

Abstract

An immersed boundary method for time-dependent, three-dimensional, incompressible flows is presented in this paper. The incompressible Navier–Stokes equations are discretized using a low-diffusion flux splitting method for the inviscid fluxes and second-order central-differences for the viscous components. Higher-order accuracy achieved by using weighted essentially non-oscillatory (WENO) or total variation diminishing (TVD) schemes. An implicit method based on artificial compressibility and dual-time stepping is used for time advancement. The immersed boundary surfaces are defined as clouds of points, which may be structured or unstructured. Immersed-boundary objects are rendered as level sets in the computational domain, and concepts from computational geometry are used to classify points as being outside, near, or inside the immersed boundary. The velocity field near an immersed surface is determined from separate interpolations of the components tangent and normal to the surface. The tangential velocity near the surface is constructed as a power-law function of the local wall normal distance. Appropriate choices of the power law enable the method to approximate the energizing effects of a turbulent boundary layer for higher Reynolds number flows. Five different flow problems (flow over a circular cylinder, an in-line oscillating cylinder, a NACA0012 airfoil, a sphere, and a stationary mannequin) are simulated using the present immersed boundary method, and the predictions show good agreement with previous computational and experimental results. Finally, the flow induced by realistic human walking motion is simulated as an example of a problem involving multiple moving immersed objects.

Keywords: Immersed boundary method; Incompressible Navier–Stokes equations; High Reynolds number flows

1. Introduction

The prediction of contaminant dispersal in workplace environments involves consideration of bulk transport effects, such as those induced by heating, ventilation and air conditioning (HVAC) activity, as well as

* Corresponding author. Tel.: +1 919 515 5264; fax: +1 919 515 7968.
E-mail addresses: jchoi2@unity.ncsu.edu (J.-I. Choi), rcoberoi@unity.ncsu.edu (R.C. Oberoi), jredward@unity.ncsu.edu (J.R. Edwards), rosati.jacky@epa.gov (J.A. Rosati).

0021-9991/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2006.10.032

local transport effects, such as those induced by human activity and opening/closing doors. Reaerosolization of deposited particulates may result from human activity, and entrainment of reaerosolized material into wakes induced by motion events can also serve as a transport mechanism. The prediction of such effects requires obtaining time-dependent solutions for the hydrodynamic fields (velocity, pressure) induced by the motion of immersed objects (such as a walking person). While it is possible to simulate complex motion of such objects by using boundary-conforming, adaptive remeshing of (usually) unstructured grids, the development of such techniques is quite complicated. Recently, non-boundary conforming grid methods are attracting attention in simulations of such problems due to their ability to handle motion of multiple objects without significant coding complexity.

Non-boundary conforming methods can be divided into two major classes based on the specific treatment of the boundary cells [1]; (1) *Cartesian cut-cell* approaches based on the construction of irregular grid cells near surfaces, and (2) *Immersed boundary* methods, which enforce wall conditions indirectly through the use of forcing functions. The *Cartesian cut-cell* approach has been used in Euler flows [2,3] and has been extended to simulate viscous flows including moving boundaries [4–6]. In this approach, the Cartesian grid cells that intersected by the immersed body are reconstructed so that local boundary conformity is achieved. The governing equations in these cells can be discretized using finite volume methods, but special interpolation procedures may be needed in order to preserve overall second-order accuracy. This approach allows boundary conditions on body to be imposed in a manner similar to that used for body-fitted grids. However, an iterative solution procedure is required due to the irregular stencil near the immersed boundaries, and special remedies might be needed for a cut-cell formation for complex 3D geometries.

The *immersed boundary* method was first introduced by Peskin [7] in order to study blood flow in the human heart. The boundary was modeled as a set of elements linked by springs. Goldstein et al. [8] introduced a *feedback forcing* approach to enforce the desired boundary conditions at the immersed boundary. This feedback forcing is successful for low Reynolds number flows, but the computational time step may be restricted. Recently, Mohd-Yusof [9] and Fadlun et al. [10] recast the immersed boundary method based on a *direct forcing* approach. A key to this method is the enforcement of boundary conditions indirectly, through specification of local distributions of the fluid properties (velocity and sometimes pressure) near the immersed boundary. Fadlun et al. [10] utilized linear interpolations for the velocity at the first grid point external to the immersed body. The choice of the interpolation direction can be ambiguous for complex geometries. Kim et al. [11] utilized second-order linear or bi-linear interpolation schemes within a finite volume context and also introduced mass source/sink forcing in the continuity equation. Gilmanov et al. [12] developed a general velocity interpolation scheme along the direction normal to an immersed body, which itself was constructed as an unstructured, triangular mesh. The same interpolation idea was successfully applied to large-eddy simulation of turbulent flow in a wavy channel [13]. They also extended their immersed boundary methods to moving-body problems at low and moderate Reynolds numbers using a quadratic interpolation near the body and a hybrid staggered/non-staggered grid arrangement [14].

The objective of the present study is to develop a more general immersed boundary method that is valid at all Reynolds numbers and is suitable for implementation on arbitrary grid topologies. The basic approach is similar to that of Gilmanov, Balaras, and Sotiropoulos [12–14]. Immersed-boundary objects are generated as clouds of points, which may be structured or unstructured, and are rendered as level sets in the computational domain. This facilitates the decomposition of the flow field into cells within, near, and outside the body using concepts from computational geometry. We introduce the concept of *tangency correction* by decomposing the velocity into tangential and normal components along the outward normal direction to the immersed surface. The tangential velocity near the surface is constructed as a power-law function of the wall normal distance. Suitable choices of the power-law enforce more flow tangency to the surface, thus mimicking the energizing effect of a turbulent boundary layer. General interpolation techniques are used to connect the functional forms for the fluid properties near the surface to flow-property and gradient information away from the surface. Five different problems (flow over a circular cylinder, an in-line oscillating cylinder, a NACA0012 airfoil, a sphere, and a stationary mannequin) are solved in order to verify the feasibility of the present method. Finally, the present method is applied to simulate the flow induced by realistic human walking motion as an example of a problem involving multiple moving immersed objects.

2. Numerical method

2.1. Governing equations

The governing equations for a three-dimensional, unsteady, incompressible flow can be written as

$$\frac{\partial u_i}{\partial x_i} = 0, \tag{1}$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_i u_j + p\delta_{ij} - \tau_{ij}) = \rho f_i, \tag{2}$$

where u_i is the velocity vector, ρ is the density, p is the pressure, τ_{ij} is the viscous stress tensor, and f_i is an external force (set to zero for the remaining developments). The elements of the stress tensor τ_{ij} for a Newtonian fluid can be defined as

$$\tau_{ij} = \mu\left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right), \tag{3}$$

where μ is the molecular viscosity.

2.2. Basic formulation

We solve the three-dimensional incompressible Navier–Stokes equations in a generalized coordinate system using a finite volume approach. Time integration of the discrete Navier–Stokes equations is achieved by an artificial compressibility approach [15] which is facilitated by a dual time stepping procedure at each physical time step. At time level $n+1$, sub-iteration k, the discrete representation of Eqs. (1) and (2) can be written as

$$\mathbf{A}(\mathbf{U}^{n+1,k+1} - \mathbf{U}^{n+1,k}) = -\mathbf{R}^{n+1,k}. \tag{4}$$

The flow variables $\mathbf{U}$ are advanced from time level $n(\mathbf{U}^{n+1,k=0} = \mathbf{U}^n)$ to time level $n+1$ over number of sub-iterations $k_{\max}(\mathbf{U}^{n+1,k=k_{\max}} = \mathbf{U}^{n+1})$. The primitive variable vector is defined as $\mathbf{U} = (p, u_i)^{\mathrm{T}}$ and $\mathbf{A}$ is the system Jacobian matrix. The corresponding residual vectors $\mathbf{R} = (R_{\mathrm{C}}, R_{\mathrm{M,i}})^{\mathrm{T}}$ can be written as

$$R_{\mathrm{c}}^{n+1,k} = \left[\frac{\partial u_i}{\partial x_i}\right]^{n+1,k}, \tag{5}$$

$$R_{\mathrm{M,i}}^{n+1,k} = \rho\left(\frac{3u_i^{n+1,k} - 4u_i^n + u_i^{n-1}}{\Delta t}\right) + \left[\frac{\partial}{\partial x_j}(\rho u_i u_j + p\delta_{ij} - \tau_{ij}) - \rho f_i\right]^{n+1,k}, \tag{6}$$

where R_{c} and $R_{\mathrm{M,i}}$ are the residuals for the continuity and momentum equations, respectively. Eq. (3) is solved approximately at each sub-iteration using an implicit technique based on incomplete LU decomposition [16]. For the spatial discretization, the inviscid fluxes in the governing equations are discretized using a low-diffusion flux splitting scheme (LDFSS) [17,18], while second-order central-differencing methods are used to discretize the viscous components. Higher-order spatial accuracy for the interface fluxes is achieved by using total variation diminishing (TVD) or weighted essentially non-oscillatory (WENO) variable extrapolation methods [19]. The effects of smaller subgrid fluctuations are modeled using a Smagorinsky subgrid eddy viscosity [20]. The present flow solver uses METIS [21] to partition a general multi-block grid over the number of allowable processors. Message-passing interface (MPI) communication routines are used to pass information among the processors. The incompressible flow solver and its components have been validated for a range of model problems [17].

2.3. Immersed boundary method

The present immersed boundary method is based on the *direct forcing* technique proposed by Fadlun et al. [10]. They presented a novel formulation of Peskin's immersed boundary method [7] that has been adapted for use with our Navier–Stokes solver. The key concept is that the velocity of the fluid must be equal to the

velocity of the moving body at its surface. This is achieved by applying a forcing function to the momentum equations such that the desired boundary velocity is recovered as the solution at each time step. We used this basic idea as the starting point but define the forcing in the context of the sub-iteration method discussed in Section 2.2.

We introduce forcing terms that relax the velocity and pressure solutions near and within immersed-boundaries to interpolated values $u_{B,i}$ and p_B over the course of the sub-iterations. The residuals defined in Eqs. (5) and (6) are redefined as follows:

$$R_{c,B}^{n+1,k} = (1 - G(\Phi^{n+1}))R_c^{n+1,k}\big|_{\text{Eq. (5)}} + G(\Phi^{n+1})\left(\frac{p^{n+1,k} - p_B^{n+1,k}}{\beta^2 \Delta t}\right), \tag{7}$$

$$R_{M,i}^{n+1,k} = (1 - G(\Phi^{n+1}))R_{M,i}^{n+1,k}\Big|_{\text{Eq. (6)}} + G(\Phi^{n+1})\left(\frac{u_i^{n+1,k} - u_{B,i}^{n+1,k}}{\Delta t}\right), \tag{8}$$

where $G(\Phi)$ is a sharp Heaviside function, $\Phi(\mathbf{x}, t)$ is a signed distance function at location $\mathbf{x}$ and time t, and β is an artificial compressibility parameter. Specific forms for $G(\Phi)$, $\Phi(\mathbf{x}, t)$, $u_{B,i}$, and p_B are discussed in the sections that follow. In the absence of an immersed body, the sub-iterations serve to eliminate time linearization errors and to satisfy a divergence-free constraint on the velocity field. When an immersed body is included, the sub-iterations also serve to adjust the velocity and pressure fields to the updated motion of the immersed body.

2.3.1. Classification of computational nodes

The quantities $u_{B,i}$ and p_B are defined according to interpolation methods presented later and are generally functions of flow properties at *field points*, as opposed to *band points* or *interior points* as shown in Fig. 1. We develop a classification algorithm for computational nodes based on the signed distance function $\Phi(\mathbf{x}, t)$, which is less than zero for cells within the immersed body and greater than zero for cells outside the body.

The Heaviside function $G(\Phi(\mathbf{x}, t))$ is defined to be one for points just outside the immersed body and within the immersed body and is zero otherwise. The calculation of the Heaviside function is initiated by first initializing $G(\Phi(\mathbf{x}_k,t)) = 0$ for all points $\mathbf{x}_k$. Then, given a point $\mathbf{x}_k$, if $\Phi(\mathbf{x}_k,t) > 0$ and if any $\Phi(\mathbf{x}_{k'},t) < 0$, where $\mathbf{x}_{k'}$ is a nearest neighbor of $\mathbf{x}_k$, then $G(\Phi(\mathbf{x}_k,t))$ is set to 1. If $\Phi(\mathbf{x}_k,t) \leqslant 0$, then $G(\Phi(\mathbf{x}_k,t))$ is also set to 1. The set of *nearest neighbors*, for a structured grid discretized according to a cell-centered finite volume method, is generally defined as the 26 cells that are immediately adjacent to a particular mesh cell, though smaller subsets can be used. Finally, we can define the Heaviside function as

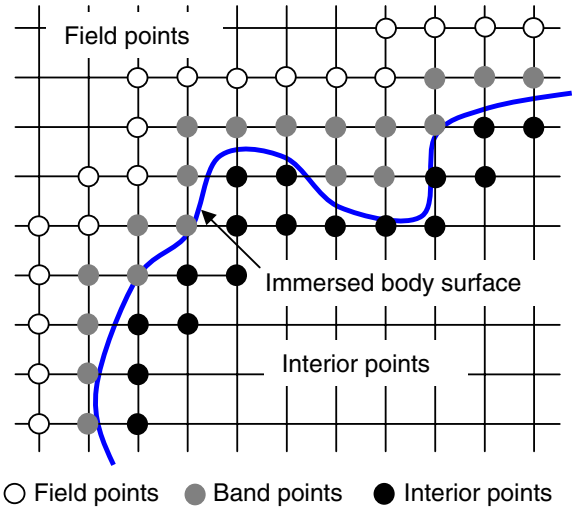

Fig. 1. Schematic illustrating classification of cell-centered points for a complex immersed body surface. Open, gray and close circles represent field (Ω_F), band (Ω_B) and interior points (Ω_I), respectively, and the thick line represents an immersed body surface.

$$G(\Phi(\mathbf{x}_k,t)) = \begin{cases} 0 & \mathbf{x}_k \in \Omega_F, \\ 1 & \mathbf{x}_k \notin \Omega_F, \end{cases} \tag{9}$$

where Ω_F represents the set of the node points shown as the open circles in Fig. 1. Classification of the node points can be summarized as follows:

- Field points : $\mathbf{x}_k \in \Omega_F$ if $\Phi(\mathbf{x}_k,t) > 0$ and $G(\Phi) = 0$, (10)
- Band points : $\mathbf{x}_k \in \Omega_B$ if $\Phi(\mathbf{x}_k,t) > 0$ and $G(\Phi) = 1$, (11)
- Interior points : $\mathbf{x}_k \in \Omega_I$ if $\Phi(\mathbf{x}_k,t) \leqslant 0$ and $G(\Phi) = 1$ (12)

where Ω_B and Ω_I represents the set of the node points shown as the gray and closed circles in Fig. 1, respectively. The zero iso-surface of the signed distance function defines the immersed body surface. Note that the flow properties at band and interior points are calculated from Eqs. (7) and (8) in the limit of $G(\Phi) = 1$.

2.3.2. Surface definition and distance function

The first step in the simulation is to define an immersed body as a collection of surface points. This can be done using a computer aided design (CAD) format or through other means, but the key is that a list of surface points and outward-pointing unit normal vectors is created for each separate component of the body, as different components may move at different rates. The next step is to calculate the signed distance from each point $\mathbf{x}_k \in \Omega$ to the nearest surface point $\mathbf{x}_{s,l(k)} \in \Gamma_l$ on the lth body component as shown in Fig. 2. This is accomplished first through the use of approximate nearest-neighbor (ANN) searching techniques [22], which return the unsigned distance. In practice, this is done only for the number of field points that are within a *bounding box* surrounding the particular surface, as it is these points that are likely to be influenced immediately by the body. Distances outside the bounding box are assigned to be a very large positive number. Initially, the signed distance function $\Phi_l(\mathbf{x}_k,t)$ was obtained by multiplying the unsigned distance with the sign of the dot product of the distance vector with the outward normal vector $\mathbf{n}_{s,l(k)}$:

$$\Phi_l(\mathbf{x}_k,t) = \mathrm{sgn}((\mathbf{x}_k - \mathbf{x}_{s,l(k)}) \cdot \mathbf{n}_{s,l(k)}) \times \left\|\mathbf{x}_k - \mathbf{x}_{s,l(k)}\right\|, \tag{13}$$

where $\mathrm{sgn}(\phi)$ returns a value of 1 for each non-negative element and -1 for each negative element of ϕ, and $\|\ \|$ denotes the magnitude of the vector.

This simple procedure was found not to work properly for some very complex CAD objects. Usually, the CAD objects are defined as triangular surface elements that contain each vertex and face normal vector. If a nearest surface point at a given field point is located at an edge or vertex, the simple signed distance function may not be calculated correctly. Therefore, we augment the original set of face-centered surface points by including all vertices. At each vertex, we define an angle-weighted pseudo-normal vector [23] as follows. For a given vertex $\mathbf{x}_{v,l}$, we can determine the triangular elements shared with the vertex and can calculate

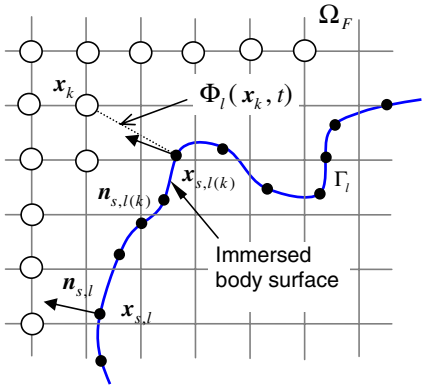

Fig. 2. Schematic illustrating approximate nearest point and the corresponding signed distance function $\Phi_l(\mathbf{x}_k,t)$. Open and closed circles represent cell-centered grid points for field points Ω_F and surface node points for the lth immersed body surface Γ_l.

the incident angle $\alpha_{i,l}$ for each element with outward-pointing face normal vector $\mathbf{n}_{i,l}$ in Fig. 3. The angle weighted pseudo-normal vector $\mathbf{n}_{v,l}$ at the vertex can be defined as

$$\mathbf{n}_{v,l} = \frac{\sum_i \alpha_{i,l}\mathbf{n}_{i,l}}{\left\|\sum_i \alpha_{i,l}\mathbf{n}_{i,l}\right\|}, \tag{14}$$

where i denotes the triangle elements that surround the vertex. Based on the pseudo-normal vectors at the vertices and the face normal vectors $\mathbf{n}_{i,l}$ at the element centers $\mathbf{x}_{i,l}$, we can determine an inside/outside decision using Eq. (13). This procedure essentially averages local fluctuations in the outward normal that could result from small features in the CAD file.

Our inside/outside decision algorithm is based on finding the nearest face-center or vertex and the corresponding normal vector or pseudo-normal vector instead of searching for the closest point on the closest element. As such, the present decision procedure is simpler than the searching algorithms used in computational geometry. Even with the use of the pseudo-normal vector, the decision process can still fail for very complex CAD geometries because of holes or gaps in the geometry. To help correct this problem, we have implemented a consensus algorithm [24] that compares results for neighboring points to ensure smoothness. As mentioned above, for a given point, we have to find the nearest surface point using the ANN algorithm. In the process of doing this search, the ANN algorithm can also give us a cloud of nearby points $\Gamma_{l(k)} \subset \Gamma_l$, in order of closest distance. Therefore, we can consider a committee of these nearby points to determine the correct inside/outside classification. For a given number of committee points N, we can define N_i and N_o as the number of inside and outside decisions, respectively. The corrected signed distance function $\tilde{\Phi}_l(\mathbf{x}_k, t)$ is defined as

$$\tilde{\Phi}_l(\mathbf{x}_k, t) = \mathrm{sgn}(N_\mathrm{o} - N_\mathrm{i})|\Phi_l(\mathbf{x}_k, t)|, \tag{15}$$

where N is initially set to be an odd number and $|\ |$ denotes the absolute value of the original signed function computed at point k.

To define a global signed distance function $\Phi(\mathbf{x}_k,t)$ at any given mesh point, a simple priority rule is exercised. First, the global distance function is initialized to a large number. Then, the global signed distance function at a particular point is taken as the minimum of the individual signed distance functions for each component l at that point:

$$\Phi(\mathbf{x}_k, t) = \min_l(\tilde{\Phi}_l(\mathbf{x}_k, t)). \tag{16}$$

The collections of points that comprise the surfaces are allowed to move according to prescribed rate laws. Some examples for human walking motion are discussed in Section 3.6. Once new surface positions are obtained, the preceding steps for determining the signed-distance and Heaviside functions have to be repeated.

For complex CAD objects, the procedure for determining the signed distance function is summarized as follows:

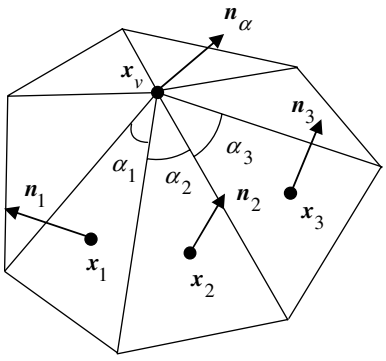

Fig. 3. The incident angles $\{\alpha_1, \alpha_2, \alpha_3, \ldots\}$ of triangular elements shared with a vertex $\mathbf{x}_v$. The corresponding outward normal vectors are $\{\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3, \ldots\}$ at the node points $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots\}$.

- Determine a set of face-centered surface points and outward normal vectors (this information is usually supplied by the CAD program). Augment this set by adding the vertices with pseudo-normal vectors calculated using Eq. (14).
- For a given point in the field, find the nearest surface point and a cloud of nearby surface points using the ANN searching algorithm.
- Compute the signed distance between the given point and the nearest surface point using Eq. (13).
- Compute the signed distance between the given point and the cloud of nearby surface points using Eq. (13).
- Correct the signed distance to the nearest surface point using the consensus algorithm in Eq. (15).
- Repeat the above procedure for each component.
- Compute the global signed distance function using Eq. (16).

To assess the different variations in the point classification scheme, a human mannequin is visualized using the zero level set of the signed distance function. The immersed surface is rendered as a stereolithography (STL) file using a trial version of Mannequin Elite ver.1.2 (NextGen Ergonomics Inc.). To improve the surface resolution of human body, additional triangle elements are inserted, considering the edge distance of the largest triangle element as a metric for refinement. Consistent face-based outward-pointing normal vectors were obtained using ADMesh ver.0.95 (http://www.varlog.com/products/admesh/) for the refined surface meshes. The size of human model is 1.74 m tall and 0.6 m wide at chest. The average edge distance of the triangle mesh is about 0.01 m. The entire human model is separated into 15 body segments. Two different data sets are considered. The first is the original set of face-centered surface points with face normal vectors and the other is an extended data set that also includes vertices of triangular elements with pseudo-normal vectors. The total numbers of surface points for the original data set and the extended data set are 86,880 and 130,444, respectively. Fig. 4 compares the effects of the different classification schemes (original data set with and without consensus algorithm (schemes 1 and 2); extended data set with and without consensus algorithm (schemes 3 and 4)) on the immersed body rendering. A uniformly-spaced background mesh ($\Delta h = 0.01$ m) is used in the assessment, and the total number of cells is 640,000. For purposes of comparison, we assume that the inside/outside decision obtained using the extended data set with the consensus algorithm is the correct one. Based on this reference rendering, incorrect inside/outside decisions using the other schemes are highlighted in Fig. 4 as dark contours,

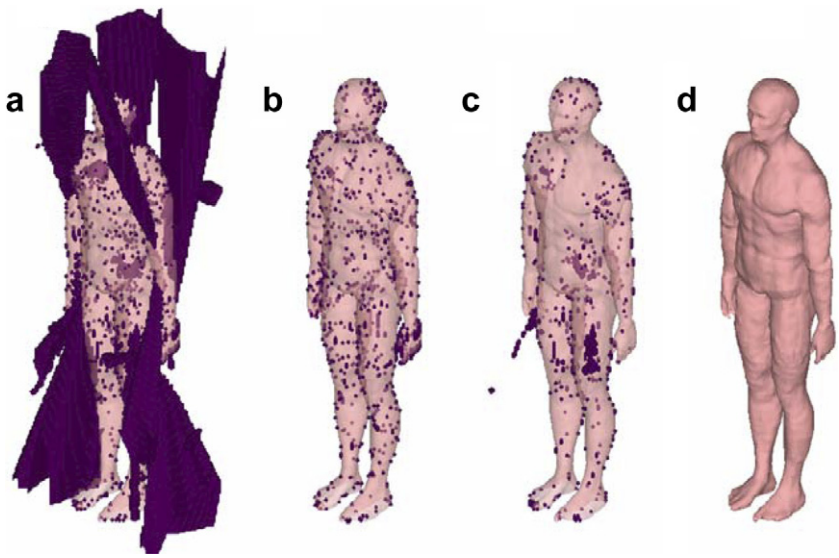

Fig. 4. Assessment of surface definitions using the different schemes: (a) face-centered data set, (b) face-centered data set with consensus algorithm, (c) extended data set and (d) extended data set with consensus algorithm. The iso-surface of the human rendering is extracted from the extended data set with consensus algorithm. The dark iso-surfaces show misclassified cells based on the rendering of case (d).

Table 1
Misclassification on the surface definitions using the different schemes

	Scheme 1	Scheme 2	Scheme 3	Scheme 4
Number of surface points	86,880	86,880	130,344	130,344
Number of misclassified cells	71,304	796	990	–
Percentage of misclassifications (%)	11.1	1.24	1.55	–

Table 2
Volume recovery of the surface definition at the different background grid resolutions

Resolutions	$\Delta h = 0.04$	$\Delta h = 0.02$	$\Delta h = 0.01$	$\Delta h = 0.005$
Number of cells	10,000	80,000	640,000	5,120,000
Volume ($\times 10^{-2}$ m^3)	6.157	7.206	7.528	7.662
Volume recovery (%)	80.4	94.0	98.3	–

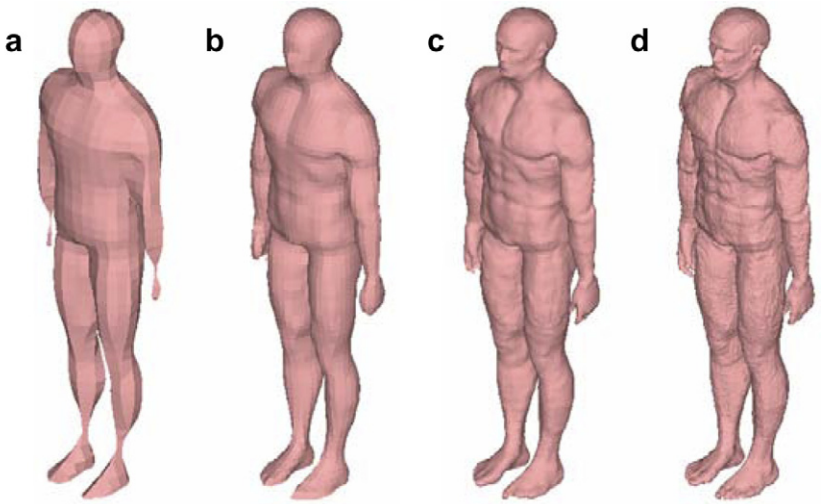

Fig. 5. The sensitivity of the classification procedure to the relative background grid resolution: (a) $\Delta h = 0.04$, (b) $\Delta h = 0.02$, (c) $\Delta h = 0.01$ and (d) $\Delta h = 0.005$.

and quantitative comparisons in terms of the number of points classified incorrectly are shown in Table 1. The use of the consensus algorithm with the face-centered data set provides substantial improvement (1.24% misclassified cells versus 11.1%) as does the use of the extended data set alone (1.55% misclassified cells versus 11.1%).

The sensitivity of the surface rendering to the relative resolution of the immersed surface and background grid is assessed by varying the background grid resolution. Fig. 5 shows the immersed body rendering extracted from the extended surface data set with the consensus algorithm on four different background grids. For the finest resolution (5.12 million cells) in Fig. 5(d), the fine details of the surface are well captured since the background mesh spacing is smaller than the average surface resolution of 0.01 m. As background grid is coarsened, the detected surface maintains the general features of the mannequin but loses the details. Table 2 summarizes the total number of cells, the occupied volume of the immersed body, and the volume recovery ratio (measured relative to the surface rendering at the finest background resolution) as a quantitative measure of the sensitivity of the present classification scheme. When the background resolution is less than $\Delta h = 0.02$ m, the volume of the immersed body can be recovered to 94% of the original volume. As a rule of practice, we will typically require that the average surface resolution be at least twice as fine as the average mesh spacing to ensure that the inside/outside classification process is performed correctly. This restriction

may be lessened by searching for the true distance to the nearest point on the faceted surface, instead of the distance to the nearest face-center or vertex.

2.3.3. Interpolation methods

In our interpretation of the direct forcing technique [10], the Navier–Stokes equations are not solved for *band* and *interior points*. Rather, the solution for the flow properties at these points is forced as a function of the *field point* solutions in neighboring cells and the surface velocities. As it is unlikely that a cell-centered grid will coincide with the surface, the velocity solutions within the band points need to be determined using interpolation techniques. Many researchers have investigated different methods for performing this interpolation, primarily in the context of Cartesian grid topologies [1,10–14]. As our techniques are designed for use on arbitrary grid topologies (even unstructured meshes), a different approach is proposed. For interior points, the forcing velocity is simply set to that of the nearest surface point, and the forcing pressure is set to a free-stream value.

2.3.3.1. Velocity interpolation. A linear interpolation scheme for velocity interpolation near the immersed surfaces has often been used for direct forcing in the immersed boundary treatment [10]. This rationale is based on the assumption of a one-dimensional linear velocity profile near a solid surface and is appropriate for some low Reynolds number flows. Several studies [12–14] that compare the accuracy and efficiency of different velocity interpolation methods have been published recently. In the present work, we introduce the concept of *tangency correction* by decomposing the velocity into the tangential and normal component along the outward normal direction to the immersed surface. A velocity distribution is defined in terms of the normal coordinate n, which is not necessarily equal to the distance function itself but should be nearly so. Given that the vector $\mathbf{n}$ defines the local surface normal vector at the nearest surface point $\mathbf{x}_S$ to the band point $\mathbf{x}_B$ as shown in Fig. 6, a general velocity distribution can be decomposed as follows:

$$u_{B,i}(n) = u_{T,i}(n) + u_{N,i}(n) + u_{S,i}. \tag{17}$$

where the subscripts T and N denote the tangential and normal directions in a orthogonal plane to the surface, respectively, and $u_{S,i}$ is the velocity at the nearest point on the immersed surface. The tangential velocity component (defined relative to the velocity at the nearest surface point), is expressed as a power-law function of the normal coordinate:

$$u_{T,i}(n) = \left[u_{T,i}(d_I) + \underbrace{\left(1 - \frac{n}{d_I}\right)\left(k u_{T,i}(d_I) - d_I \left.\frac{\mathrm{d}u_{T,i}}{\mathrm{d}n}\right|_{n=d_I} \right)}_{\text{second-order correction}} \right] \left(\frac{n}{d_I}\right)^k. \tag{18}$$

Here, the tangential velocity and its gradient at the interpolation points can be obtained as

$$u_{T,i}(d_I) = (u_i(d_I) - u_{S,i}) - [(u_j(d_I) - u_{S,j})n_j]n_i \quad \text{and} \quad \left.\frac{\mathrm{d}u_{T,i}}{\mathrm{d}n}\right|_{n=d_I} \approx \frac{\partial u_i(d_I)}{\partial x_j} n_j - \left[\left(\frac{\partial u_k(d_I)}{\partial x_j} n_j\right) n_k\right] n_i, \tag{19}$$

where a summation is indicated by repeated indices.

The normal velocity component is expressed as a cubic function of the normal coordinate since the second derivative of the wall normal velocity should vanish at the surface in order to conserve momentum [10]:

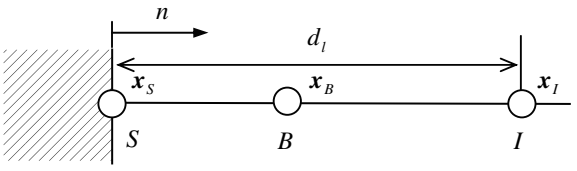

Fig. 6. Interpolation stencil near the band point B along the outward normal direction n. The characters S and I represent the immersed surface point and the interpolation point, respectively.

$$u_{\mathrm{N,i}}(n) = \left[u_{\mathrm{N,i}}(d_{\mathrm{I}}) + \underbrace{\frac{1}{2}\left(1 - \left(\frac{n}{d_{\mathrm{I}}}\right)^2\right)\left(u_{N,i}(d_{\mathrm{I}}) - d_{\mathrm{I}}\left.\frac{\mathrm{d}u_{\mathrm{N,i}}}{\mathrm{d}n}\right|_{n=d_{\mathrm{I}}}\right)}_{\text{second-order correction}} \right]\left(\frac{n}{d_{\mathrm{I}}}\right), \tag{20}$$

The interpolated normal velocity and its gradient can be evaluated as

$$u_{\mathrm{N,i}}(d_{\mathrm{I}}) = [(u_j(d_{\mathrm{I}}) - u_{\mathrm{S},j})n_j]n_i \quad \text{and} \quad \left.\frac{\mathrm{d}u_{\mathrm{N,i}}}{\mathrm{d}n}\right|_{n=d_{\mathrm{I}}} = \left[\left(\frac{\partial u_k(d_{\mathrm{I}})}{\partial x_j}n_j\right)n_k\right]n_i. \tag{21}$$

In this, $u_{\mathrm{i}}(d_{\mathrm{I}})$ is an interpolated velocity, $\frac{\partial u_k(d_{\mathrm{I}})}{\partial x_j}$ is a velocity gradient, and d_I is the location in the direction of the normal coordinate at which these quantities are defined (the *interpolation point*). Methods for evaluating these quantities are presented in Section 2.3.3.3. To arrive at the forcing velocity for a particular band point$\mathbf{x}_k$, one simply evaluates the interpolation polynomials at $n = \Phi(\mathbf{x}_k, t)$. The terms in braces in Eqs. (18) and (20) are second-order corrections. These allow the functional forms to match the slope of the imposed velocity profile at the interpolation point d_I.

The velocity distribution for the tangential component is written in terms of a general power law $\sim n^k$, rather than the common assumption of a linear velocity profile, since small values of k ($k = 1/7$ or $1/9$) approximate the logarithmic distribution expected in the near wall region for an attached turbulent flow. While this feature generally promotes flow attachment to the immersed surface, it is not a slip-wall condition, as for $n = \Phi(\mathbf{x}_k, t)$ near enough to zero; the velocity will go to zero as long as k is finite. Nor does it necessarily prohibit separation of the flow. The normal velocity distribution is defined so that its second-derivative is maintained at the surface ($n = 0$) in order to satisfy the Neumann conditions for wall normal pressure gradient as discussed in Fadlun et al. [10]. Within this framework, there are several ways to control the velocity distribution. The general trend is to promote more flow attachment to the surface with lower values of k. The proper specification of this parameter is problem-dependent. Numerical experiments indicate that very low Reynolds number flows ($Re < 1000$) seem to require a linear distribution ($k = 1$), whereas high Reynolds number flows ($Re > 10{,}000$) appear to require a $k = 1/7$ or $k = 1/9$ power-law distribution in order to inhibit massive flow separation.

2.3.3.2. Pressure interpolation. A general procedure for interpolating the pressure to the surface is given by the following. We assume that the difference between the pressure and its free-stream value is represented as a second-order polynomial in the normal distance$n$ for points within the band:

$$p_{\mathrm{B}}(n) - p_\infty = a + bn + cn^2 \tag{22}$$

with boundary conditions at $n = 0$ given as

$$p_{\mathrm{B}}(0) - p_\infty = p_w - p_\infty = a \quad \text{and} \quad \left.\frac{\mathrm{d}p_B}{\mathrm{d}n}\right|_{n=0} = b, \tag{23}$$

where b is related to the acceleration of the surface (zero for constant velocity or no movement). Evaluating these expressions at the location of the interpolation point outside the band gives

$$p_{\mathrm{B}}(d_{\mathrm{I}}) = a + bd_{\mathrm{I}} + cd_{\mathrm{I}}^2 \quad \text{and} \quad \left.\frac{\mathrm{d}p_{\mathrm{B}}}{\mathrm{d}n}\right|_{n=d_{\mathrm{I}}} = b + 2cd_{\mathrm{I}}. \tag{24}$$

Given a value for b, this system can be solved for the unknowns a and c, yielding a final form for the interpolation polynomial:

$$p_{\mathrm{B}}(n) = p_{\mathrm{B}}(d_{\mathrm{I}}) + bn - \frac{1}{2}d_{\mathrm{I}}\left(\left.\frac{\mathrm{d}p_{\mathrm{B}}}{\mathrm{d}n}\right|_{n=d_{\mathrm{I}}} + b\right)\left(1 - \left(\frac{n}{d_{\mathrm{I}}}\right)^2\right). \tag{25}$$

This equation is evaluated at the band point $n = \Phi(\mathbf{x}_k, t)$ to give the desired result. The predicted value for the wall pressure is determined by evaluating Eq. (25) at $n = 0$.

2.3.3.3. Determination of information at the interpolation point. The preceding developments hinge on the determination of flow properties and gradients at a certain distance d_{I} away from the immersed surface. Given a band point $\mathbf{x}_k$ and a list of nearest neighbors to that point $\mathbf{x}_l$ in Fig. 7, a merit function w_l is defined as follows:

$$w_l = \frac{1}{\sqrt{\|\mathbf{x}_l - \mathbf{x}_k\|^2 - [(\mathbf{x}_l - \mathbf{x}_k)\cdot\mathbf{n}]^2 + \varepsilon}} \quad \text{if } (\mathbf{x}_l - \mathbf{x}_k)\cdot\mathbf{n} > 0, \quad \text{and } w_l = 0 \text{ otherwise,} \tag{26}$$

where $(\mathbf{x}_l - \mathbf{x}_k)\cdot\mathbf{n}$ is the projection of the distance d_l from $\mathbf{x}_k$ to $\mathbf{x}_l$ in the direction of the outward normal, and $\|\mathbf{x}_l - \mathbf{x}_k\|$ is the magnitude of the distance vector itself. Note that ε is added to the denominator in Eq. (26) in order to avoid a numerical singularity in defining the merit function. If point $\mathbf{x}_l$ is located directly along the outward normal line corresponding to band point $\mathbf{x}_k$, and if $(\mathbf{x}_l - \mathbf{x}_k)\cdot\mathbf{n}$ is positive, meaning that point $\mathbf{x}_l$ is further away from the surface than point $\mathbf{x}_k$, then the merit function returns a very large value ($\sim 1/\varepsilon$, where ε is assumed to be 10^{-12} in the present study).

The actual calculation of w_l is performed in three stages. First, only field points are considered as members of the list of nearest neighbors. Then, w_l is calculated according to Eq. (26), and the sum of the weights $\sum_m w_m$ is calculated. If this sum is non-zero, then the weight function ω_l for each nearest-neighbor is determined as

$$\omega_l = w_l \Big/ \sum_m w_m. \tag{27}$$

Otherwise, the process is repeated, now considering both field points and other band points as members of the list of nearest neighbors. If this application also results in no viable interpolation points being found, then the band point $\mathbf{x}_k$ is effectively set to an interior point.

The location at which interpolated properties are defined, d_{I}, is calculated for a particular field point as

$$d_{\mathrm{I}} = \sum_l \omega_l d_l = \sum_l \omega_l (\mathbf{x}_l - \mathbf{x}_k)\cdot\mathbf{n}. \tag{28}$$

Note that this distance is in the direction of the normal coordinate n. A specific flow property q and its gradient $\frac{\mathrm{d}q}{\mathrm{d}n}$ at the interpolation point are calculated by the weighting functions as follows:

$$q(d_{\mathrm{I}}) = \sum_l \omega_l q_l \quad \text{and} \quad \left.\frac{\mathrm{d}q}{\mathrm{d}n}\right|_{n=d_{\mathrm{I}}} = \sum_l \omega_l \left(\frac{\partial q}{\partial x_j}\right)_l n_j. \tag{29}$$

Note that the normal vector in Eq. (29) refers to the band point in question, not the field point l.

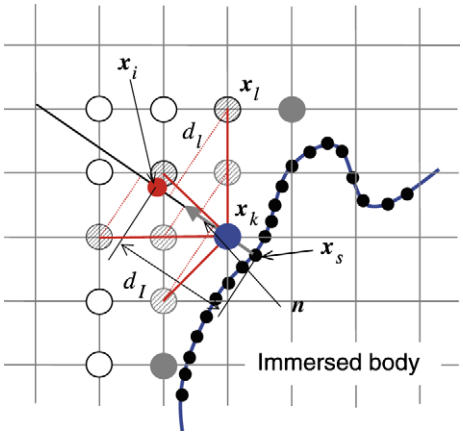

Fig. 7. Schematic determination of the distance d_{I} between the interpolation point $\mathbf{x}_{\mathrm{i}}$ and surface node point for a given band point $\mathbf{x}_k$ using the projected distance d_l from neighbor points $\mathbf{x}_l$ to outward normal line based on surface normal vector $\mathbf{n}$ at the immersed surface node $\mathbf{x}_{\mathrm{s}}$. Large closed circle represent the band point to be interpolated with the information at neighbor point. Hatched black and gray circles represent the field points and band points associated with the present determination, respectively.

3. Numerical examples

3.1. Flow over a circular cylinder

To quantify the accuracy of the present immersed boundary method, we have computed several classical benchmark problems used to validate incompressible-flow solvers. The first case is low Reynolds number, 2D flow over a cylinder, which exhibits vortex shedding at Reynolds numbers (Re) above ~43 and is steady for lower Reynolds numbers. Reynolds numbers (based on the cylinder diameter D and the free stream velocity u_∞) up to 200 are considered in the present study. The size of computation domain is $80D \times 80D$, where D is the diameter of the cylinder. The numbers of grid points are 321×321 in the streamwise and transverse directions, respectively. A uniform 41×41 square mesh is used in the $D \times D$ control volume surrounding the cylinder for $Re = 40$, while a 81×81 uniform mesh is used for $Re = 100$ and 200. The mesh is stretched to the inflow, outflow and far-field boundaries. A Dirichlet boundary condition $u/u_\infty = 1$, $v = 0$ is applied at the inflow and far-field boundaries, and a subsonic outflow condition is used at the outflow boundary. A power $k = 1$ is used for this case and for the case involving flow over a sphere.

We have performed a grid refinement study for 2D flow over a cylinder in order to demonstrate the second-order accuracy of the present method. The Reynolds number of this flow is set to $Re = 40$. Four uniformly spaced, successively finer mesh sizes are used for error analysis, with 21^2, 41^2, 81^2 and 161^2 grid points in the vicinity of the cylinder. The finest mesh solution is considered to be the *exact* solution. The L_2 norm error can be defined as

$$\varepsilon_g^2 = \frac{1}{n} \sum_n \left| \phi_n^g - \phi_n^{g=161} \right|^2, \tag{30}$$

where ϕ denotes solution variables such as velocity and pressure, g represents different grid levels, and n is the number of grid points within the $D \times D$ control volume surrounding the cylinder. The results of the grid refinement study are summarized in Fig. 8, which shows the variation of the error with grid spacing Δh in a log–log scale. It is evident from Fig. 8(a) that both the first- and second-order velocity interpolation methods converge at a second-order rate. A local error in the band points is defined to investigate the effect of the interpolation method very near the immersed surface (Fig. 8 (b)). Results indicate that the second-order velocity interpola-

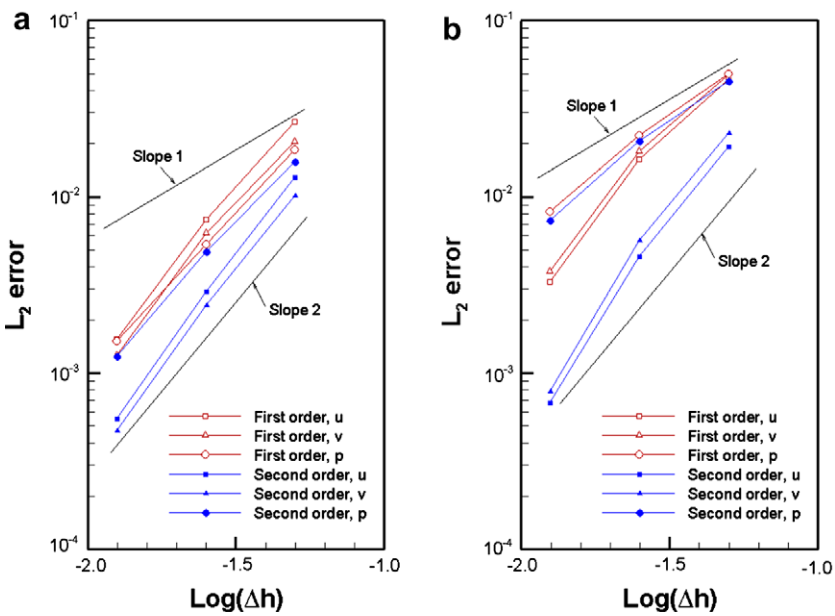

Fig. 8. Grid convergence study for flow over a cylinder: (a) Interior flow field ($D \times D$) and (b) band points in the vicinity of the cylinder.

tion provides essentially second-order convergence. However, the convergence rate of the pressure is closer to first order. This is because the pressure gradient near the immersed surface is defined using a first-order finite difference. The convergence rate analysis confirms that the present method is globally second-order accurate.

Direct evaluation of surface forces in our immersed-boundary method is difficult, as the underlying surface mesh may contain no structure and as not all points on the surface are directly used by neighboring band points. Recently, Balaras [13] proposed to compute the forces on the cylinder surface by applying the momentum theorem to the control volume V. The corresponding surface is bounded by the surface $S = S_o \cup S_b$, where S_o is a fixed surface and S_b is the immersed body. The force vector F_i is given by

$$F_i = \int_V \frac{\partial(\rho u_i)}{\partial t}\,\mathrm{d}V + \int_{S_o} (\rho u_i u_j + p\delta_{ij} - \tau_{ij}) n_j \,\mathrm{d}s_o. \tag{31}$$

Then, the drag (C_D) and lift coefficients (C_L) are computed by,

$$C_D = \frac{F_x}{(1/2)\rho u_\infty^2 D} \quad \text{and} \quad C_L = \frac{F_y}{(1/2)\rho u_\infty^2 D} \tag{32}$$

where x and y denote the streamwise and normal directions in Cartesian coordinates. Table 3 presents drag coefficient, reattachment length (L/D) and separation angle (θ) for the present method at $Re = 20$ and 40 compared with literature values. Both first- and second-order methods show good agreement for reattachment length and drag coefficient. The first-order method under-estimates the separation angle but the second-order method shows a significant improvement. This is because matching the velocity gradient at the interpolation point enhances the prediction of flow separation, which is closely related to the inflection of the velocity profile near the surface. Fig. 9 shows the pressure coefficient and spanwise vorticity variations along the cylinder surface at $Re = 40$. It is shown in Fig. 9 that the pressure coefficient obtained using the first- or second-order

Table 3
Drag coefficient, reattachment length and separation angle for flow over a cylinder

	$Re = 20$			$Re = 40$		
	L/D	θ	C_D	L/D	θ	C_D
Ye et al. [5]	0.92	–	2.03	2.27	–	1.52
Fornberg [25]	0.91	–	2.00	2.24	–	1.50
Dennis and Chang [26]	0.94	43.7	2.05	2.35	53.8	1.52
Calhoun [27]	0.91	45.5	2.19	2.18	54.2	1.62
Russell and Wang [28]	0.94	43.3	2.13	2.29	53.1	1.60
Body fitted grid	0.92	43.8	1.98	2.21	53.6	1.49
Present (first-order)	0.89	35.3	2.03	2.24	45.1	1.52
Present (second-order)	0.90	40.8	2.02	2.25	51.0	1.52

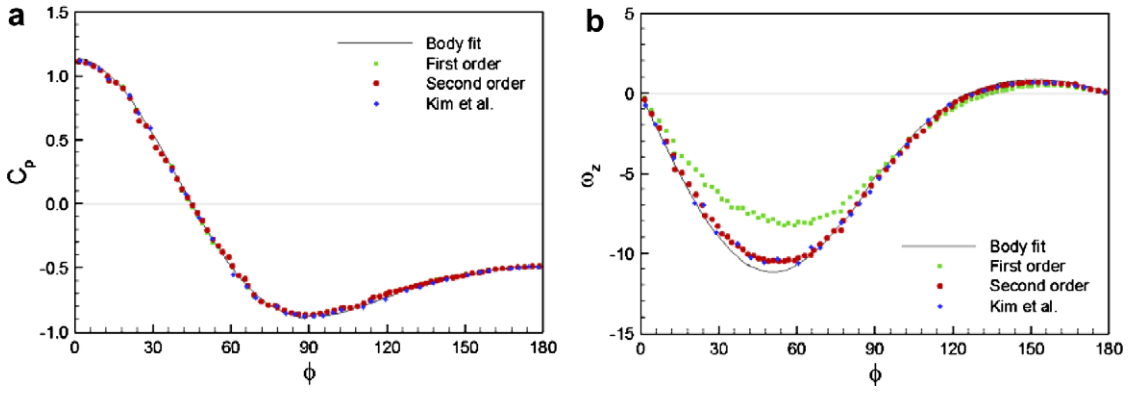

Fig. 9. Flow variables along the cylinder surface at $Re = 40$; (a) wall pressure coefficient (C_p) and (b) wall spanwise vorticity (ω_z).

methods agrees well with that obtained on a body fitted grid using the same computational code and with Kim et al.'s [11] results obtained using a different immersed boundary method. However, the magnitude of the wall vorticity using the first order interpolation is underestimated at $20° \leqslant \phi \leqslant 90°$. Predictions from the second-order method agree well with the other results.

To assess the present immersed-body approach for unsteady flows shed by bluff objects, we consider 2D flow over a cylinder at $Re = 100$ and 200. Table 4 shows results for drag and lift coefficients and Strouhal number ($St = fD/u_\infty$). Note that f is shedding frequency. As shown in Table 4, the present second-order method accurately predicts unsteady characteristics such as shedding frequency and oscillations of C_D and C_L for $Re = 100$ and 200. Fig. 10 shows streamlines super-imposed on pressure coefficient contours for $Re = 100$. The present method clearly captures vortex shedding behind the cylinder.

3.2. Flow over an in-line oscillating cylinder

To validate the present method for a moving boundary problem, an in-line oscillating cylinder in a fluid at rest is considered. The two characteristic parameters for this flow are the Reynolds number, $Re = U_{max}D/\nu$, where U_{max} is the maximum velocity of the cylinder, and the Keulergan–Carpenter number, $KC = U_{max}f/D$, where f is the frequency of the oscillation. In-line oscillatory motion of the cylinder is given by a simple harmonic relationship

$$x_c(t) = -A\sin(2\pi ft), \tag{33}$$

where $x_c(t)$ is the position of the cylinder center in the direction of oscillation and A is the amplitude of oscillation. The two characteristic parameters are chosen as $Re = 100$ and $KC = 5$ in order to compare the present

Table 4
Drag and lift coefficients, and Strouhal number for flow over a cylinder

	$Re = 100$			$Re = 200$		
	C_D	C_L	St	C_D	C_L	St
Kim et al. [11]	1.33	±0.32	0.165	–	–	–
Calhoun [27]	1.35 ± 0.014	±0.300	0.175	1.17 ± 0.058	±0.67	0.202
Russell and Wang [28]	1.38 ± 0.007	±0.322	0.169	1.29 ± 0.022	±0.50	0.195
Rosenfeld et al. [29]	–	–	–	1.31 ± 0.04	±0.65	0.20
Wright and Smith [30]	–	–	–	1.33 ± 0.04	±0.68	0.196
Braza et al. [31]	1.36 ± 0.015	±0.25	–	1.40 ± 0.05	±0.75	–
Liu et al. [32]	1.35 ± 0.012	±0.339	0.164	1.31 ± 0.049	±0.69	0.192
Present (second-order)	1.34 ± 0.011	±0.315	0.164	1.36 ± 0.048	±0.64	0.191

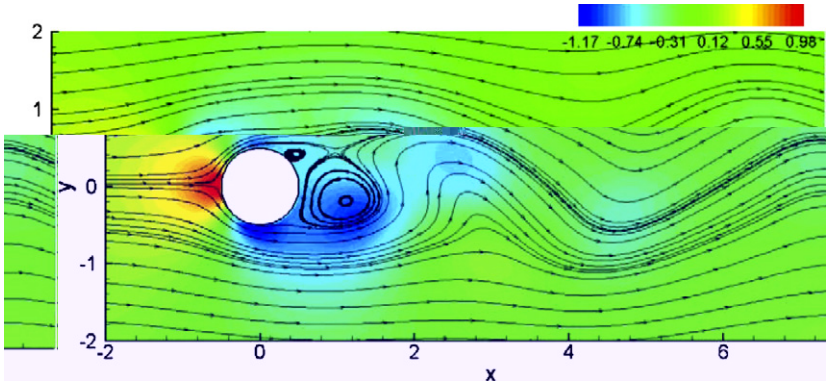

Fig. 10. Snapshot of unsteady wake flow over a cylinder (Re = 100). Contour and line show the pressure coefficient and stream trace line, respectively.

results with experimental data of Dütsch et al. [33]. The computational domain is $50D \times 30D$ in the oscillatory and transverse directions with 480×240 grid points. A similar mesh was used by Yang and Balaras [34] in their calculations of this flow field. The center of the cylinder is initially located at the center of the computational domain. A uniform 41×41 mesh is used near the cylinder. Neumann boundary conditions for the velocity and a zero gauge pressure condition are used at all far-field boundaries. As suggested by Yang and Balaras [34], a non-zero pressure gradient $(\partial p/\partial n)|_{n=0} = -(Du_i/Dt)n_i$ at the moving body surface is enforced in Eq. (25) in order to obtain the correct behavior of the pressure field near the immersed boundaries. A power law value of 1 is used in this calculation.

Fig. 11 compares calculated profiles of velocity in the axial and transverse directions at four different x locations ($x = -0.6D$, $0.0D$, $0.6D$ and $1.2D$) for three different phase angles with the experimental data of Dütsch et al. [33]. The present simulation shows good agreement with the experimental data, and the results are very similar to those obtained by Yang and Balaras [34]. Fig. 12 compares the predicted time history of the drag coefficient in the axial direction with the results of Dütsch et al. [33]. Good agreement is again observed, indicating that the method accurately predicts the surface shear stress and the pressure on the cylinder.

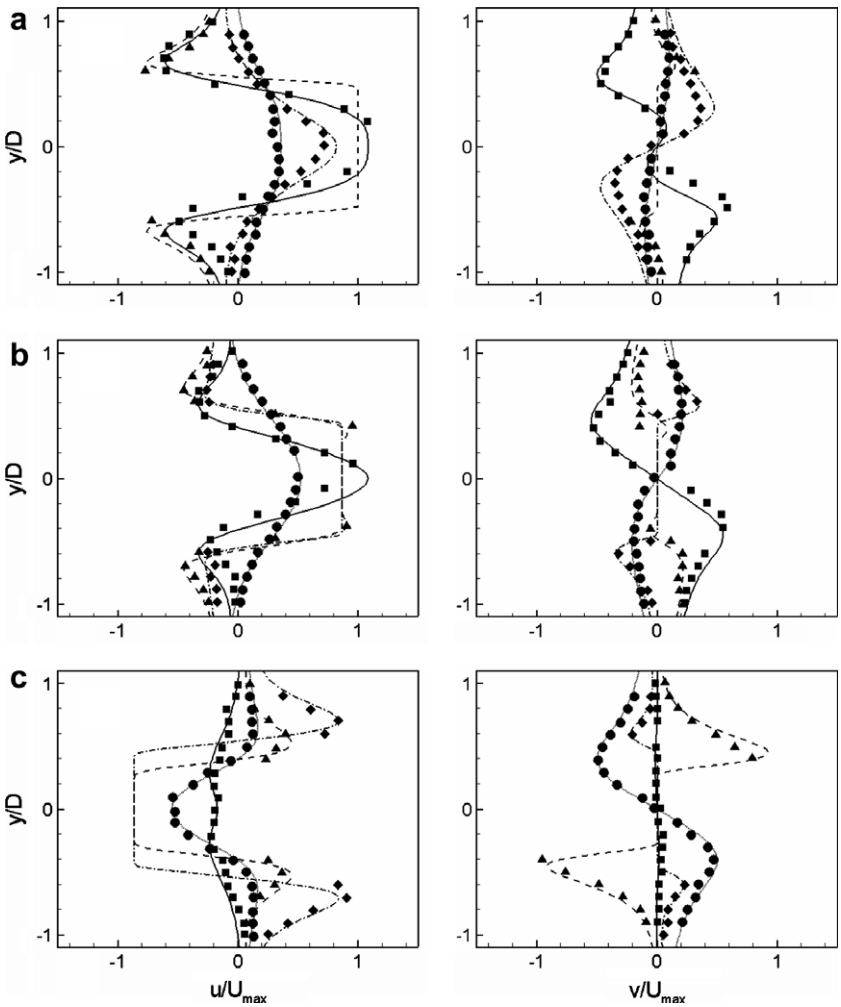

Fig. 11. Velocity profiles $u/U_{\max}$ and $v/U_{\max}$ in oscillatory and transverse directions, respectively, for three different phase angles ($\phi = 2\pi ft$): (a) $\phi = 180°$; (b) $\phi = 210°$; (c) $\phi = 330°$. Lines are the present results; symbols are the experimental data in Dütsch et al. [33]: (--- and ■) at $x = -0.6D$; (--- and ▲) at $x = 0.0D$; (-·- and ♦) at $x = 0.6D$; (··· and ●) at $x = 1.2D$.

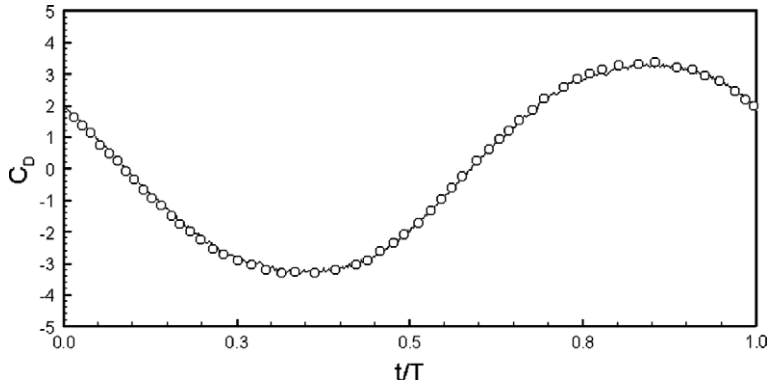

Fig. 12. Time history of the drag coefficient at $Re = 100$ and $KC = 5$: line and symbols represent the present computation and result of Dütsch et al. [33], respectively.

3.3. Flow over an airfoil

To evaluate the present method for a higher Reynolds number flow, we have investigated flow over a NACA 0012 airfoil at $Re = 5 \times 10^5$ and $Re = 10^9$ based on the chord length (l_c). The corresponding freestream velocity is about 70 m/s. The size of the computation domain is $(9l_c, 10l_c) \times (-9l_c, 9l_c)$ and a Cartesian grid system containing 243×177 nodes is used. The grid system is clustered to the leading and trailing edges of the airfoil. Fig. 13 presents surface pressure coefficients for $Re = 5 \times 10^5$ flow over the airfoil at angles of attack of 3° and 7°. Results are compared with inviscid predictions obtained from XFOIL [35], an established computer code for predicting airfoil characteristics. Good agreement with the XFOIL results is indicated. For the $Re = 10^9$ case, the angle of attack (α) is varied from $-19°$ to 0°. The Reynolds number is high enough so as to require modeling of turbulence effects, which transfer momentum from the external flow toward the surface and thus reduce the tendency of the flow to separate at high angles of attack. In the present method, the power-law interpolation of the tangential velocity serves to mimic the energizing effect of a turbulent boundary layer. Fig. 14 presents the effect of varying the power law from 1 (a linear profile as used in the low Reynolds number calculations) to 1/9 (representative of a fully-developed turbulent boundary layer) on velocity magnitude predictions for an angle of attack of $-16°$. For a power of 1 (k in Eq. (18)), the flow detaches from the

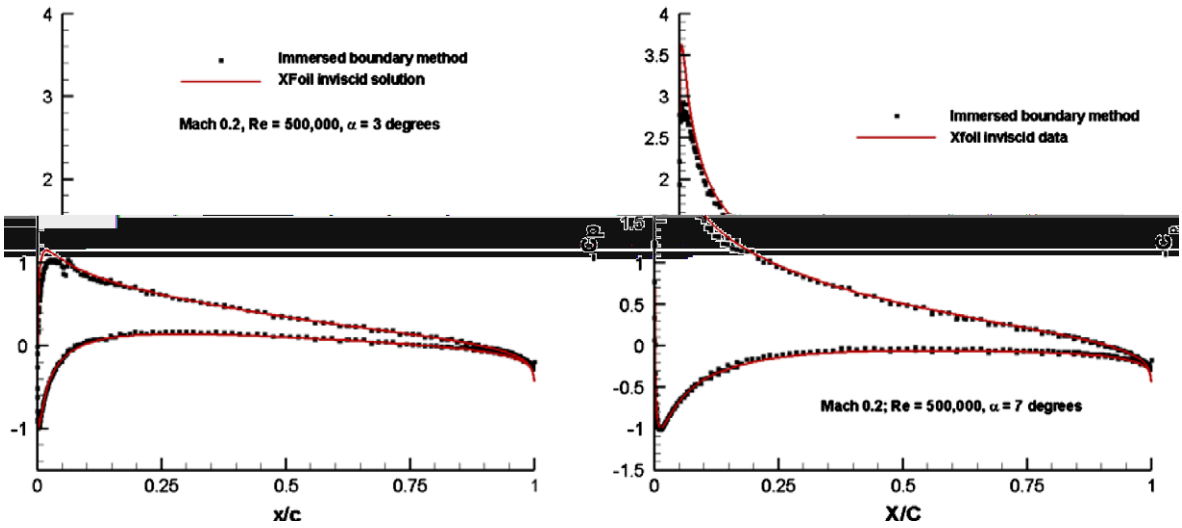

Fig. 13. Comparison of surface pressure predictions with XFOIL inviscid data [35] ($Re = 5 \times 10^5$).

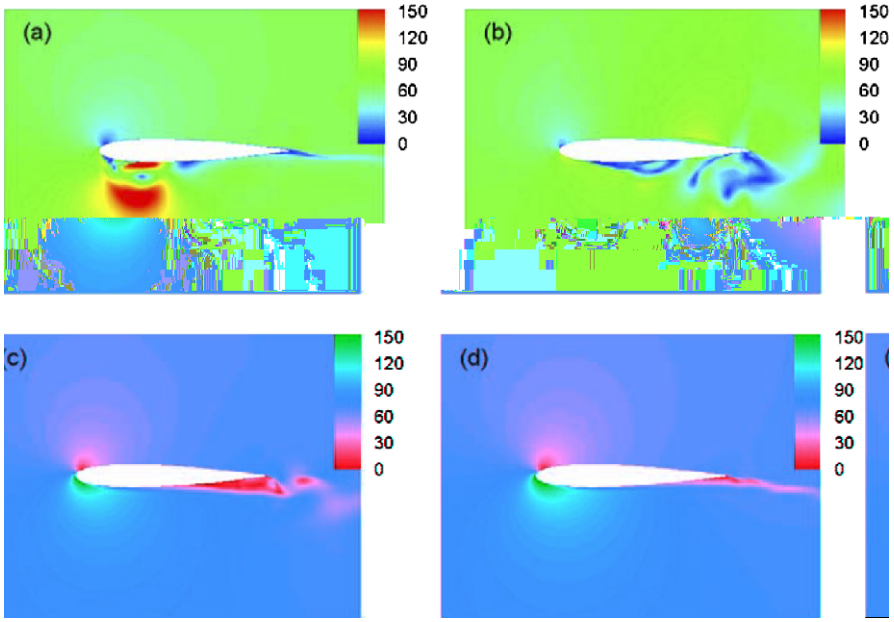

Fig. 14. The effect of power on velocity magnitude for flow over a NACA 0012 airfoil: (a) $k = 1$, (b) $k = 1/3$, (c) $k = 1/7$; and (d) $k = 1/9$.

lower surface the airfoil near the leading edge. As the power k is reduced, the onset of separation moves toward the trailing edge, and the flow field becomes more steady.

The effect of the angle of attack on the lift coefficient is shown in Fig. 15 for a power of 1/7. The lift generated by an airfoil generally increases linearly with increasing angle of attack until a maximum value is reached. An increment of the angle of attack beyond the maximum results in decreasing lift due to the presence of massive flow separation. The present results show good agreement with experimental data [36] and XFOIL analysis [35]. Considering that the grid used in this study does not resolve fine details of the boundary layer and that no turbulence modeling is employed, the level of agreement is encouraging.

3.4. Flow over a sphere

Results for simulations of flow over a sphere are presented in this section. Three different Reynolds numbers ($Re = 100$, 250 and 300) based on the sphere diameter and the free stream velocity are considered since the change in Reynolds number results in three different flow regimes: steady axisymmetric flow ($Re < 200$),

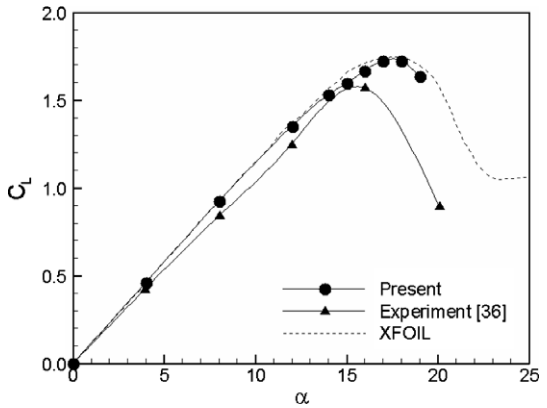

Fig. 15. Variation of lift coefficient (C_L) due to the angle of attack (α) for flow over a NACA 0012 airfoil.

steady asymmetric flow ($210 < Re < 270$) and unsteady flow ($Re > 280$) [37]. The computation domain used is $30D \times 30D \times 30D$, where D is the diameter of sphere. Calculations at $Re = 100$ use a $151 \times 151 \times 151$ mesh, containing $41 \times 41 \times 41$ uniformly-spaced mesh points in the $D \times D \times D$ domain surrounding the sphere. A Dirichlet boundary condition $u/u_\infty = 1$, $v = 0$ is applied at the inflow and far field boundaries, and a subsonic outflow condition is used at the outflow boundary. Calculations at $Re = 250$ and $Re = 300$ were performed on a finer $196 \times 196 \times 196$ mesh with an $81 \times 81 \times 81$ uniformly-spaced mesh in the $D \times D \times D$ domain surrounding the sphere. Twenty-five sub-iterations were used in the unsteady calculations at $Re = 300$.

Table 5 shows our results for the drag and lift coefficients and Strouhal number versus those of several other studies. As shown in Table 5, the present method accurately predicts unsteady characteristics such as shedding frequency and oscillations of C_D and C_L for the $Re = 300$ case, as well as steady drag and lift coefficients for the lower Reynolds numbers. Note that C_D and C_L for $Re = 300$ are time-averaged values. Surface pressure coefficient and wall azimuthal vorticity ω_θ for the axisymmetric $Re = 100$ case are plotted in Fig. 16. Comparison with the results of other researchers indicates that the present immersed boundary method represents the flow near the spherical surface very well. Fig. 17 shows a snapshot of the three-dimensional vortical structure at $Re = 300$. Here, the vortical surfaces are identified using the method of Jeong and Hussain [38].

Table 5
Drag and lift coefficients and Strouhal number for flow over a sphere

	$Re = 100$	$Re = 250$		$Re = 300$		
	C_D	C_D	C_L	C_D	C_L	St
Kim et al. [11]	1.09	0.70	0.059	0.657	0.067	0.134
Johnson and Patel [39]	1.08	0.70	0.062	0.656	0.069	0.137
Constantinescu and Squires [40]	–	0.70	0.062	0.655	0.065	0.136
Present (second-order)	1.09	0.70	0.052	0.658	0.068	0.134

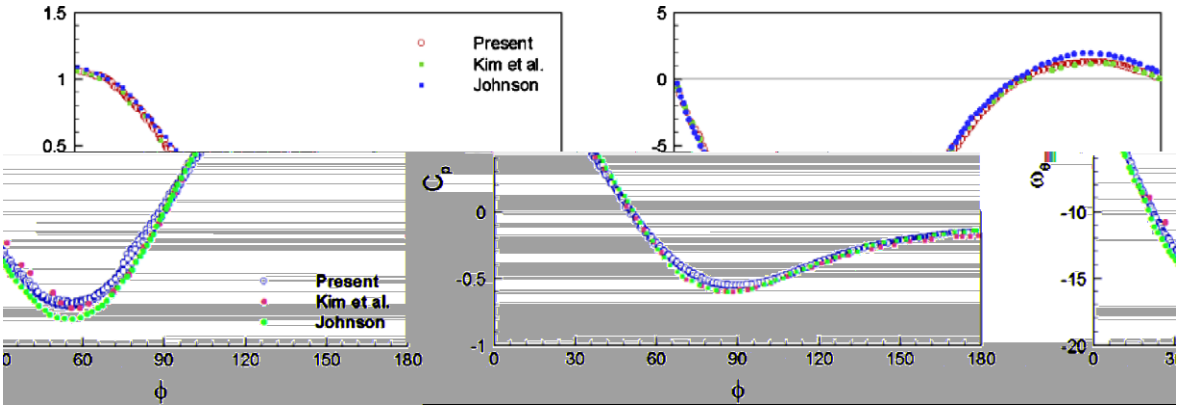

Fig. 16. Flow variables along the sphere surface at $Re = 100$; (a) wall pressure coefficient (C_p) and (b) wall azimuthal vorticity ω_θ.

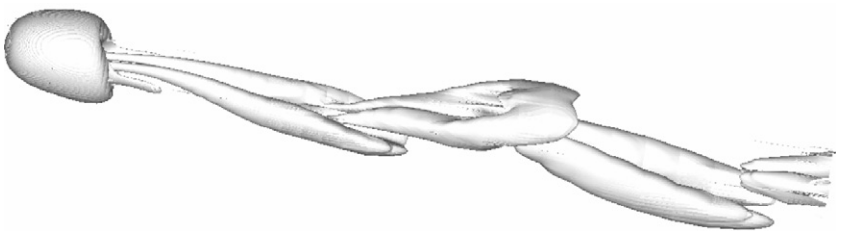

Fig. 17. Instantaneous vortical structure at $Re = 100$.

The flow is unsteady and the vortices are asymmetrically shed. This behavior is nearly the same as shown in Johnson and Patel [39] and indicates that the present immersed boundary method accurately captures the three-dimensional vortex dynamics.

3.5. Flow over a stationary mannequin

To validate the present method in predicting flow dynamics behind a realistic human body, a simulation of an experiment of Kim and Flynn [41] involving flow over a mannequin was conducted. These experiments used smoke-wire flow visualization and hot-wire anemometry to arrive at a characterization of the wake flow behind a stationary mannequin. We obtained a stereolithography (STL) file for a child mannequin similar to that of Kim and Flynn [41] by using a trial version of Mannequin Elite ver.1.2 (NextGen Ergonomics Inc.). Locally refined surface triangles were then added to achieve better resolution of the mannequin. The inside/outside algorithm using the extended surface data set with consensus logic (Section 2.3.2) was used to render the CAD file as a level set in the computational domain. The size of the mannequin was adjusted to approximate that of Kim and Flynn [41]: 1.04 m tall and 0.20 m wide at the chest.

The Kim and Flynn experiments were carried out in a low-speed, open circuit wind tunnel 0.465 m^2 and 2.419 m deep with a velocity range of 0.127–1.397 m/s. To avoid feedback effects, the inflow boundary condition is applied at a location 1.219 m upstream of the wind-tunnel test-section entrance in the experiment. Also to avoid the distortion of flow characteristics near the outflow boundary, a buffer region of length 2.438 m is added to the end of the domain. The computation domain size in units of meters is $(-2.438, 4.879) \times (-0.762, 0.762) \times (0, 1.524)$ for the streamwise, spanwise and normal directions, respectively. The corresponding grid resolution is $181 \times 105 \times 119$. The freestream air velocity is $u_\infty = 0.762$ m/s, and no-slip boundary conditions are enforced on the wind-tunnel walls. The computational time step is chosen as $\Delta t = 0.015$ s to maintain a CFL number less than one. The calculation is performed until $t = 420$ s. Two different powers ($k = 1/7$ and 1) for the tangential velocity interpolation are tested to assess the effect of the power law in the prediction of flow separation and reattachment. The Reynolds number based on the chest width is $Re = 22053$.

After an initial transient region of 60 s, statistics (based on 4000 samples) were taken over the next 360 s. Convergence to a statistically steady state was determined by monitoring the mass balance between inflow and outflow planes and the time history of the velocity field. Fig. 18 shows iso-surfaces of time-averaged axial velocity. The structure of the upper separation zone is very similar to the scanned figure from Kim and Flynn [41] (Fig. 18(a)). In general, the width and length of the separation zone is proportional to the width of the mannequin at each level. However, in the upper chest region, the present results show that the separation is delayed and the corresponding width of the separation zone is almost the same as the width of the neck for both $k = 1$ and $k = 1/7$. A grooved region in the upper separation zone is observed near the junction between the arm and chest region. This is because the high shear flows at the junction delay separation and then enhance the axial momentum transfer. Two flow separation zones in the present simulation are

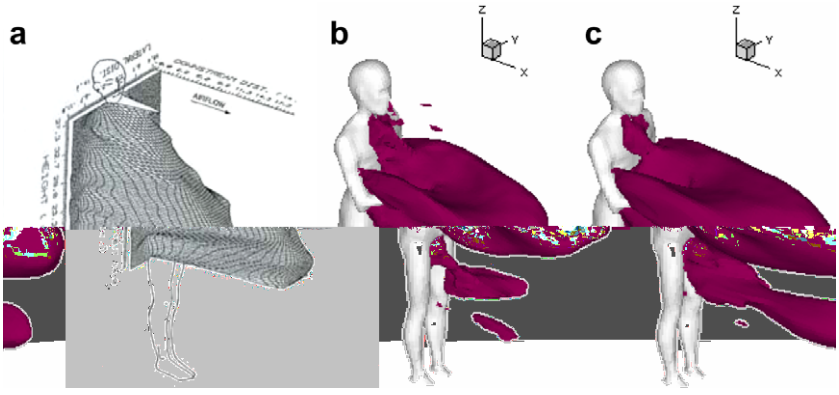

Fig. 18. Iso-surfaces of zero axial velocity. (a) Experiment [41], (b) power $k = 1/7$ and (c) power $k = 1$.

observed, in accord with the description of the flow structure provided in Kim and Flynn [41] but not with their isosurface data. Apparently, smoke-wire images in planes below the waist were not mapped in their experiment. The lower separation zone is driven by the high shear flow between the legs of the mannequin. Fig. 18(b) and (c) show that the maximum axial displacement of the upper separation zone occurs at a plane near the belly button, which is consistent with the experimental data.

Fig. 19 shows contour plots of time-averaged axial velocity in the different planes for the two power laws. As shown in Fig. 19(a) (left and right) two regions of flow separation are observed. In the upper region, the flow separates near the chin and then attaches near the belly button. In the lower region, the flow separates near the hip and then attaches near the thigh. The separation zones for a power law of 1 are more elongated in the axial direction than for a power law of 1/7. Figs. 19(b)–(d) show the recirculation regions in a lateral plane. The mean averaged flow fields are not bilaterally symmetric, as the arms are not placed symmetrically. It is worthy to note that the contour plots for a power law of 1 show earlier separation than for a power law 1/7. This implies that the use of the lower power law energizes the near-surface velocity field, delaying its departure from the body. The predicted sizes of the separation zones are summarized in Table 6.

Fig. 20 shows contour plots of time-averaged turbulence kinetic energy in the different planes for the two power laws. High levels of turbulence energy are observed near the boundary of the two separation zones for a power of 1 while a more well-distributed range of turbulence energy levels is observed in that region for a power of 1/7. It may be conjectured that large scale vortices due to the earlier separation for a power of 1 enhance turbulence energy transfer in that region. Especially high turbulence kinetic energy levels are found

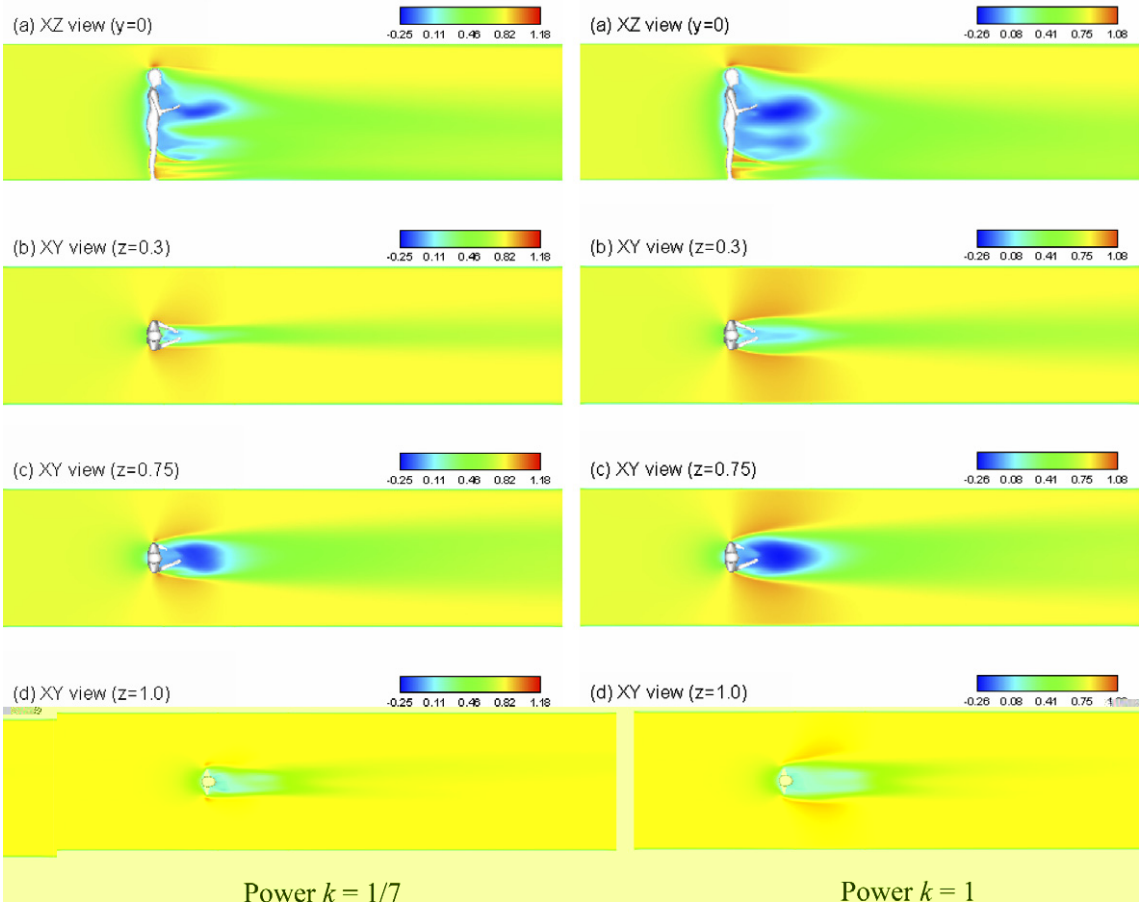

Fig. 19. Contour plots of time averaged axial velocity for power $k = 1/7$ (left) and $k = 1$ (right). (a) Center ($y = 0$, xz plane view). (b) Thigh region ($z = 0.3$, xy-plane view). (c) Hip region, $z = 0.5$ and (d) chest region, $z = 1.0$.

Table 6
Estimated sizes of the separation zone (m)

	Power, $k = 1/7$	Power, $k = 1$	Experiment [41]
Upper			
Length	0.610	0.762	0.559
Width	0.305	0.356	0.330
Lower			
Length	0.178	0.584	–
Width	0.025	0.076	–

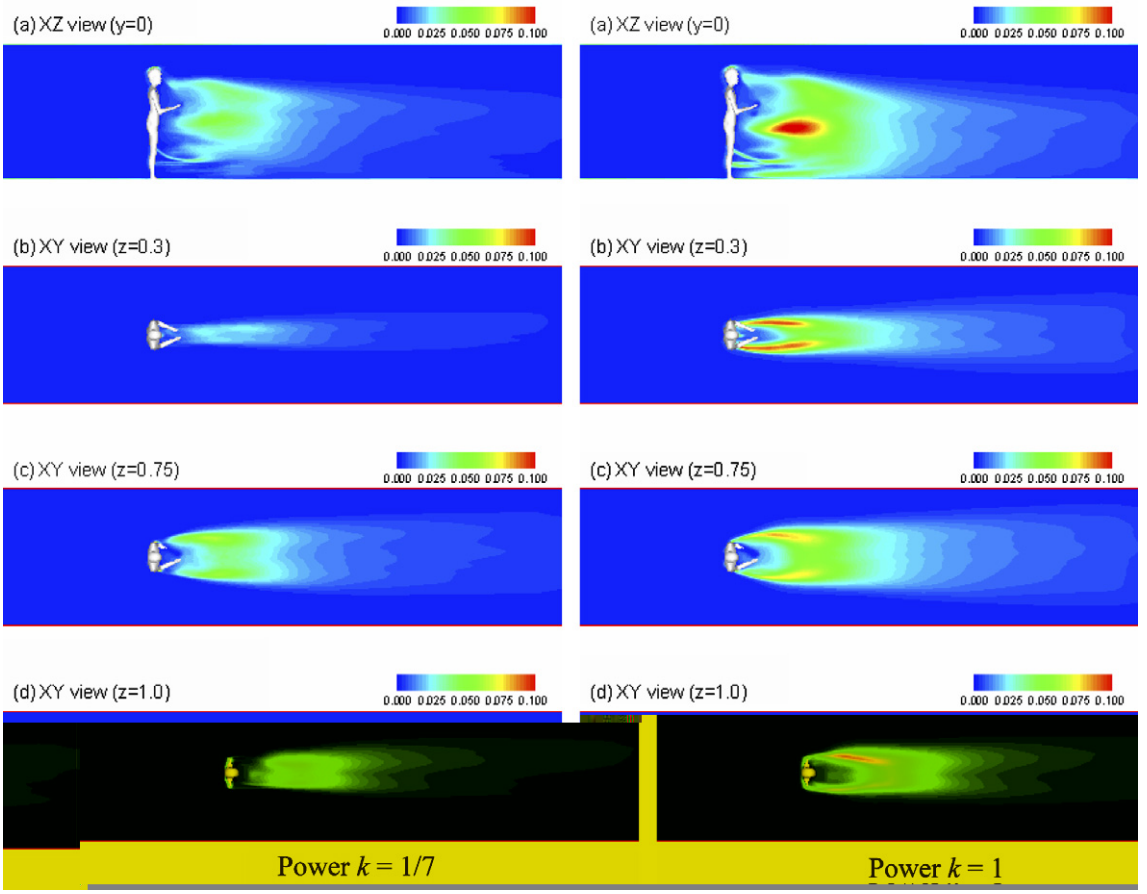

Fig. 20. Contour plots of time averaged kinetic energy for power $k = 1/7$ (left) and $k = 1$ (right). (a) Center ($y = 0$, xz plane view). (b) Thigh region ($z = 0.3$, xy-plane view). (c) Hip region, $z = 0.5$ and (d) chest region, $z = 1.0$.

near the thigh region for a power of $k = 1$. This strong turbulence energy may enlarge the wake region in the lateral direction, compared to that for a power of $k = 1/7$. Interestingly, the turbulence kinetic energy is very small near the chest region. Particles or contaminant gases trapped in this region may thus have a poor dispersion or mixing rate.

The unsteady characteristics of flow behind the mannequin are investigated by determining vortex-shedding frequencies. The sensors are located at seven different heights representing the levels of the nose, neck, chest, elbows, waist, hip and thigh (i.e., 0.356, 0.457, 0.559, 0.660, 0.762, 0.864 and 0.965 m from the wall, respectively). The sensor array is located 0.914 m downstream from the mannequin and 0.203 m away from the center axis. Shedding frequencies are analyzed from $t = 60$ to $t = 420$ s. In order to find a dominant

Table 7
Estimated shedding frequency (Hz)

	Present (power $k = 1/7$)		Experiment [41]
	Left location	Right location	
Hip	0.891	1.669	0.791
Waist	0.154/0.986	0.871/2.51	0.703
Elbows	0.179	0.316	0.689
Chest	0.506	0.995	0.908

shedding frequency, a spectral analysis was performed. We utilized an overlapped time window technique in applying the fast Fourier transformation (FFT) in order to enlarge the ensemble sample size. In the present study, we used $N = 2048$ sample data points and 12 ensemble data sets. The sampling time is $\Delta t = 0.015$ s. The corresponding Nyquist frequency is 33.33 Hz. The power spectrum for power $k = 1$ shows no predominant frequency, while the spectrum for power $k = 1/7$ shows dominant frequencies at each different location but these frequencies are not strongly dominant. This is because the flow depends on the various characteristic length scales in each region. Table 7 shows the estimated dominant frequencies at the different locations. The dominant frequencies in the present study are fairly good agreement with those in the experiment even though the posture of the mannequin in the present simulations is not quite the same as in the experiment.

3.6. Flow induced by human motion

Contaminant transport as induced by human motion and other factors within a workplace environment is important in determining evacuation and cleanup procedures. Results presented in this section correspond to realistic simulations of human walking motion using the present immersed-boundary method. The adult human rendering discussed in Section 2.3.2 is used.

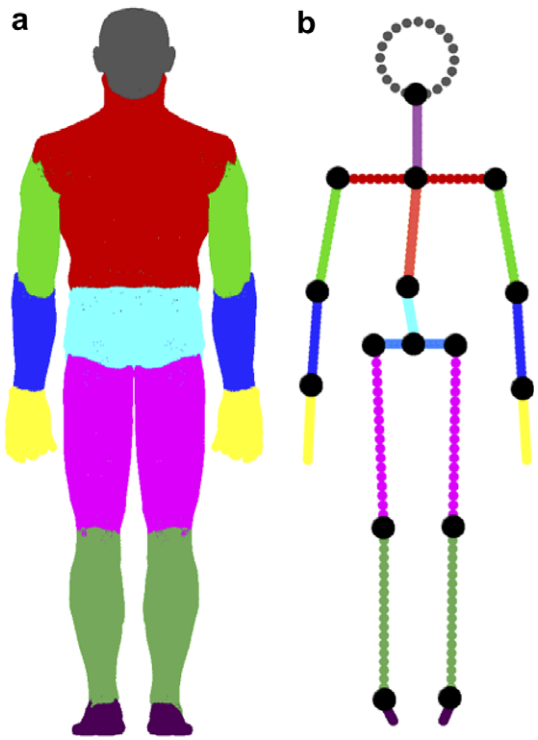

Fig. 21. Front views of: (a) human rendering model (15 segments) and (b) skeleton model (black circles represent joint positions).

3.6.1. Human locomotion

The biomechanics of human locomotion is extremely complex since the dynamics of bones and muscles are mutually correlated. A simple but popular model is a skeleton model (*kinematic chain*), consisting of rigid links connected by joints [42]. Based on the present mannequin model, we construct a skeleton model, which has eighteen rigid links and sixteen joints as shown in Fig. 21. The corresponding joints are shown as black circles. The positions of each joint are determined as the center positions of the overlapped regions between the two segments. These joints are used in a reference position for rotating each segment and in constraints for connecting each segment.

A complete kinematic model of a human body calculates forces and torques at the joints based on the contact forces. In this study, however, we focus on movement of multiple joints and corresponding body segments. At a given joint angle progression within a movement cycle, we define the joint locations, rotation axes, and relative motion, each of which depends on the movement of each segment. For movement of the body segments within an advection time step Δt, the kinematic relation between the initial position vector $\mathbf{x}_1^k$ and the final position vectors $\mathbf{x}_2^k$ for the kth segment can be written as

$$\mathbf{x}_2^k = \mathbf{A}(\Delta\theta^k)(\mathbf{x}_1^k - \mathbf{x}_r^k) + \mathbf{x}_r^k + \mathbf{x}_J^k + \mathbf{x}_P^k. \tag{34}$$

Here, $\boldsymbol{A}$ is the transformation matrix, $\Delta\theta^k = \theta_2^k - \theta_1^k$ is the joint angle movement vector, $\mathbf{x}_r^k$ is a reference position vector, $\mathbf{x}_J^k$ is a constitutive joint movement vector and $\mathbf{x}_P^k$ is a translation vector due to the motion of the entire system. A three-dimensional rotating transformation matrix can be decomposed in a Cartesian coordinate system as

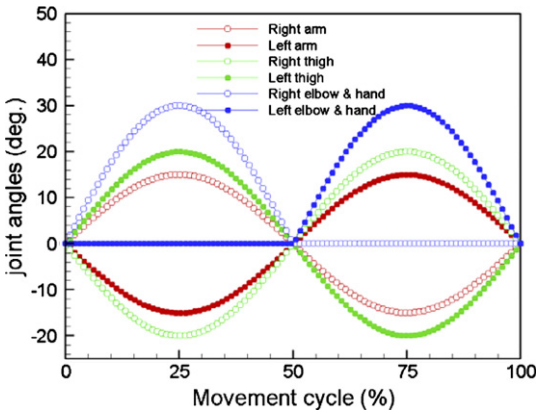

Fig. 22. Progression of joint angles within a movement cycle.

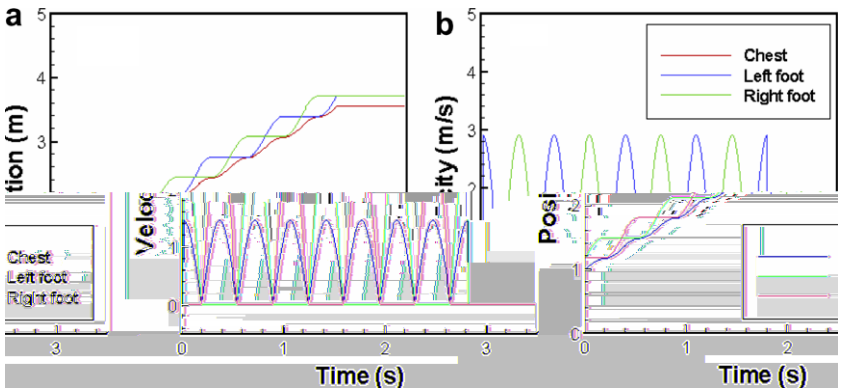

Fig. 23. Kinematics of walking motion: (a) position and (b) velocity at chest, left foot and right foot.

$$\mathbf{A} = \mathbf{A_x}(\varphi_x)\mathbf{A_y}(\varphi_y)\mathbf{A_z}(\varphi_z), \tag{35}$$

where φ_x, φ_y and φ_z are rotation angles in a Cartesian coordinate system. The transformation matrix can be decomposed in each direction as

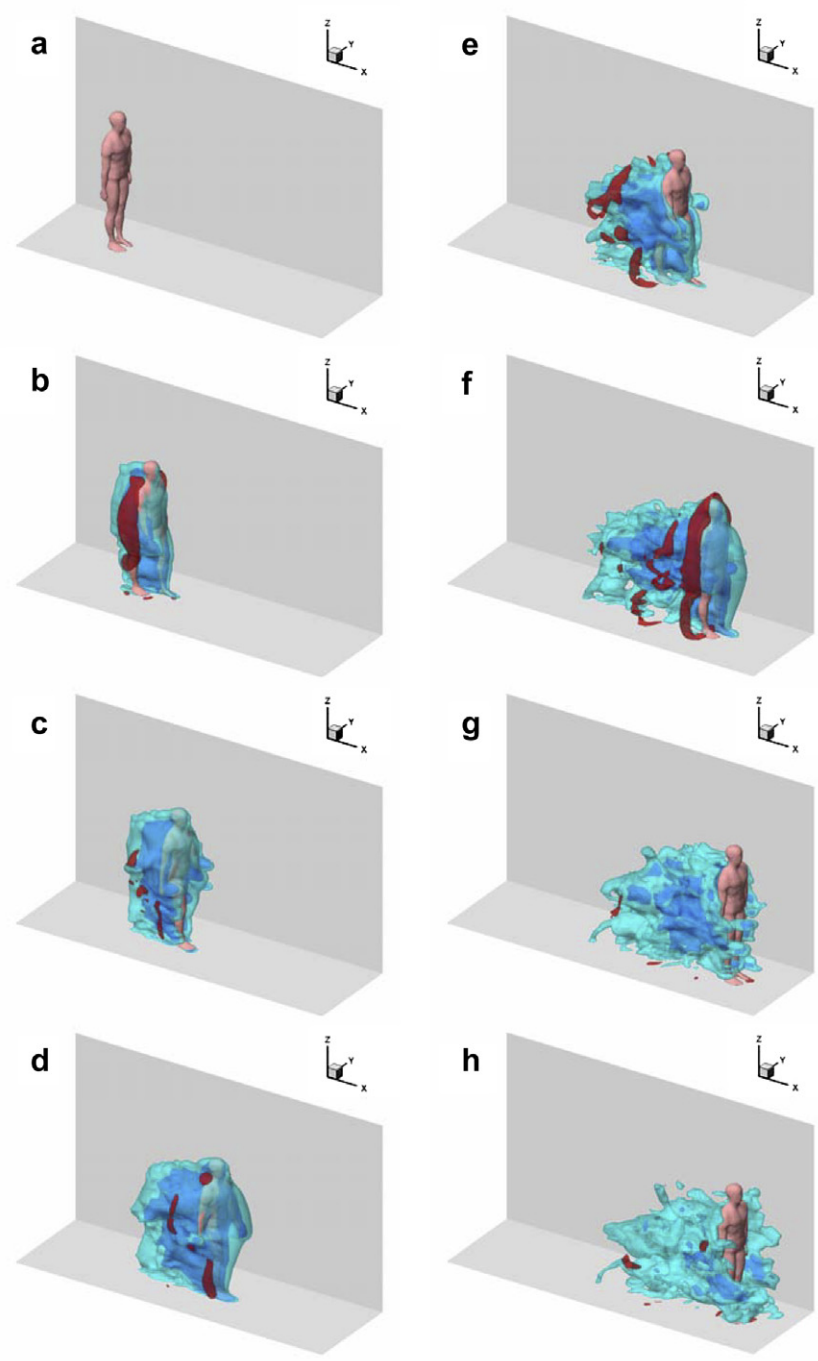

Fig. 24. Evolution of iso-surfaces of streamwise velocity at each time instant: (a) $t = 0$, (b) $t = 0.5$, (c) $t = 1.0$, (d) $t = 1.5$, (e) $t = 2.0$, (f) $t = 2.5$, (g) $t = 3.0$, and (h) $t = 3.5$ s.

$$\mathbf{A_x} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi_x & -\sin\varphi_x \\ 0 & \sin\varphi_x & \cos\varphi_x \end{bmatrix}, \quad \mathbf{A_y} = \begin{bmatrix} \cos\varphi_y & 0 & -\sin\varphi_y \\ 0 & 1 & 0 \\ \sin\varphi_y & 0 & \cos\varphi_y \end{bmatrix}, \quad \mathbf{A_z} = \begin{bmatrix} \cos\varphi_z & -\sin\varphi_z & 0 \\ \sin\varphi_z & \cos\varphi_z & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{36}$$

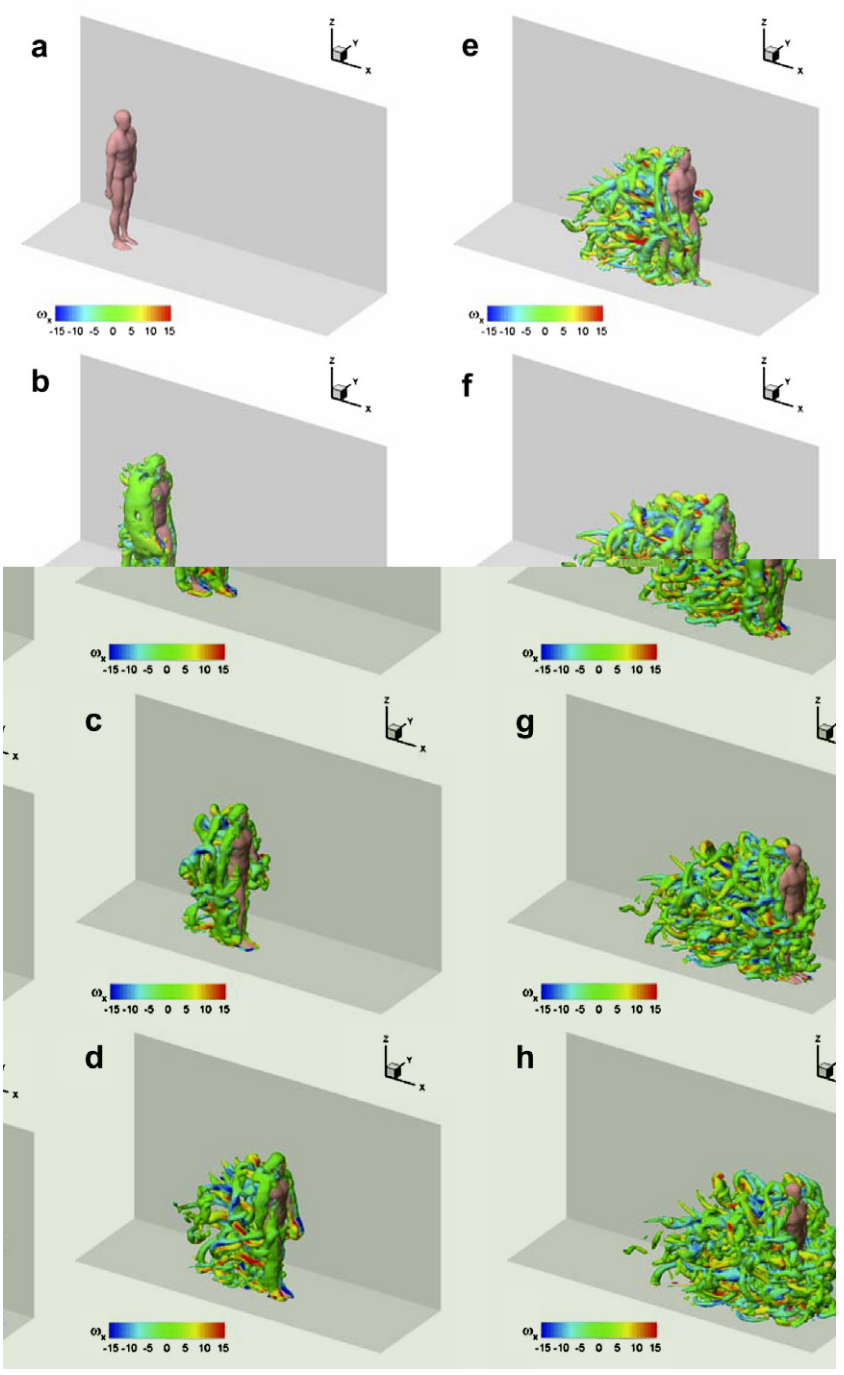

Fig. 25. Evolution of coherent vortical structures induced human walking motion at each time instant: (a) $t = 0$, (b) $t = 0.5$, (c) $t = 1.0$, (d) $t = 1.5$, (e) $t = 2.0$, (f) $t = 2.5$, (g) $t = 3.0$ and (h) $t = 3.5$ s. Contours show the magnitude of the streamwise vorticity.

When the body is moved, the outward normal vectors at the immersed surface have to be adjusted to the rotated coordinate system. The corresponding outward normal vector at the final posture can be written as

$$\mathbf{n}_2^k = \mathbf{A}(\Delta\theta^k)\mathbf{n}_1^k, \tag{37}$$

where $\mathbf{n}_1^k$ is the outward normal vector at the initial posture. The velocity $\mathbf{u}_S$ at the immersed surfaces can be simply defined as,

$$\mathbf{u}_S = (\mathbf{x}_2 - \mathbf{x}_1)/\Delta t. \tag{38}$$

To mimic human walking, we design the progression of joint angles within a movement period as shown in Fig. 22. The period of walking motion is set to 0.7 s. While thighs, arms, elbows and hands are moved, the other segments keep their initial posture during walking. Maximum joint angle displacements for thighs, arms, elbows and hands are 20°, 15°, 30° and 30°, respectively (see Fig. 23).

3.6.2. Wake flows

As a test case, we consider human walking motion in a narrow hallway. The computational domain size in unit of meters is $(0, 4.22) \times (0, 1.27) \times (0, 2.44)$ for the streamwise, transverse and normal directions, respectively. A mesh containing $257 \times 65 \times 109$ nodes is used in the calculation. The freestream velocity is initially set to $u_\infty = 0.0001$ m/s (nearly zero since the flow will be induced by the body motion). The computational time step is chosen as $\Delta t = 0.01$ s. Outflow conditions are used in both the inlet and outlet regions in the streamwise direction while no-slip conditions are applied to the other boundaries. The human model is initially located 1 m upstream of the hall entrance and is centered within the hall. We used the same joint angle progressions for walking motion as shown in Fig. 22. However, a pivot position is introduced in order to walk forward. For example, the left foot acts as pivot when the right foot is moving forward. Fig. 24 shows the position and velocity at the chest, right foot and left foot during walking motion. Walking is started at $t = 0.02$ s and is stopped at $t = 2.82$ s. The average walking velocity in the simulation is 0.907 m/s, which is lower than the average human walking speed (1.29 m/s) [43].

Figs. 24 and 25 provide details of the unsteady flow structure in the vicinity of the human over 3.5 s of activity. Figs. 24 shows iso-surfaces of streamwise velocity ($u = -0.5$, 0.5 and 1.0 m/s) during the walking event. The time interval between snapshots is 0.5 s. When the human starts to walk, a region of negative velocity is formed behind it. When the human stops walking, the wake behind human continues to move upstream due to fluid inertia. Fig. 25 shows the evolution of vortical structures defined by the vortex identification method [38]. The coloring of the iso-surfaces represents the magnitude of the streamwise vorticity. During the walking motion, vortices are shed from body surfaces. Especially strong vortical structures are observed near the hands and legs, which locally move more rapidly than the average walking speed. The vortical structures elongate as the human accelerates but begin to dissipate after walking ceases.

4. Conclusions

A new immersed boundary method for incompressible flow simulations is proposed in this work. Immersed-boundary objects are rendered as level sets in the computational domain, and the decomposition of the flow field into cells within, near, and outside the body is performed using a Heaviside function based on a signed distance function. A novel decision procedure using a consensus algorithm is developed in order to enhance the efficiency and accuracy of the classification of points as being *inside* or *outside* an immersed body. The velocity field near immersed bodies is determined from separate interpolations of the components tangent and normal to the immersed surface. The tangential velocity near the surface is constructed as a power-law function of the local wall normal distance. Appropriate choices of the power law enable the method to approximate the energizing effects of a turbulent boundary layer for higher Reynolds number flows. Laminar flows past a circular cylinder and sphere are computed in order to validate the present method. The results show good predictions of the near-wall velocity, pressure fields and unsteady characteristics for the different flow regimes. Results for high Reynolds number flow over a NACA 0012 airfoil show good agreement with experimental data and XFOIL analysis, even though the turbulent boundary layers are not resolved. A large eddy simulation has been performed for flow over a mannequin. Two distinct separation zones are observed,

and the use of a power-law of 1/7 is shown to result in good agreement for the size and shape of the recirculation zone located in front of the mannequin. Finally, the technique is used to simulate the flow induced by realistic human walking motion. This case illustrates the ability of the procedure to handle motion induced by immersed objects with multiple moving components.

Acknowledgements

This work has been supported by the US Environmental Protection Agency under contract 4C-R138-NAEX and by the Defense Advanced Research Projects Agency (DARPA) under Grant HR0011-04-1-0057. Computer resources were provided by the High Performance Computing component of North Carolina State University's Information Technology Division (www.ncsu.edu/itd/hpc). The first author is grateful to Prof. HyungSeok Kim of Konkuk University and Dr. Xudong Xiao of North Carolina State University for their helpful suggestions regarding the surface definition of CAD-based immersed objects.

References

[1] R. Mittal, G. Iaccarino, Immersed boundary methods, Annual Review of Fluid Mechanics 37 (2005) 239–261.
[2] D. De Zeeuw, K. Powell, An adaptively refined Cartesian mesh solver for the Euler equations, Journal of Computational Physics 104 (1993) 56–68.
[3] J.J. Quirk, An alternative to unstructured grids for computing gas dynamic flows around arbitrary complex two-dimensional bodies, Computers and Fluids 23 (1994) 125–142.
[4] H.S. Udaykumar, W. Shyy, M.M. Rao, Elafint: a mixed Eulerian–Lagrangian method for fluid flows with complex and moving boundaries, International Journal for Numerical Methods in Fluids 22 (1996) 691–705.
[5] T. Ye, R. Mittal, H.S. Udaykumar, W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, Journal of Computational Physics 156 (1999) 209–240.
[6] H.S. Udaykumar, R. Mittal, P. Rampuggoon, A. Khanna, A sharp interface Cartesian grid method for simulating flows with complex moving boundaries, Journal of Computational Physics 174 (2001) 345–380.
[7] C.S. Peskin, Flow patterns around heart valves: a numerical method, Journal of Computational Physics 10 (1972) 220–252.
[8] D. Goldstein, R. Handler, L. Sirovich, Modeling a no-slip flow boundary with an external force field, Journal of Computational Physics 105 (1993) 354–366.
[9] J. Mohd-Yusof, Combined immersed boundaries/B-splines methods for simulations of flows in complex geometries. CTR Annual Research Briefs, NASA Ames/Stanford University (1997) 317–327.
[10] E.A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary/finite-difference methods for three-dimensional complex flow simulations, Journal of Computational Physics 161 (2000) 35–60.
[11] J. Kim, D. Kim, H. Choi, An immersed boundary finite volume method for simulations of flow in complex geometries, Journal of Computational Physics 171 (2001) 132–150.
[12] A. Gilmanov, F. Sotiropoulus, E. Balaras, A general reconstruction algorithm for simulating flows with complex 3-d immersed boundaries on Cartesian grids, Journal of Computational Physics 191 (2003) 660–669.
[13] E. Balaras, Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations, Computers and Fluids 33 (2004) 375–404.
[14] A. Gilmanov, F. Sotiropoulos, A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies, Journal of Computational Physics 207 (2005) 457–492.
[15] A.J. Chorin, A numerical method for solving incompressible Navier–Stokes equations, Journal of Computational Physics 2 (1967) 12–26.
[16] P. Wesseling, Introduction to Multigrid Methods, NASA CR-195045, 1995.
[17] J.R. Edwards, M.-S. Liou, Low-diffusion flux-splitting methods for flows at all speeds, AIAA Journal 36 (1998) 1610–1617.
[18] M.D. Neaves, J.R. Edwards, All-speed time-accurate underwater projectile calculations using a preconditioning algorithm, ASME Journal of Fluids Engineering 128 (2006) 284–296.
[19] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, Journal of Computational Physics 126 (1996) 202–228.
[20] R.A. Baurle, C.J. Tam, J.R. Edwards, H.A. Hassan, Hybrid simulation approach for cavity flows: blending, algorithm, and boundary treatment issues, AIAA Journal 41 (2003) 1463–1480.
[21] G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, SIAM Journal of Scientific Computing 20 (1998) 359–392.
[22] S. Arya, D.M. Mount, N.S. Netanyahu, R. Silverman, A.Y. Wu, An optimal algorithm for approximate nearest-neighbor searching, Journal of the ACM 45 (1998) 891–923.
[23] J.A. Bærentzen, H. Aanæs, Signed distance computation using the angle weighted pseudonormal, IEEE Transactions of Visualization and Computer Graphics 11 (2005) 243–253.
[24] R. Sagawa, K. Ikeuchi, Taking consensus of signed distance field for complementing unobservable surface, in: Proceeding of Fourth International Conference on 3-D Digital Imaging and Modeling, 2003, pp. 410–417.

[25] B. Fornberg, A numerical study of steady viscous-flow past a circular-cylinder, Journal of Fluid Mechanics 98 (1980) 819–855.
[26] S.C.R. Dennis, G.-Z. Chang, Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100, Journal of Fluid Mechanics 42 (1970) 471–489.
[27] D. Calhoun, A Cartesian grid method for solving the two-dimensional stream function-vorticity equations in irregular region, Journal of Computational Physics 176 (2002) 231–275.
[28] D. Russell, Z.J. Wang, A Cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow, Journal of Computational Physics 191 (2003) 177–205.
[29] M. Rosenfeld, D. Kwak, M. Vinokur, A fractional step solution method for the unsteady incompressible Navier–Stokes equations in generalized coordinate systems, Journal of Computational Physics 94 (1991) 102–137.
[30] J.A. Wright, R.W. Smith, An edge-based method for the incompressible Navier–Stokes equations on polygonal meshes, Journal of Computation Physics 169 (2001) 24–43.
[31] M. Braza, P. Chassaing, H.H. Minh, Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder, Journal of Fluid Mechanics 165 (1986) 79–130.
[32] C. Liu, X. Sheng, C.H. Sung, Preconditioned multigrid methods for unsteady incompressible flows, Journal of Computational Physics 139 (1998) 35–57.
[33] H. Dütsch, F. Durst, S. Becker, H. Lienhart, Low-Reynolds-number flow around an oscillating circular cylinder at low Keulegan–Carpenter numbers, Journal of Fluid Mechanics 360 (1998) 249–271.
[34] J. Yang, E. Balaras, An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries, Journal of Computational Physics 315 (2006) 12–40.
[35] M. Drela, XFOIL: an analysis and design system for low Reynolds number airfoils, in: T.J. Mueller (Ed.), Lecture Notes in Engineering, vol. 54, Springer Verlag, 1989.
[36] I.H. Abbott, A.E. Von Deonhoff, Theory of Wing Sections, Dover Publications Inc., 1959, pp. 462–463.
[37] T.A. Johnson, Numerical and experimental investigation of flow past a sphere up to a Reynolds number of 300, Ph.D., dissertation, University of Iowa, 1996.
[38] J. Jeong, F. Hussain, On the identification of a vortex, Journal of Fluid Mechanics 285 (1995) 59–94.
[39] T.A. Johnson, V.C. Patel, Flow past a sphere up to a Reynolds number of 300, Journal of Fluid Mechanics 378 (1999) 19–70.
[40] G.S. Constantinescu, K.D. Squires, LES and DES investigations of turbulent flow over a sphere, AIAA Paper 2000-0540, 2000, pp. 1–11.
[41] T. Kim, M.R. Flynn, Airflow pattern around a worker in a uniform freestream, American Industrial Hygiene Association Journal 52 (1991) 287–296.
[42] V.M. Zatsiorsky, Kinetics of Human Motion, Human Kinetics, 2002.
[43] R.W. Selles, J.B.J. Bussmann, R.C. Wagenaar, H.J. Stam, Comparing predictive validity of four ballistic swing phase models of human walking, Journal of Biomechanics 34 (2001) 1171–1177.