Proceedings of the ASME 2010 3rd Joint US-European Fluids Engineering Summer Meeting and 8th International Conference on Nanochannels, Microchannels, and Minichannels
FEDSM-ICNMM2010
August 1-5, 2010, Montreal, Canada

FEDSM-ICNMM2010-30529

IMMERSED BOUNDARY METHOD FOR GENERALISED FINITE VOLUME AND FINITE DIFFERENCE NAVIER-STOKES SOLVERS

A. Pinelli
CIEMAT
Unidad de Modelización y Simulación Numérica
Madrid
Spain
a.pinelli@ciemat.es

I. Z. Naqavi[*]
U. Piomelli
Department of Mechanical and Materials Engineering
Queen's University
Kingston, Ontario, K7L 3N6
Canada
naqavi@appsci.queensu.ca, ugo@me.queensu.ca

ABSTRACT

In Immersed Boundary Methods (IBM) the effect of complex geometries is introduced through the forces added in the Navier-Stokes solver at the grid points in the vicinity of the immersed boundaries. Most of the methods in the literature have been used with Cartesian grids. Moreover many of the methods developed in the literature do not satisfy some basic conservation properties (the conservation of torque, for instance) on non-uniform meshes. In this paper we will follow the RKPM method originated by Liu et al. [1] to build locally regularized functions that verify a number of integral conditions. These local approximants will be used both for interpolating the velocity field and for spreading the singular force field in the framework of a pressure correction scheme for the incompressible Navier- Stokes equations. We will also demonstrate the robustness and effectiveness of the scheme through various examples.

INTRODUCTION

In the last two decades several authors have turned their attention to the immersed boundary methods (IBMs) for their ability to handle moving or deforming bodies with complex surface geometry embedded in a fluid flow. Peskin [2] first introduced the method by solving the flow motion over a uniform mesh (from now on termed the *Eulerian* grid) while representing the immersed surface with a set of Lagrangian markers. The key feature of IBMs is the fact that the Eulerian grid is not required to conform to the immersed body geometry, since the no-slip boundary conditions are enforced at the Lagrangian points by introducing appropriate boundary forces. The boundary forces that exist as singular functions along the surface in the continuous equations are described by mollified delta functions that spread (regularize) the forcing field over the neighboring Eulerian cells.

The original forcing scheme was designed for flexible boundaries and had some drawbacks for rigid boundaries in form of sever restriction on the time step size. To avoid this restriction Fadlun *et al.* [3] introduced a direct formulation of the force term. The basic idea consists of modifying the entries of the implicit matrix of the discretized momentum equation such that the interpolated velocity at the Lagrangian points takes on the desired values. Fadlun *et al.* [3] present an example of a flow involving moving boundaries; however, the lack of smoothness of the boundary force during the relative motion was recognized by Uhlmann [4]. He presented an alternative direct-forcing scheme where the force is computed on the Lagrangian markers and later on spread onto the neighboring Eulerian nodes. The algorithm conveys very smooth hydrodynamic forces while preserving the global order of the spatial scheme. Recently, Vanella and Balaras [5] have presented an extension of the method that, while keeping the simplicity of the original technique, leads to sharp behavior similar to Eulerian direct-forcing schemes and bound-

[*]Address all correspondence to this author.

ary conforming methods.

Fewer applications of immersed-boundary methods to curvilinear grids have been developed [6, 7]. The common argument against using immersed-boundary methods with curvilinear grids is the increased cost of curvilinear codes, compared to Cartesian ones. However, when clustering of grid points is needed close to solid boundaries, Cartesian grids may be suboptimal, as the refinement is extended to regions of the flow in which it is not needed. Furthermore, in wall-bounded turbulent flows it is generally desirable to use grid cells that are longer in the flow direction than in the other two; the use of meshes in which the grid lines are nearly aligned with the streamlines can lead to very significant savings. It is our goal, therefore, to develop a method that can be applied on Cartesian (uniform or non-uniform) as well as curvilinear meshes.

One of the reasons that has limited the use of the IBM techniques to Cartesian grids is related with the use of discrete delta functions, which cannot be trivially extended to more complex grid systems without losing some fundamental conservation properties of the regularized force field. In particular, consider the integral conservation of the spread force-field and of the first k moments generated by the singular force distribution:

$$\sum_{i,j,k} \mathbf{f}(\mathbf{x}_{i,j,k})\,\Delta v_{i,j,k} = \sum_m \mathbf{F}(\mathbf{X}_m)\,\Delta V_m, \tag{1}$$

$$\sum_{i,j,k} \mathbf{x}^k_{i,j,k} \times \mathbf{f}(\mathbf{x}_{i,j,k})\,\Delta v_{i,j,k} = \sum_m \mathbf{X}^k_m \times \mathbf{F}(\mathbf{X}_m)\,\Delta V_m. \tag{2}$$

In equations (1) and (2) the sum on the left-hand-side is performed on all the Eulerian grid nodes $\mathbf{x}_{i,j,k}$ (over which the force has been spread), while the one on the right is over all the Lagrangian markers located at $\mathbf{X}_m$. The sums are the discrete counterparts of three-dimensional integrals, $\Delta v_{i,j,k}$ being the volume of the (i,j,k) Eulerian cell and ΔV_m the volume defined about the m-th Lagrangian marker. The identities (1) and (2) are verified if the regularized delta function $\delta_h(\mathbf{s})$ verifies [8]:

$$\sum_{i,j,k} \delta_h(\mathbf{x}_{i,j,k} - \mathbf{X}))\,\Delta v_{i,j,k} = 1 \tag{3}$$

$$\sum_{i,j,k} (\mathbf{x}_{i,j,k} - \mathbf{X})) \cdot \delta_h(\mathbf{x}_{i,j,k} - \mathbf{X}))\,\Delta v_{i,j,k} = 0. \tag{4}$$

While conditions (3) and (4) are easily met on an uniform Cartesian grid by a number of regularized delta functions available in the literature, some correction terms must be introduced in the delta approximant if reproducing-conditions (3) and (4) have to be satisfied on any other underlying mesh.

In this paper we will follow the ideas originated by Liu *et al.* [1] to build locally regularized functions that verify a number of integral conditions. These local approximants will be used both for interpolating the velocity field and for spreading the singular force field in the framework of a pressure correction scheme for the incompressible Navier-Stokes equations.

MATHEMATICAL FORMULATION

The time advancement method

The solution of Navier-Stokes equation is advanced in time with a fractional step scheme [9, 10];

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -\mathscr{N}_l(\mathbf{u}^n, \mathbf{u}^{n-1}) - \mathscr{G}\phi^{n-1} + \mathscr{L}(\mathbf{u}^*, \mathbf{u}^n) \tag{5}$$

$$\mathscr{L}\phi = \frac{1}{\Delta t}\mathscr{D}\mathbf{u}^* \tag{6}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \mathscr{G}\phi^n, \tag{7}$$

where $\mathbf{u}^*$ is the predicted (non-solenoidal) velocity field, $\mathbf{u}^n$ is the solenoidal velocity field at time-step n, Δt is the time step, $\mathscr{N}_l$ is the discrete non-linear operator, $\mathscr{G}$ is the discrete gradient operator, $\mathscr{L}$ is the discrete Laplacian, ϕ is a projection variable (related to the pressure field).

Following Uhlmann [4], in order to impose the boundary values on the embedded boundary the advancement of the momentum equations is carried out in two stages. First, a fully explicit version of (5) is advanced in time from t^n to $t^n + \Delta t$ without imposing any value on the embedded boundary:

$$\mathbf{u}^* = \mathbf{u}^n - \Delta t\left[\mathscr{N}_l(\mathbf{u}^n, \mathbf{u}^{n-1}) - \mathscr{G}\phi^{n-1} + \mathscr{L}(\mathbf{u}^n)\right] \tag{8}$$

The velocity field obtained from (8) is *interpolated* on the embedded geometry Γ, which discretized through a number of markers of coordinates $\mathbf{X}_k$, using an interpolation operator (to be described later) $\mathscr{I}$:

$$\mathbf{U}^*(\mathbf{X}_k) = \mathscr{I}(\mathbf{u}^*) \tag{9}$$

The value of $\mathbf{U}^*$ allows to determine a distribution of singular forces along Γ that would restore the prescribed boundary values $\mathbf{U}^\Gamma(X_k, t)$ on the embedded boundary:

$$\mathbf{F}^*(\mathbf{X}_k) = \frac{\mathbf{U}^\Gamma - \mathbf{U}^*(\mathbf{X}_k)}{\Delta t}. \tag{10}$$

The singular force field defined over Γ is then transformed by a *spreading* operator $\mathscr{C}$ into a volume force-field defined on the mesh points $\mathbf{x}_{i,j,k}$:

$$\mathbf{f}^*(\mathbf{x}_{i,j,k}) = \mathscr{C}\left(\mathbf{F}^*(\mathbf{X}_k)\right). \tag{11}$$

Finally, the time-advancement of the momentum equations (in its explicit or semi-implicit version) is undertaken including the computed force field given by (11):

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -\mathscr{N}_l(\mathbf{u}^n, \mathbf{u}^{n-1}) - \mathscr{G}\phi^{n-1} + \mathscr{L}(\mathbf{u}^*, \mathbf{u}^n) + \mathbf{f}^* \qquad (12)$$

At this point the algorithm continues with the usual solution of the pressure Poisson equation and the consequent projection step (7). The key points in the algorithm concern the proper definition of the interpolation operator $\mathscr{I}$ and of the convolution kernel involved in $\mathscr{C}$. This will be the subject of the next section.

Interpolation and convolution: Dirac's delta approximants

To define the specific kernel of the interpolation and convolution operators that has been selected and justify such a choice, some basic notions about the *Reproducing Kernel Particle Method* (RKPM) are reviewed. We will just focus on the concepts and results that have a direct use in the context of the present methodology. A complete review and analysis of the method con be found in [1, 11]. Applications of the RKPM embedded domain technique in a finite element framework is illustrated in [12].

The approximation $f^a(x)$ of the value of a given smooth function $f(s)$ at a certain point of coordinate x can be generated by a *kernel* approximation:

$$f^a(x) = \int_\Omega w_d(x-s) f(s) ds \qquad (13)$$

where $w_d(x-s) > 0$ is the kernel or a weighting function, and the subscript indicates that the kernel also depends on an additional parameter d, the dilatation parameter. The kernel function is assumed to be of compact support (*i.e.*, nonzero in a sub-domain Ω_I of Ω and zero outside in $\Omega \backslash \Omega_I$). The sub-domain Ω_I is determined by the dilatation parameter d. Note that if the kernel function is the delta function, $f^a(x) = f(x)$ and the function is said to be reproduced exactly. A possible approximation to the exact delta function that we will use in the following, is the one suggested by Roma *et al.* [8]:

$$w_d(r) = \begin{cases} \frac{1}{6}\left(5 - 3|r| - \sqrt{-3(1-|r|)^2 + 1}\right), & 0.5 \le |r| \le 1.5 \\ \frac{1}{3}\left(1 + \sqrt{-3r^2 + 1}\right), & |r| \le 0.5 \\ 0, & \text{otherwise,} \end{cases} \qquad (14)$$

where $r = (x-y)/d$. The function in (14) was determined by requiring that a certain set of properties be satisfied by the discrete version of Dirac delta function [8]:

1. $w_d(r)$ is continuous for all real numbers r;
2. $w_d(r) \ge 0$, $|r| \ge 1.5$;
3. $\sum_l w_d(r-l) = 1$, $\forall r$;
4. $\sum_l (r-l) \cdot w_d(r-l) = 0$, $\forall r$;
5. $\sum_l (w_d(r-l))^2 = 1/2$, $\forall r$;

The last three properties involve values $l \in \mathbb{N}$ and, therefore, allow meeting conditions (3) and (4) only if the nodes are equispaced. In order to comply with the above-mentioned conditions on a non-uniform sequence of nodes, following Liu *et al.* [11], we introduce a modified version of (14) as

$$\begin{aligned} \widetilde{w}_d(x-s) = [&b_0(s,d) + b_1(s,d)(x-s) + \\ &b_2(s,d)(x-s)^2 + \ldots] \cdot w_d(x-s) \end{aligned} \qquad (15)$$

where the polynomial coefficients $b_0(s,d), b_1(s,d), b_2(s,d)\ldots$ are determined by imposing the reproducing conditions

$$\int_\Omega \widetilde{w}_d(x-s) ds = 1 \qquad (16)$$

$$\int_\Omega (x-s)^i \cdot \widetilde{w}_d(x-s) ds = 0 \qquad (i = 1, 2, ..) \qquad (17)$$

which are the continuous equivalent of the third and fourth conditions met by $w_d(x-s)$ on an uniform mesh. It can be shown that the moment conditions:

$$\begin{aligned} \widetilde{m}_0(x) &= \int_\Omega \widetilde{w}_d(x-s) ds = 1 \quad \text{and} \\ \widetilde{m}_i(x) &= \int_\Omega (x-s)^i \cdot \widetilde{w}_d(x-s) ds = 0 \end{aligned} \qquad (18)$$

(for $i = 1, 2, \ldots$) are equivalent to the polynomial reproducing conditions:

$$x^i = \int_\Omega s^i \cdot \widetilde{w}_d(x-s) ds, \; i = 0, 1, 2\ldots \qquad (19)$$

If we insert in (18) the definition of the polynomial correction (15), it is possible to write the conditions on the modified moments $\widetilde{m}_i(x)$ in terms of the original moments $m_i(x) = \int_\Omega (x-s)^i \cdot w_d(x-s) ds$:

$$\begin{aligned} \widetilde{m}_0(x) &= \int_\Omega \widetilde{w}_d(x-s) ds = \sum_{i=0}^{n} b_i(x) m_i(x) = 1 \\ \widetilde{m}_1(x) &= \int_\Omega (x-s) \cdot \widetilde{w}_d(x-s) ds = \sum_{i=0}^{n} b_i(x) m_{i+1}(x) = 0 \\ \cdots &= \cdots \\ \widetilde{m}_n(x) &= \int_\Omega (x-s)^n \cdot \widetilde{w}_d(x-s) ds = \sum_{i=0}^{n} b_i(x) m_{i+n}(x) = 0 \end{aligned} \qquad (20)$$

By using any quadrature technique in (20), a symmetric linear system:

$$\begin{pmatrix} m_0 & m_1 & \cdots\cdots & m_n \\ m_1 & m_2 & \cdots\cdots & m_{n+1} \\ \cdots & \cdots & \cdots\cdots & \\ \cdots & \cdots & \cdots\cdots & \\ m_n & m_{n+1} & \cdots\cdots & m_{2n} \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ \cdot \\ \cdot \\ b_n \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{pmatrix} \qquad (21)$$

is obtained for the unknown polynomial coefficients b_i, $i = 0,1,2....$

Modified window functions that verify the reproducing conditions in higher dimensions can be obtained by imposing the exact representation of a complete polynomial basis. In 2D all the second order polynomials are linear combinations of $\{1,x,y,xy,x^2,y^2\}$, and the *mother* window function can be given as a Cartesian product of (14) with itself: $w_{\delta,\eta}(x-s,y-t) = w_\delta(x-s)\cdot w_\eta(y-t)$. Here, δ and η are the dilatation parameters in the two coordinate directions. Next we look for corrected window functions

$$\widetilde{w}_{\delta,\eta}(x-s,y-t) = \left[b_0+(x-s)b_1+(y-t)b_2+(x-s)\right.$$
$$\left.(y-t)b_3+(x-s)^2b_4+(y-t)^2b_5\right] \times w_{\delta,\eta}(x-s,y-t) \quad (22)$$

(where $b_i = b_i(\delta,\eta,x,y)$ for $i = 0,...,5$) in 2D. By imposing the exact representation of the members of the polynomial basis, a symmetric linear system $\mathbf{M}_{2D}\,\vec{b} = \vec{e}_1$ is obtained. Where $\vec{b}$ is the vector of unknown polynomial coefficients (same order as the one in the definition of the given correction polynomials) and $\vec{e}_1$ is the first unit vector of the canonical basis in $\mathbb{R}^6$ in 2D. The elements of the matrices are:

$$m_{i,j} = \int_{\Omega_I}(x-s)^i(y-t)^j\cdot w_{\delta,\eta}(x-s,y-t)dsdt \text{ in 2D} \quad (23)$$

To ensure that the linear system is well behaved some precautions must be taken in the choice of the discrete support for Ω_I and in the solution of the linear system. These topics together with more implementation details are given in the following section.

Interpolation and convolution: discrete approach

We now discuss the implementation of RKPM method in a discrete, generalized-coordinate finite-difference or finite-volume code. As a first step the embedded boundary curve is discretized into a number of nodes $\mathbf{X}_I$, $I = 1..N_e$. Around each node $\mathbf{X}_I$ we define a rectangular cage Ω_I (Figure 1) with the following properties: (i) it must contain at least three nodes of the underlying mesh (from now on termed "the mesh") for each direction; (ii) the number of nodes of the mesh contained in the cage must be minimized. The modified kernel will be identically zero outside Ω_I. With reference to the definition of the window function proposed by Roma *et al.* [8], the edges of the rectangle will measure 3δ in x direction and 3η in y. The reason to have at least three nodes in each direction within the cage is related with avoiding a singular moment matrix when considering second order polynomials for the correction of the original window function.

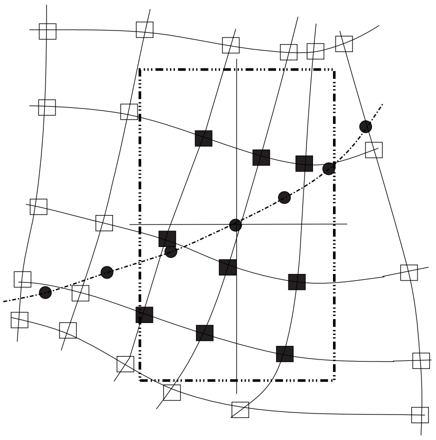

FIGURE 1. Definition of the support cage. ● are the nodes on the embedded curve, □ are the grid points, ■ are the grid points within the support; the dashed line is the embedded curve and the dashed rectangle is Ω_I centered about X_I (the center of the rectangle).

In order to determine the dimensions of the rectangular support centered in $\mathbf{X}_I$, first we look for the mesh node $\mathbf{x}_{\hat{i},\hat{j}} = (x_{\hat{i},\hat{j}}, y_{\hat{i},\hat{j}})$ closest to $\mathbf{X}_I$, next we consider (in a structured mesh) the nodes neighboring $\mathbf{x}_{\hat{i},\hat{j}}$ (*i.e.*, $\mathcal{N}_I = \left\{\mathbf{x}_{\hat{i}+k,\hat{j}+l}\right\}, k = -1,0,1, l = -1,0,1$) and evaluate $h_x^\pm$ and $h_y^\pm$ as:

$$h_x^+(\mathbf{X}_I) = \max\left\{|x_{i,j}-x_{i-1,j}| : x_{i,j}, x_{i-1,j} \in \mathcal{N}_I\right\}, \quad (24)$$
$$h_x^-(\mathbf{X}_I) = \min\left\{|x_{i,j}-x_{i-1,j}| : x_{i,j}, x_{i-1,j} \in \mathcal{N}_I\right\}, \quad (25)$$
$$h_y^+(\mathbf{X}_I) = \max\left\{|y_{i,j}-y_{i,j-1}| : y_{i,j}, y_{i,j-1} \in \mathcal{N}_I\right\}, \quad (26)$$
$$h_y^-(\mathbf{X}_I) = \min\left\{|y_{i,j}-y_{i,j-1}| : y_{i,j}, y_{i,j-1} \in \mathcal{N}_I\right\}. \quad (27)$$

Based on these values, we define the length of the edges of the rectangle ($3\delta_I$ and $3\eta_I$) through the local dilatation factors:

$$\delta_I = \left(\frac{5}{6}h_x^+(\mathbf{X}_I) + \frac{1}{6}h_x^-(\mathbf{X}_I) + \varepsilon_x(\mathbf{X}_I)\right) \quad (28)$$

$$\eta_I = \left(\frac{5}{6}h_y^+(\mathbf{X}_I) + \frac{1}{6}h_y^-(\mathbf{X}_I) + \varepsilon_y(\mathbf{X}_I)\right) \quad (29)$$

where $\varepsilon_x(\mathbf{X}_I)$ and $\varepsilon_y(\mathbf{X}_I)$ are small fractions of the typical local mesh spacing. Those terms are added to avoid the eventuality of the support boundary touching some of the support nodes (to be defined later) since in this case the window function would take on zeros values on those nodes, thus making the discretized moment matrix singular.

As a final step, we need to determine a set of grid nodes within the support cage:

$$\mathscr{S}_I = \left\{ \mathbf{x}_{i,j} : |x_{\hat{i},\hat{j}} - x_{i,j}| < \frac{3}{2}\delta_I, \quad |y_{\hat{i},\hat{j}} - y_{i,j}| < \frac{3}{2}\eta_I \right\}. \tag{30}$$

We have verified that with this particular choice the set of nodes within the support is nine almost anywhere when the underlying mesh is reasonably smooth.

To assemble the local window function centered in $\mathbf{X}_I$, the elements of the moment matrix $\mathbf{M}^I = \{m^I_{i,j}\}$

$$m^I_{i,j} = \int_{\Omega_I} (x-s)^i (y-t)^j \cdot w_{\delta,\eta}(x-s, y-t) ds dt \tag{31}$$

must be evaluated numerically. To this end we approximate the entries in the moment matrix relative to node $\mathbf{X}_I = (X_I, Y_I)$ using the mid-point quadrature rule:

$$m^I_{i,j} = \sum_{k,l \in \mathscr{S}_I} (x_{k,l} - X_I)^i (y_{k,l} - Y_I)^j \times w_{\delta_I,\eta_I}(x_{k,l} - X_I, y_{k,l} - Y_I) \Delta A_{k,l}, \tag{32}$$

where $\Delta A_{k,l}$ is the area of the cell centered about $\mathbf{x}_{k,l}$. The extension of such an approximation to the elements of the moment matrix in the three-dimensional case is an easy task.

Once the discrete counterpart of the moment matrix is assembled for each $\mathbf{X}_I$ node, the coefficients of the correction polynomials are found by solving

$$\mathbf{M}^I \mathbf{b}^I = \mathbf{e}_1, \; I = 1, ..., N_e \tag{33}$$

(*i.e.*, a 6×6 symmetric linear systems in a two-dimensional case with a second order correction polynomial). A word of caution about the solution of these linear systems is necessary. Due to the very low values that the window function may take at the nodes close to the boundary of the support cage, the moment matrix may become ill-conditioned, giving inaccurate values of the polynomial coefficients. A possible way of limiting round-off errors is to rescale the linear system by solving the equivalent system

$$\mathbf{H}^I \mathbf{M}^I (\mathbf{H}^I)^{-1} \, \mathbf{b}^I = \mathbf{e}_1 \tag{34}$$

in two stages:

$$\begin{gathered} \mathbf{H}^I \mathbf{M}^I \mathbf{c}^I = \mathbf{e}_1, \text{ and } \mathbf{b}^I = \mathbf{H}^I \mathbf{c}^I, \text{ with} \\ \mathbf{H}^I = \text{diag}\left(1, \frac{1}{\delta_I}, \frac{1}{\eta_I}, \frac{1}{\delta_I \eta_I}, \frac{1}{\delta_I^2}, \frac{1}{\eta_I^2}\right). \end{gathered} \tag{35}$$

The coefficient matrix of the first linear system can be equivalently obtained by normalizing the distances $(x_{k,l} - X_I)$ and $(y_{k,l} - Y_I)$ appearing in (32) with the dilatation parameters (δ_I and η_I). The matrix product that follows is needed to undo the scaling.

The methodology developed so far allows the definition of a localized window function $\widetilde{w}_{\delta_I,\eta_I}(\mathbf{x} - \mathbf{X}_I)$ to be used in the convolution integrals appearing in the time advancement scheme of the Navier-Stokes algorithm. In particular, given a component of the velocity field $u_i(x,y)$ known at the mesh nodes $\mathbf{x}_{i,j}$, the interpolated value at node $\mathbf{X}_I$ on the embedded line can be approximated numerically by:

$$U_i(\mathbf{X}_I) = \sum_{k,l \in \mathscr{S}_I} u_i(\mathbf{x}_{l,k}) \cdot \widetilde{w}_{\delta_I,\eta_I}(x_{k,l} - X_I, y_{k,l} - Y_I) \Delta A_{k,l} \tag{36}$$

having used the same quadrature rule as in (32). Conversely, once the force component for each momentum equation on each embedded node $F_i(\mathbf{X}_I)$ has been found by *forcing* the desired boundary values, the distribution of the singular forces over the mesh nodes can be obtained using as a discrete convolution operator the quadrature formula:

$$f_i(\mathbf{x}_{l,k}) = \sum_{I=1}^{N_e} F_i(\mathbf{X}_I) \cdot \widetilde{w}_{\delta_I,\eta_I}(x_{k,l} - X_I, y_{k,l} - Y_I) \; \varepsilon_I \; \Delta s_I \tag{37}$$

where Δs_I is length of the arc joining $\mathbf{X}_{I+1/2}$ to $\mathbf{X}_{I-1/2}$, and ε_I is a characteristic length related to the local dilatation coefficients of the window function $\widetilde{w}_{\delta_I,\eta_I}(x_{k,l} - X_I, y_{k,l} - Y_I)$ and to the distribution of the markers on the embedded boundary line. The conservations properties concerning the spread values of forces and torques are verified independently of the value assigned to ε_I, whose role is only to control the intensity of the spread force field.

The Navier-Stokes solver

The immersed-boundary method discussed has been implemented in a three-dimensional curvilinear finite-volume solver for the incompressible Navier-Stokes equations. The time-advancement follows the procedure described above [Equations (5–7)].

The incompressible Navier-Stokes equations are discretized on a non-staggered grid using a curvilinear finite volume code. The method of Rhie and Chow [13] is used to avoid pressure oscillations. Both convective and diffusive fluxes are approximated by second-order central differences. A second-order semi-implicit fractional-step procedure [14] is used for the temporal discretization. The Crank-Nicolson scheme is used for the temporal discretization of wall-normal diffusive terms, and the second-order Adams-Bashforth scheme for all the other terms. Fourier transforms are used to reduce the three-dimensional Poisson equation into a series of two-dimensional Helmoltz equations in wavenumber space, which are then solved iteratively using the biconjugate gradient stabilized (BiCGStab) method. The code is parallelized using the MPI message-passing library and the domain-decomposition technique, and has been widely tested [15–18] in simulations of turbulent flows using curvilinear, body-fitted grids.

RESULTS

In this section the accuracy, convergence and robustness of the scheme will be established. We consider first the laminar flow over a circular cylinder; we then examine the flow over a two-dimensional hill to show the implementation of the scheme on non-uniform and non-orthogonal grids.

Flow over a circular cylinder

The steady flow over a circular cylinder is considered at two Reynolds numbers, $Re_D = 30$ and $Re_D = 185$, (based on the free-stream velocity U_∞ and the cylinder diameter D). The low Reynolds number flow remains steady, while at $Re_D = 185$ periodic vortex shedding is expected. In the latter case the results depend critically on the accurate reproduction of the vorticity field in the vicinity of the cylinder, which, in turn, depends on a correct prescription of pressure and shear forces at the cylinder boundary.

The steady-state flow over a cylinder at $Re_D = 30$ is simulated with a domain of dimensions $[-9D, 40D]$ in the streamwise (x) direction and $[-17D, 17D]$ in the vertical (y) direction; the centre of the cylinder is at $(0,0)$. The dimensions of the domain are comparable to those used in other numerical studies [4, 5]. A steady uniform flow field is given at the inlet plane and at the outlet plane convective boundary condition are specified. A free-slip condition is applied on the top and the bottom walls.

The grid is uniform near the cylinder, in the region $-0.6D \leq x \leq D$ and $-D \leq y \leq D$. Outside this region, the grid is stretched with a stretching ratio of less than 1.02. Four different grids (with $\Delta x = \Delta y = 0.0025D$, $0.005D$, $0.01D$ and $0.02D$ in the uniform region) are considered with 1260, 629, 316 and 153 uniformly spaced Lagrangian points on the boundary of the cylinder. The results for the finest grid are considered accurate, and the L_2 and L_∞ norms of the error obtained on the coarser grids are calculated and shown in Figure 2. The results demonstrate the second-order accuracy of the method.

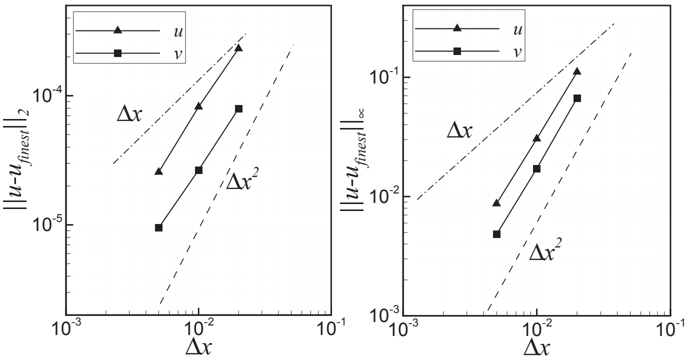

FIGURE 2. Grid convergence study. *(a)* L_2 and *(b)* L_∞ norms of the error.

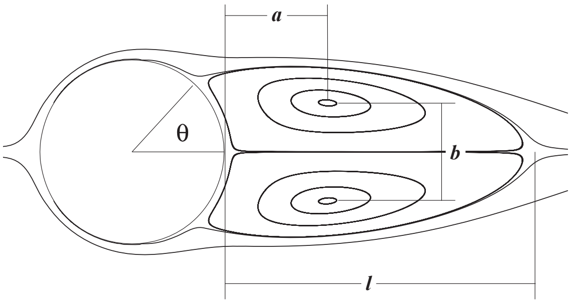

FIGURE 3. Definition of different parameters of the wake.

The simulation results are also compared with the experimental data. The most important physical feature of this flow is the presence of a recirculating region in the wake of the cylinder. The important parameters associated with the wake are shown in Figure 3; l is the length of the wake, a is the distance from the cylinder to the centre of the wake vortex, b is the distance between the centres of the wake vortices, and θ is the angle of separation measured from x-axis. Another important flow parameter is the drag coefficient, $C_D = 2\mathscr{D}/\rho U_\infty^2 D^2$ (where $\mathscr{D}$ is the drag force). In the current scheme interpolation and spreading operators conserve the force therefore the drag force $\mathscr{D}$ can be calculated directly from the summation of the forces at all

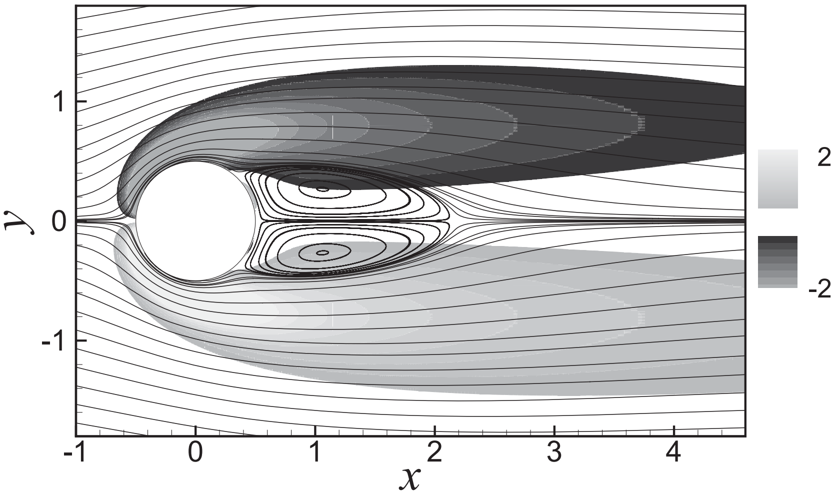

FIGURE 4. Streamlines and vorticity contours for the steady-state flow over a cylinder at $Re_D = 30$.

Lagrangian points:

$$\mathscr{D} = \sum_n F_x^*(\mathbf{X}_n)\Delta\mathscr{A}_n. \tag{38}$$

where $F_x^*(\mathbf{X}_n)$ is the x-component of the force at Lagrangian point, resulting from u-momentum equation and $\Delta\mathscr{A}_n$ is the area associated with the Lagrangian point.

The streamlines and vorticity contours for this flow are shown in Figure 4. The figure shows the expected flow topology, and the wake parameters computed from the simulation are in good agreement with the experimental data, as shown in Table 1.

TABLE 1. Comparison of wake parameters and drag coefficient for steady-state flow over cylinder at $Re_D = 30$ with experimental data

	l/D	a/D	b/D	θ	C_D
Present	1.70	0.56	0.52	48.05^o	1.80
Coutanceau [19]	1.55	0.54	0.54	50.00^o	—
Tritton [20]	—	—	—	—	1.74

We then examined a higher Re_D case, to verify that the unsteadiness of the flow could be captured correctly. The domain and the boundary conditions for the higher Reynolds number case were similar to the low-Re_D case. Only two grids were considered, with $\Delta x = \Delta y = 0.01D$ and $0.005D$, respectively, in the region of interest ($-0.6D \leq x \leq D$ and $D \leq y \leq D$).

Figure 5 shows instantaneous vorticity contours and a time series of the drag and lift coefficients, C_D and C_L (where $C_L = 2\mathscr{L}/\rho U_\infty^2 D^2$, and $\mathscr{L}$ is the lift force on the cylinder) obtained using the finer grid. The periodic fluctuations of drag and lift co-efficients indicate a stable vortex shedding behind the cylinder. The instantaneous vorticity contours for this flow are shown in Figure 5*(b)*. The numerical simulation captures the Kármán vortex street formed in the wake. The coefficients of drag, lift and Strouhal number $St = D\omega/U_\infty$ (where ω is the shedding frequency) are compared with the values reported in the literature in Table 2. The results obtained with the present method are well within the range of values obtained by other researchers.

TABLE 2. Comparison of coefficients of drag, lift and Strouhal number for flow over cylinder at $Re_D = 185$.

	C_D	C_L^{rms}	St
Present $\Delta x = 0.005D$	1.430	0.423	0.196
Present $\Delta x = 0.01D$	1.509	0.428	0.199
Vanella and Balaras [5]	1.377	0.461	-
Guilmineau and Queutey [21]	1.280	0.443	0.195
Lu and Dalton [22]	1.310	0.422	0.195
Williamson [23]	-	-	0.193

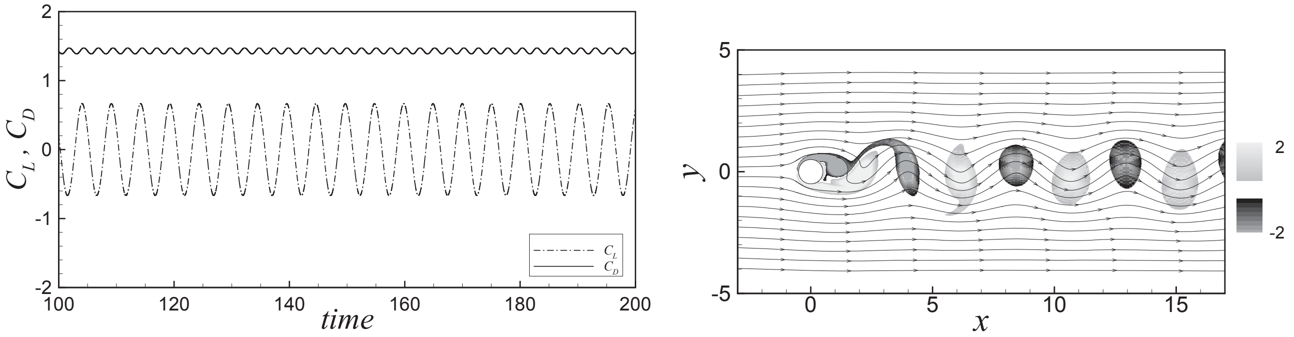

FIGURE 5. Flow over a cylinder at $Re_D = 185$. Left: Time-series of lift and drag coefficients; right: instantaneous vorticity contours and streamlines.

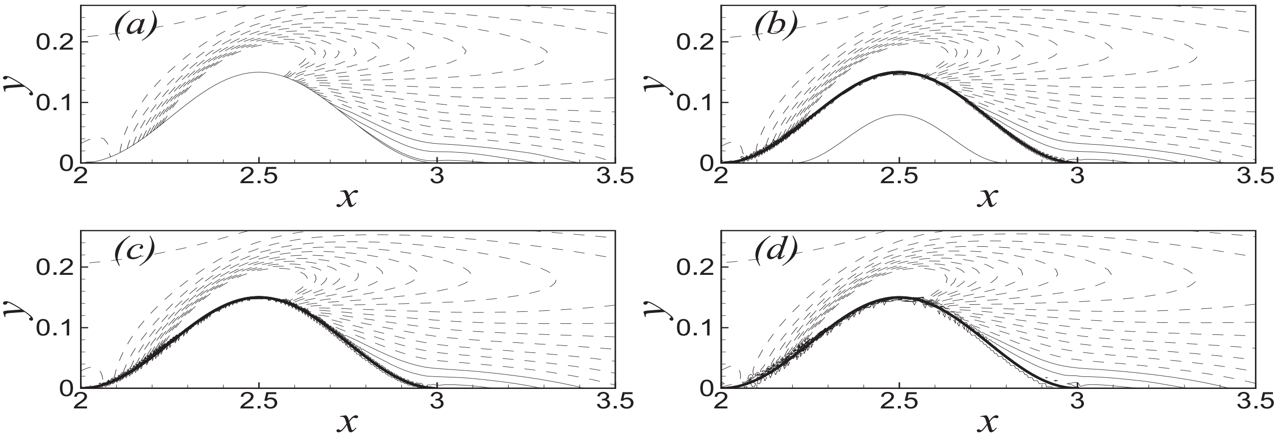

FIGURE 6. Comparison of vorticity for flow over the hill. *(a)* Body-fitted non-orthogonal grid. *(b)* Non-orthogonal grid with immersed boundary. *(c)* Non-uniform orthogonal grid with immersed body. *(d)* Uniform orthogonal grid with immersed boundary. 21 equally spaced contour levels from 2 to -8 are shown here.

Flow over a two-dimensional hill

An important property of the current forcing scheme is its applicability to non-uniform and non-orthogonal grids without loss of accuracy. To demonstrate this property a two dimensional flow over a small hill in a channel is considered here with a Reynolds number of $Re = 600$ based on the channel height H. The shape of the hill is defined by $y_h = 0.15H\sin^2[\pi(x-2.0)]$; thus, the maximum height is 15% of the channel height; its peak is at $x = 2.5$. The no-slip boundary condition is enforced on the top and bottom walls of the channel and a periodic boundary condition is assumed in the streamwise direction. The domain, however, is long enough that the flow returns to a fully developed state, with a parabolic profile, before the end of the computational domain.

In this study four different grids are considered. First a non-orthogonal body-fitted grid is used, which conforms to the hill. In this case hill becomes an integral part of the domain boundary and direct no-slip boundary condition is applied. The maximum grid size in the region $2H \le x \le 4H$ and $0 \le y \le 0.25H$ is $0.01H$. This is the region of interest for this flow, since it covers the hill and the recirculation zone. The second grid is also a non-orthogonal grid, but the hill does not coincide with a grid line, and the no-slip conditions on the hill itself are imposed by the immersed-boundary method. The maximum grid size in the region of interest for this case is also $0.005H$. The third grid is a non-uniform orthogonal grid, is clustered near the top and bottom of the hill in the y-direction and near the leading and trailing part of the hill in the x-direction. In the central part of the hill the grid is stretched in both directions. Here the grid size in the region of interest varies between $0.0025H$ and $0.0075H$. The fourth grid is a uniform orthogonal grid, also with a grid size $\Delta x = \Delta y = 0.005H$ in the region of interest. The hill is defined as an immersed boundary with 650 uniformly distributed Lagrangian points along the geometry. The no-slip condition on the hill is applied through the forcing described previously.

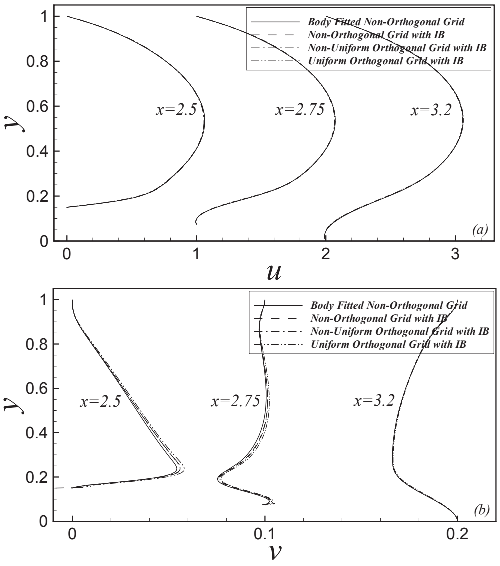

FIGURE 7. Comparison of velocity profiles at $x = 2.5$, $x = 2.75$ and $x = 3.2$ over the hill. *(a) u*, *(b) v*.

The vorticity contours on the hill and in the recirculation region for the four grids are shown in Figure 6. The non-orthogonality and non-uniformity of the grid has no significant effect on the forcing as far as large scale flow features are concerned. The vorticity contours for all the three grids with forcing are in accord with the body-fitted non-orthogonal grid. A detailed comparison of different grids is also shown in Figure 7 through u and v velocity profiles. The velocity profiles are shown at three locations, $x = 2.5$ (peak of the hill), $x = 2.75$ (50% down the hill) and $x = 3.2$ (in the recirculation region). The u and v velocity profiles for all the four grids are in very good agreement with each other.

CONCLUSIONS

We have extended an immersed-boundary method based on the *Reproducing Kernel Particle Method* (RKPM) proposed by Liu *et al.* [1] to non-orthogonal, non-uniform grids. The method satisfies the integral conservation properties for the first two moments of the force field. This makes the calculation of the drag and lift forces exerted by the fluid on the body straightforward, as it becomes a simple integration. No corrections are necessary, as required by other methods.

We have applied this method to the simulation of the flow around a 2D cylinder, both in the steady and unsteady flow regimes. The simulation results compare very well with the data in the literature. We have also performed calculations of the flow over a two-dimensional bump, to verify that the IBM proposed maintained its accuracy on non-orthogonal and non-uniform meshes. The IBM compares very well with the results of a simulation performed using a body-fitted grid. Extensions of the method to three-dimensional flows and moving bodies are ongoing.

ACKNOWLEDGMENT

This research was began while AP was visiting the Department of Mechanical and Materials Engineering at Queen's University. The computational support of the High Performance Computing and Virtual Laboratory, Queen's University site, is gratefully acknowledged.

REFERENCES

[1] Liu, W. K., Chen, Y., Uras, R. A., and Chang, C. T., 1996. "Generalized multiple scale reproducing kernel particle methods". *Comput. Meth. Appl. Mech. Eng.,* ***139***(1-4), pp. 91–157.

[2] Peskin, C. S., 1972. "Flow patterns around heart valves: a numerical method". *J. Comput. Phys.,* ***10***, pp. 252–271.

[3] Fadlun, E. A., Verzicco, R., Orlandi, P., and Mohd-Yusof, J., 2000. "Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations". *J. Comput. Phys.,* ***161***, pp. 35–60.

[4] Uhlmann, M., 2005. "An immersed boundary method with direct forcing for the simulation of particulate flows". *J. Comput. Phys.,* ***209***(2), pp. 448–476.

[5] Vanella, M., and Balaras, E., 2009. "A moving-least-squares reconstruction for embedded-boundary formulations". *J. Comput. Phys.,* ***228***(18), pp. 6617–6628.

[6] Moin, P., 2002. "Advances in large eddy simulation methodology for complex flows". *Int. J. Heat Fluid Flow,* ***23***, pp. 710–720.

[7] Roman, F., Napoli, E., Milici, B., and Armenio, V., 2009. "An improved immersed boundary method for curvilinear grids". *Comput. Fluids,* ***38***, pp. 1510–1527.

[8] Roma, A. M., Peskin, C. S., and Berger, M. J., 1999. "An adaptive version of the immersed boundary method". *J. Comput. Phys.,* ***153***, pp. 509–534.

[9] Chorin, A. J., 1968. "Numerical solution of Navier-Stokes equations". *Math. Comput.,* ***22***(104), pp. 745–762.

[10] Temam, R., 1969. "Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires (I)". *Arch. Rat. Mech. Anal.* ***32***(2), pp. 135–153.

[11] Liu, W. K., Jun, S., and Zhang, Y. F., 1995. "Reproducing kernel particle methods". *Int. J. Num. Meth. Fuids,* ***20***(8), pp. 1081–1106.

[12] Wang, X., and Liu, W. K., 2004. "Extended immersed boundary method using FEM and RKPM". *Comput. Meth. Appl. Mech. Eng.,* ***193***(12-14), pp. 1305–1321.

[13] Rhie, C. M., and Chow, W. L., 1983. "Numerical study of the turbulent flow past an airfoil with trailing edge separation". *AIAA J.,* ***21***, pp. 1525–1532.

[14] Kim, J., and Moin, P., 1985. "Application of a fractional

step method to incompressible Navier-Stokes equations". *J. Comput. Phys.,* ***59***, pp. 308–323.

[15] Silva Lopes, A., and Palma, J. M. L. M., 2002. "Simulations of isotropic turbulence using a non-orthogonal grid system". *J. Comput. Phys.,* ***175***(2), pp. 713–738.

[16] Silva Lopes, A., Piomelli, U., and Palma, J. M. L. M., 2006. "Large-eddy simulation of the flow in an S-duct". *J. Turbul.,* ***7***(11), pp. 1–24.

[17] Radhakrishnan, S., Piomelli, U., Keating, A., and Silva Lopes, A., 2006. "Reynolds-averaged and large-eddy simulations of turbulent non-equilibrium flows". *J. Turbul.,* ***7***(63), pp. 1–30.

[18] Radhakrishnan, S., Piomelli, U., and Keating, A., 2008. "Wall-modeled large-eddy simulations of flows with curvature and mild separation". *ASME J. Fluids Eng.,* ***130***(101203).

[19] Coutanceau, M., and Bouard, R., 1977. "Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 1. Steady flow.". *J. Fluid Mech.,* ***79***(2), pp. 231–256.

[20] Tritton, D. J., 1959. "Experiments on the flow past a circular cylinder at low Reynolds numbers". *J. Fluid Mech.,* ***6***, pp. 547–567.

[21] Guilmineau, E., and Queutey, P., 2002. "A numerical simulation of vortex shedding from an oscillating circular cylinder". *J. Fluids Struct.,* ***16***(6), pp. 773–794.

[22] Lu, X. Y., and Dalton, C., 1996. "Calculation of the timing of vortex formation from an oscillating cylinder". *J. Fluids Struct.,* ***10***(5), pp. 527–541.

[23] Williamson, C. H. K., 1988. "Defining a universal and continuous Strouhal–Reynolds number relationship for the laminar vortex shedding of a circular cylinder". *Phys. Fluids,* ***31***(10), pp. 2742–2744.