Combined immersed-boundary/B-spline methods for simulations of flow in complex geometries

By J. Mohd-Yusof

1. Motivation and objectives

For fluid dynamics simulations, the primary issues are accuracy, computational efficiency, and the ability to handle complex geometries. Spectral methods offer the highest accuracy but are limited to relatively simple geometries. In order to accommodate more complex geometries, finite-difference or finite-element methods are generally used. However, these methods suffer from relatively low accuracy, requiring fine meshes to obtain good results. Finite element schemes, while able to handle complex geometries, often require significant computational time for grid generation. Spectral element methods can be used for complex geometries, but the grid stretching inherent in these methods leads to timestep limitations and clustering of grid points in an inefficient manner.

In general, any computational scheme which requires re-gridding to accommodate changes in geometry will incur significant penalties in simulating time-varying geometries. For relatively simple motions, it is possible to use grid-stretching techniques, (Carlson *et al.*, 1995), but these are still slow. Vortex element methods for moving bodies (Koumoutsakos, 1995) are presently under development but are also rather slow, especially with respect to calculation of spectra.

To address these problems, we propose a discrete-time immersed boundary method which will allow implementation of complex moving geometries in existing pseudospectral codes. The method does not incur significant additional cost as compared to the base computational scheme, and changes in surface geometry simply require modification of the input files without any further modification of the code itself.

2. Accomplishments

The immersed boundary method has been successfully implemented and tested in the B-spline/Fourier pseudospectral numerical scheme of Kravchenko *el al.* (1996). The interpolation scheme developed for this application makes use of the compact properties of the B-spline transform to maintain the highest possible order at the immersed surface. The combined scheme has been successfully tested on the laminar ribbed channel case of Choi, Moin and Kim (1991) (hereafter CMK).

2.1 Immersed boundary concept

We begin with an examination of the continuous (in time and space) Navier-Stokes equations to demonstrate the principle of the immersed boundary technique. We consider incompressible flows governed by the Navier–Stokes equations, including the body force term:

$$\frac{\partial \mathbf{u}}{\partial t} = -\mathbf{H} - \nabla P + \frac{1}{Re}\nabla^2 \mathbf{u} + \mathbf{f} \tag{1}$$

and the continuity equation:

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

where Re is the Reynolds number, $\mathbf{u} = (u, v, w)$ is the velocity vector, $\mathbf{H} = \mathbf{u} \times \omega = (H_u, H_v, H_w)$ is the convective term, and $\mathbf{f} = (f_u, f_v, f_w)$ is the forcing vector .

The full Navier-Stokes equations allow the inclusion of an external body force. In incompressible flows, this force is generally assumed to derive from some potential field (e.g. gravity) which is constant and therefore may be neglected. However, the NS equations themselves allow the force to be a function of both time and space. In that event, the divergence of the force may be non-zero and therefore must be included in the Poisson equation for pressure if that equation is used to solve the system.

The immersed boundary method involves specifying the body force term in such a way as to simulate the presence of a flow boundary within the computational domain without altering the computational grid. The advantage of this is that bodies of almost arbitrary shape can be added without grid restructuring, a procedure which is often time-consuming. Furthermore, multiple bodies may be simulated and relative motion of those bodies may be accomplished at reasonable computational cost.

The concept of the immersed boundary technique has been used for pseudo-spectral simulations of flows in complex geometries (Goldstein *et al.* 1995). However, the timestep restriction imposed by their derivation severely limits the applicability of the method to turbulent and other strongly time-dependent flows. This restriction can be removed by the use of a *discrete-time* derivation of the forcing value (Mohd-Yusof 1996). When combined with appropriate choice of internal boundary conditions, this scheme leads to a forcing scheme which does not require any filtering of the forcing field.

A second issue of importance to the immersed boundary method is the ability of the underlying numerical scheme to place a sufficient number of grid points near the immersed boundary to adequately resolve the flow scales in that region. While the grid geometry may be considerably simplified as compared to a body-fitted grid, there is still a fundamental need to tailor the grid point distribution to the underlying flow scales. To this end, we employ a B-spline formulation, which allows flexibility of grid point distribution, zonal embedded grids, and high accuracy (Kravchenko *et al.* 1996). Coupled with Fourier-pseudo-spectral methods, this yields a numerical scheme which allows simulation of flows in complex geometries on Cartesian grids with near-spectral accuracy.

2.2 Numerical method

We now consider the discrete-time Navier-Stokes equations in general form:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\triangle t} = -\mathbf{H} - \nabla P + \frac{1}{Re}\nabla^2 \mathbf{u} + \mathbf{f} \tag{3}$$

We wish to drive the velocity, $\mathbf{u}$, on some surface, Ω, to some desired value, $\mathbf{v}(\Omega)$. Rearrangement of the discrete NS equation gives us the velocity update equation

which is of the form:

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \triangle t(-\mathbf{H} - \nabla P + \frac{1}{Re}\nabla^2\mathbf{u} + \mathbf{f}) \tag{4}$$

If we know $\mathbf{H}$, ∇P and $\nabla^2\mathbf{u}$ then the forcing term is simply:

$$\mathbf{f} = \begin{cases} \mathbf{H} + \nabla P - \frac{1}{Re}\nabla^2\mathbf{u} + \frac{1}{\triangle t}(\mathbf{v} - \mathbf{u}^n), & \text{on } \Omega; \\ 0 & \text{elsewhere.} \end{cases} \tag{5}$$

Velocity-vorticity formulation

Following the same approach as in Kim *et al.* (1991) one can reduce Eqs. (1) and (2) to a fourth-order equation for v, and a second-order equation for the normal component of vorticity g:

$$\frac{\partial}{\partial t}\nabla^2 v = h_v + \frac{1}{Re}\nabla^4 v + \nabla^2 f_v \tag{6}$$

$$\frac{\partial}{\partial t} g = h_g + \frac{1}{Re}\nabla^2 g + f_g \tag{7}$$

$$p + \frac{\partial v}{\partial y} = 0 \tag{8}$$

where

$$p = \frac{\partial u}{\partial x} + \frac{\partial w}{\partial z}, \quad g = \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x} \quad f_g = \frac{\partial f_u}{\partial z} - \frac{\partial f_w}{\partial x} \tag{9}$$

$$h_v = -\frac{\partial}{\partial y}\left(\frac{\partial H_u}{\partial x} + \frac{\partial H_w}{\partial z}\right) + \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2}\right) H_v \tag{10}$$

$$h_g = \frac{\partial H_u}{\partial z} - \frac{\partial H_w}{\partial x} \tag{11}$$

The numerical approximation of the velocity vector $\mathbf{u}$ is $\tilde{\mathbf{u}} = (\tilde{u}, \tilde{v}, \tilde{w})$. It is written in terms of spectral and B-spline functions:

$$\tilde{v}(x, y, z, t) = \sum_{k_x, j, k_z} \hat{v}_j(k_x, k_z, t) e^{ik_x x} e^{ik_z z} B_j^l(y) \tag{12}$$

$$\begin{pmatrix} \tilde{u} \\ \tilde{w} \end{pmatrix}(x, y, z, t) = \sum_{k_x, j, k_z} \begin{pmatrix} \hat{u}_j(k_x, k_z, t) \\ \hat{w}_j(k_x, k_z, t) \end{pmatrix} e^{ik_x x} e^{ik_z z} B_j^{l-1}(y) \tag{13}$$

where $B_j^l(y)$ is the B-spline of order l. B-spline functions of order l are defined on a set of knot points t_j by the following recursive relationship

$$B_j^l(y) = \frac{(y - t_j)}{(t_{j+l} - t_j)} B_j^{l-1}(y) + \frac{(t_{j+l+1} - y)}{(t_{j+l+1} - t_{j+l})} B_{j+1}^{l-1}(y) \tag{14}$$

where a B-spline of order zero is a top hat function, i.e $B_j^0(y) = 1$ if $t_j \leq y \leq t_{j+1}$ and 0 otherwise. Note that the v-velocity is represented in terms of B-splines which are one order higher than the B-spline expansion functions for u and w. This allows the continuity equation, Eq. (8), to be satisfied exactly by the numerical representation.

Using the method of weighted residuals, we obtain the discrete weak forms of Eqs. (6) and (7):

$$\int_V \xi \frac{\partial}{\partial t} \nabla^2 \tilde{v} dV = \int_V \xi \tilde{h}_v dV + \int_V \xi \frac{1}{Re} \nabla^4 \tilde{v} dV + \int_V \xi \nabla^2 \tilde{f}_v dV \tag{15}$$

$$\int_V \zeta \frac{\partial}{\partial t} \tilde{g} dV = \int_V \zeta \tilde{h}_g dV + \int_V \zeta \frac{1}{Re} \nabla^2 \tilde{g} dV + \int_V \zeta \tilde{f}_g dV \tag{16}$$

where ξ and ζ are the weight functions, which we select to be:

$$\xi(x, y, z) = e^{-ik_x' x} e^{-ik_z' z} B_i^l(y)$$

,

$$\zeta(x, y, z) = e^{-ik_x' x} e^{-ik_z' z} B_i^{l-1}(y). \tag{17}$$

Evaluating the integrals in Eqs. (15) and (16), we obtain two systems of ordinary differential equations for $\hat{\mathbf{v}}$ and $\hat{\mathbf{g}}$ for each independent Fourier mode (k_x, k_z):

$$\mathbf{M}_v \frac{d\hat{\mathbf{v}}}{dt} = \mathbf{D}_v \hat{\mathbf{v}} + \mathbf{R}_v(\hat{\mathbf{u}}, \hat{\mathbf{v}}, \hat{\mathbf{w}}) + \mathbf{F}_v \tag{18}$$

$$\mathbf{M}_g \frac{d\hat{\mathbf{g}}}{dt} = \mathbf{D}_g \hat{\mathbf{g}} + \mathbf{R}_g(\hat{\mathbf{u}}, \hat{\mathbf{v}}, \hat{\mathbf{w}}) + \mathbf{F}_g \tag{19}$$

where $\mathbf{R}_v, \mathbf{R}_g$ are the expressions resulting from the nonlinear terms, $\mathbf{F}_v, \mathbf{F}_g$ are the expressions for the forcing terms, and $\mathbf{M}_v, \mathbf{M}_g, \mathbf{D}_v, \mathbf{D}_g$ are banded matrices. The matrix elements of $\mathbf{M}_v, \mathbf{M}_g, \mathbf{D}_v, \mathbf{D}_g$ are given in Kravchenko *et al* (1996).

The time-advancement of Eqs. (18) and (19) is carried out with a semi-implicit scheme that uses Crank-Nicholson for the viscous terms and third order Runge-Kutta for the nonlinear terms *(rk3)*. All time dependent simulations were performed with variable time steps and with the CFL number never exceeding $\sqrt{3}$ to satisfy the stability condition of the third order Runge-Kutta scheme.

In this context, the form of the forcing function becomes:

$$\mathbf{f} = \begin{cases} (\alpha \mathbf{H}^n + \beta \mathbf{H}^{n-1}) + \nabla P - \dfrac{1}{Re} \nabla^2 \mathbf{u}^n + \dfrac{1}{\triangle t}(\mathbf{v} - \mathbf{u}^n), & \text{on } \Omega; \\ 0 & \text{elsewhere} \end{cases} \tag{20}$$

where α and β are the Runge-Kutta coefficients.

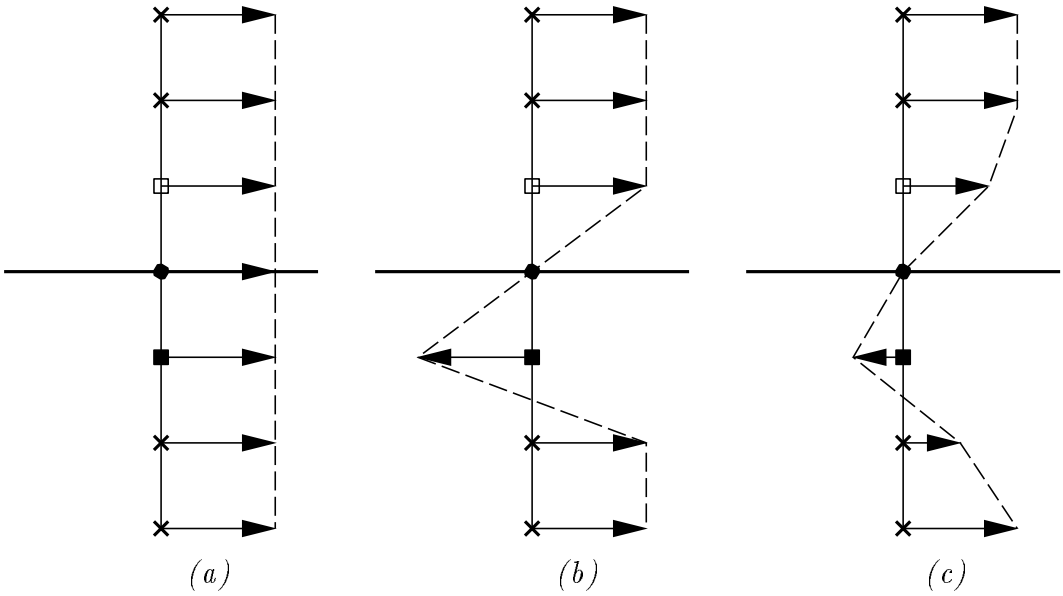


FIGURE 1. Sketch showing the effect of forcing and of a central difference diffusion operator on the velocity field imposed by the forcing. Forcing is applied at the surface (•) to drive the velocity to zero. The velocity at the point immediately interior to the surface (■) is forced to be the reverse of the velocity at the point immediately exterior to the surface (□). (a) initial velocity field, (b) velocity field imposed by forcing, (c) velocity field after diffusion step.

Generation of no-slip walls

We consider the generation of a no-slip wall, with the objective of obtaining a smooth velocity field.

In order to obtain a smooth velocity field without resorting to smoothing of the forcing function, we apply forcing on a set of points adjacent to the surface and interior to the body. At these points (see Fig. 1), we reverse tangential velocities across the solid surface and preserve normal velocities. This makes the velocity gradients smooth across the boundary, minimizing any error due to the estimation of the diffusion term.

Consider first the velocity component tangential to the surface. If we make the velocity gradient linear across the surface, we will achieve the smoothest local tangential velocity. This is accomplished by simply reversing tangential velocities across the surface. The internal tangential velocity is scaled to produce no-slip at the desired location according to $u_{tan,i} = \xi u_{tan,o}$. The scaling factor is set to $\xi = x_i/x_o$ where x_i and x_o are the distances from the desired surface location to the internal and external flow reversal points, respectively.

Notice that the effect of a second-order diffusion operator on the resulting velocity field (Fig. 1) will be to maintain the desired velocity at the surface. If the initial field is uniform, then the velocity field imposed by the forcing will reverse flow across the boundary. The effect of diffusion will then maintain the no-slip condition at the virtual surface, and vorticity will naturally diffuse into the flow. The effect

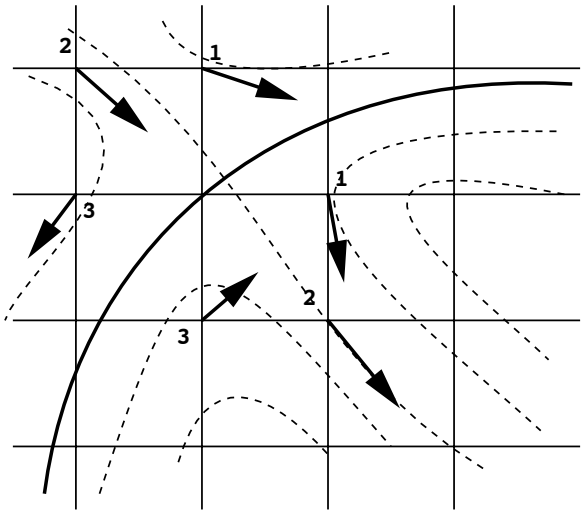


FIGURE 2. Sketch showing how the simple reversal of tangential component and preservation of normal velocity across the virtual surface leads to the expected stagnation point flow. The numbers indicate pairs of flow-reversal points.

of this operator on the 'external' flow will be identical to that in the presence of a conventional Dirichlet boundary: the vorticity at the surface will be diffused into the flow by the action of viscosity.

Since we attempt to reverse tangential velocities across a solid surface, the resultant velocity gradient normal to the surface is linear, and the velocity gradient tangent to the surface will be zero, by definition. In that case, $\nabla^2 \mathbf{u}$ will be zero on the surface, and we have eliminated one of the terms in the error. In fact, since the velocity gradient will never be exactly linear across the boundary, we also include an estimate of the Laplacian in the force term, using the velocity field at the present timestep.

Figure 2 shows how this forcing scheme will naturally reproduce the internal flow corresponding to a stagnation point on a curved surface. The (local) specification of the forcing to preserve normal velocity and reverse normal velocity at the numbered pairs of flow reversal points is sufficient to reproduce the no-slip surface and the internal flow corresponding stagnation point flow.

Note that this is the same boundary condition prescribed by Harlow and Welch (1965) and is divergence free, thus ensuring compatibility with the velocity-vorticity formulation of the velocity field. This choice of forcing may also be viewed as introducing vorticity at the immersed surface to mimic the presence of a solid, no-slip boundary. In this sense the method may be viewed as a grid-based vortex method. However, it is also distinct from most vortex methods since it requires only local information to calculate the value of the force. The effect of the forcing is to generate an internal boundary layer within the body, in the reverse direction to the external flow. One consequence of this method is the generation of internal flow fields which are comparable to the local external velocities. When we deal with three-dimensional bodies with significant surface curvature, this can lead to internal

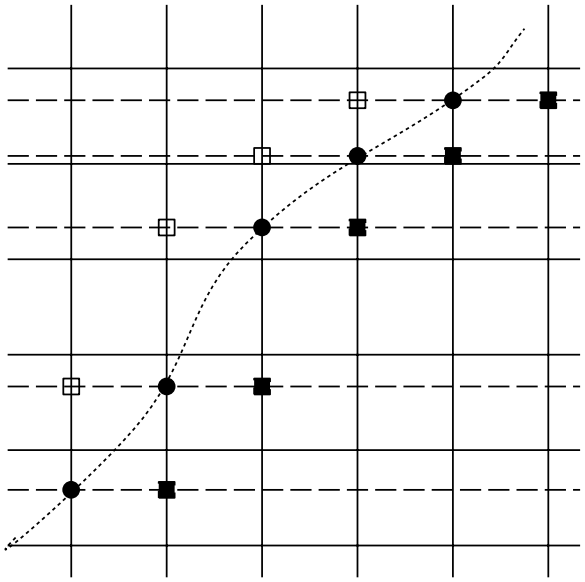


FIGURE 3. Sketch showing the relationship between the desired surface location (·····), the base computational grid (———) and the B-spline planes used for force calculation (– – – –). The surface forcing points (•), internal forcing points (■), and external velocity interpolation points (□) are also shown. Note that the force calculation planes need not be evenly spaced.

velocities which exceed the free-stream values. This can present a problem either by causing the timestep to be reduced (in a variable-time-step computational scheme), or by exceeding the CFL number stability limits (in a fixed time-step scheme). In our simulations, the effect on the timestep was not observed.

2.1.1 Floating forcing mesh

To avoid the error incurred by interpolating between grid points in the Fourier directions, we interpolate data across B-spline planes to ensure that the surface sampling points (solid circles in Fig. 3) always coincide with Fourier collocation points. Since the B-splines have compact support in physical space, it is economical to perform the direct inversion of the transform to obtain the velocity field on any desired $x - z$ plane. Note that the planes on which the forcing function is applied (dashed horizontal lines) need not necessarily coincide with the planes on which the flow is computed (solid horizontal lines). The reversal of tangential velocities is accomplished by forcing at points inside the surface (solid squares), where the velocity is forced to be the reverse of the velocity measured at the corresponding points in the external flow (dashed squares).

2.2 Laminar ribbed channel

For a test of the scheme we simulate laminar flow in a ribbed channel. The geometry of the problem is shown in Fig. 4. We consider a flow between two ribbed walls with periodic boundary conditions in the streamwise and spanwise directions. As in the case of CMK, we keep the channel cross-section and pressure gradient

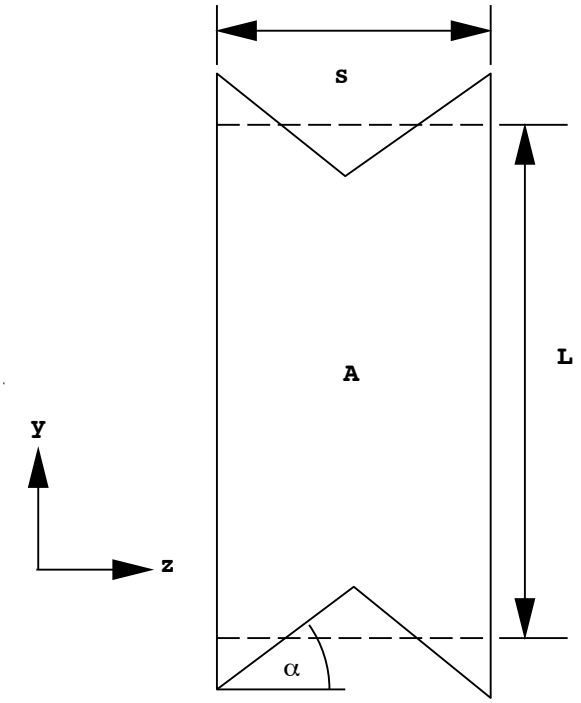


FIGURE 4. Sketch of the cross-sectional geometry of the channel. The mean channel height L is kept fixed while the riblet angle α is varied.

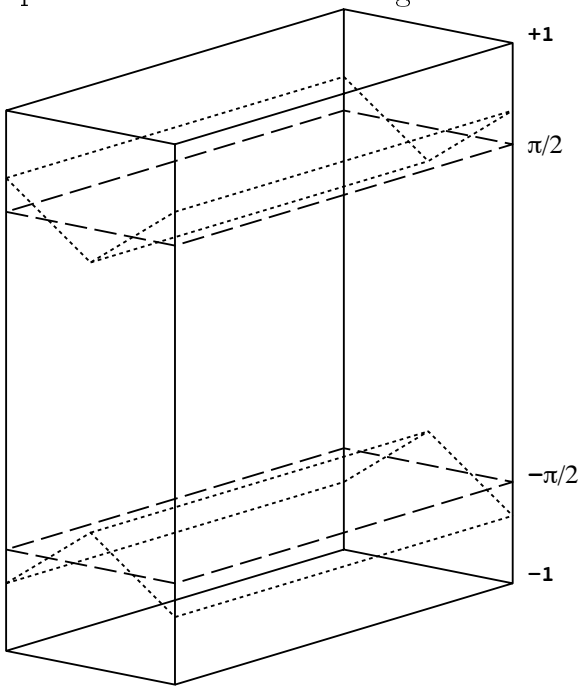
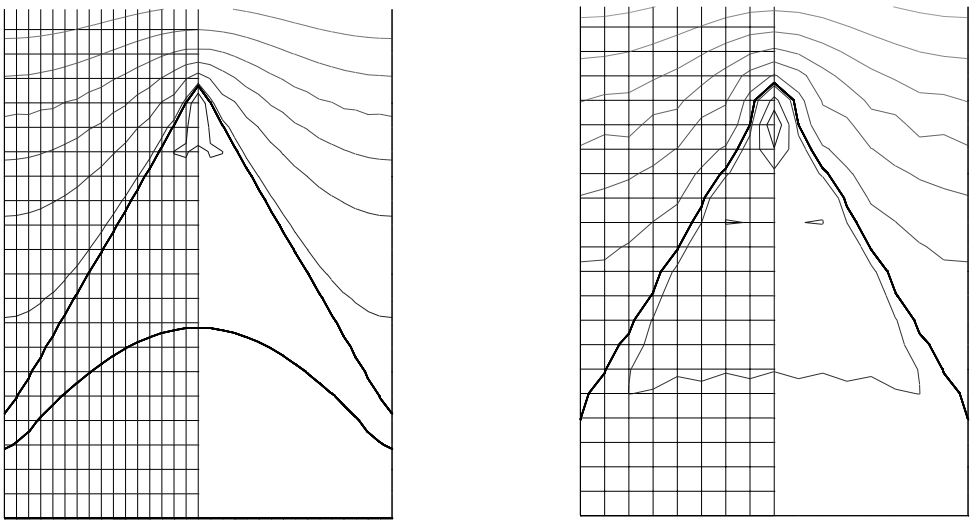


FIGURE 5. Sketch showing the computational domain (solid lines) and the location of the immersed boundary (dotted line). The mean location of the ribbed wall (corresponding to the flat channel case) is shown by the dashed line.



(a) *(b)*

FIGURE 6. Isosurface of zero streamwise velocity and contours of streamwise velocity for the 60^o riblet case, using 3rd-order B-splines. (a) 32 spanwise grid points (b) 16 spanwise grid points.

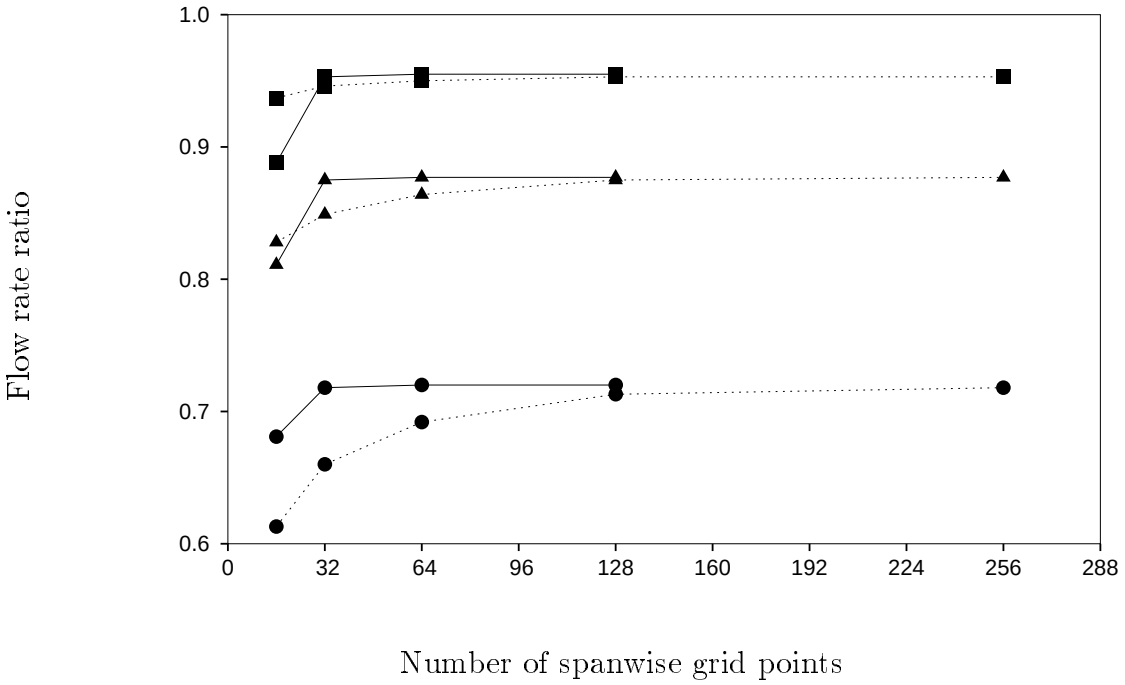


FIGURE 7. Variation of ratio of flow rates with riblets to flow rate in flat wall channel as a function of spanwise resolution. $S/L = 0.2$. —— current scheme ····· CMK results, ■ 30^o, ▲ 45^o, ● 60^o riblets.

constant and monitor the variation of the steady-state flow rate with the riblet geometry. CMK show that in the laminar case, the variation of the flow rate with geometry is independent of the Reynolds number and thus the specific choice of Re does not affect the results.

The computational domain extends from $-1 < y < +1$. No-slip conditions are applied to the computational boundary. The immersed riblet surfaces are centered at $y = \pm\pi/4$. Thus the height of the channel is $\pi/2$. The streamwise extent of the computational domain is fixed at $\pi/2$ for all simulations. A sketch of the computational domain and the location of the immersed boundary is given in Fig. 5.

2.2.1 Results

Figure 6 shows the isosurface of zero streamwise velocity for the case $\theta = 60^o$, $S/L_y = 0.2$, using 3rd-order B-splines. Note that the sharp tip of the riblet is well captured, even though only 32 spanwise grid points are used (Fig. 6a). A comparison is also shown with the case using only 16 spanwise modes (Fig. 6b). The rounding of the riblet tip can be seen in this instance and also the oscillations in the velocity contours, due to inadequate spanwise resolution.

Figure 7 shows the comparison of computed flow rate ratios for the 3rd order B-spline results with those of CMK. Note that the current method only requires 32 grid points per riblet in the spanwise directed as compared to 128 for the previous code. Note also that since there is no grid-deformation in the current scheme, the riblet angle does not affect the relative error to a large degree. In contrast, the highly non-orthogonal grids required in the boundary-fitted grid method cause much larger errors for the higher riblet angles.

Future plans

The numerical scheme will be extended to allow moving solid boundaries. Due to the expense of calculating the B-spline coefficients at each timestep, it may be necessary to utilize high order interpolation schemes to obtain field information at points that do not coincide with the solution collocation points. Such methods are readily available and are used in vortex schemes, for example, during remeshing steps. The resulting moving-boundary code will be used to test open and closed-loop control schemes (Koumoutsakos 1996) to minimize wall drag in turbulent flows.

REFERENCES

Carlson, H. A., Berkooz G. & Lumley J. L. 1995 Direct numerical simulation of flow in a channel with complex, time-dependent wall geometries: A pseudospectral method. *J. Comp. Phys.* **121,** 155-175.

Choi, H., Moin, P., & Kim, J. 1991 On the effect of riblets in fully developed laminar channel flows. *Phys. Fluids A.* **3,** 1892-1896.

Goldstein, D., Handler R. & Sirovich L. 1993 Modeling a no-slip flow boundary with an external force field. *J. Comp. Phys.* **105,** 354-366.

HARLOW, F. H. AND WELCH, J. E. 1965 Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface. *Phys. Fluids.* **8,** 2182-2189.

KIM, J., MOIN, P. & MOSER, R. 1987 Turbulence statistics in fully developed channel flow at low Reynolds number. *J. Fluid Mech.* **177,** 133-166.

KOUMOUTSAKOS, P. 1995 Fast multipole methods for three-dimensional N-body problems. *Annual Research Briefs 1995*, Center for Turbulence Research, NASA Ames/Stanford Univ., 377-390.

KOUMOUTSAKOS, P. 1996 A new method for the adaptive control of vortex-wall interactions. *Annual Research Briefs 1996*, Center for Turbulence Research, NASA Ames/Stanford Univ., 165-181.

KRAVCHENKO, A. G., MOIN, P. & MOSER, R. 1996 Zonal embedded grids for numerical simulations of wall-bounded turbulent flows. *J. Comp. Phys.* **127,** 412-423.

MOHD-YUSOF, J. 1996 Interaction of massive particles with turbulence. *PhD thesis*, Cornell University.