INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2001; **35**: 125–150

Handling complex boundaries on a Cartesian grid using surface singularities

Johan Revstedt*,[1] and Laszlo Fuchs

Division of Fluid Mechanics, Lund Institute of Technology, Lund, Sweden

SUMMARY

This paper considers flow around arbitrarily shaped objects. The boundary conditions on the solid boundaries have been applied by replacing the boundary with a surface force distribution on the surface, such that the required boundary conditions are satisfied. The velocity on the boundary is determined by interpolation or by local (Gaussian space) average. The source terms are determined iteratively as part of the solution. They are then averaged and are smoothed out to nearby computational grid points. The method has been applied both to test problems as well as to more complex engineering problems, where there are not many real competitive alternatives to the proposed method. Simulations of creeping flow around a sphere were studied in order to evaluate the performance of different, competitive approaches of imposing boundary conditions. Using local averaging first-order accuracy is obtained; this can be improved by using a Lagrangian polynomial instead, although the convergence is then considerably slower. Simulations of flows around spheres in the Reynolds number range 1–1000 have been carried out. Finally, the approach was used to describe the impellers in a turbine agitated mixer. For these cases, the results show overall good agreement with other computational and experimental results. Copyright © 2001 John Wiley & Sons, Ltd.

KEY WORDS: Cartesian grid; stirred reactor; virtual boundary method

1. INTRODUCTION

In most practical computational fluid dynamics (CFD) applications, the geometries of the solid boundaries are often very complex, e.g. wing sections, turbine cascades, bifurcating pipes, etc. The usual approach taken when simulating such flows is using either finite difference or finite volume methods, with a body fitted co-ordinate system and associated blockwise structured grids. Alternatively, one may use finite volume/element methods on unstructured grids. However, finite volume methods can only, with a considerable amount of work, be used with a discretization accuracy higher than second order (e.g. Fletcher [1]). Higher-order discretization

---

* Correspondence to: Heat and Power Engineering/Fluid Mechanics, Lund Institute of Technology, PO Box 118, S-221 00 Lund, Sweden. Tel.: +46 46 2224302; fax: +46 46 2224717.

[1] E-mail: johan.revstedt@vok.lth.se

Received July 1999
Revised January 2000

is almost always needed for attaining good accuracy without the need for very fine spatial resolution. In some cases, higher-order methods are required due to the difficulty to highly resolve the scales of the flow themselves. This is the situation when direct numerical simulation (DNS) is sought for as high a Reynolds number as possible. The situation is similar for large eddy simulations (LES), since the resolved scales are of the order of the grid size. Generalized co-ordinates require that the co-ordinate transform matrices (Jacobians) have to be stored or be recalculated. In addition to this requirement, by co-ordinate transformation, additional terms appear in the governing equations. These terms imply a considerable increase in the number of computational operations. Furthermore, non-uniformities in the grid cause a slowdown in the convergence rate of many iterative solvers. Also, the grid generation has to be carried out with great care, as degenerate computational cells can, at least locally, dramatically increase the numerical errors. A further aspect not very often considered is that the grid that is generated *a priori* is seldom optimal for the particular problem that is being solved.

Using Cartesian grids, the structure of the solver is substantially simplified and the implementation of higher-order discretization is straightforward. However, the classical way of describing complex boundaries on such grids (full and partial blocking of computational cells) leads to low-order accuracy near the boundaries. This problem can be treated in some ways. For example Gullbrand *et al.* [2] used a method in which the boundary conditions are interpolated to the right position in the domain leading to improved accuracy. There are some potential drawbacks with this approach, primarily when the resolution is not adequate. One could think of using moving grids (fixed to the boundaries). This approach would imply that a grid has to be generated in each time step. The grid has also a tendency to be skewed and/or with a largely variable aspect ratio. These two factors imply a reduction of computational accuracy and efficiency. One also has to account for an additional condition (on the grid) so as to satisfy mass conservation [3].

Another alternative to the above-mentioned approaches is to use overlapping grids. By this technique, one may generate, independently, a grid around different components of the problem so that the union of all grids 'cover' the full domain of interest. If the grids are imposed to match each other, one obtains patched grids. By removing this constrain and by allowing the grids to overlap each other one may solve the flow past arbitrarily moving objects without the need to regenerate new grids. This technique started to evolve in the mid-1980s for two-dimensional problems (e.g. Fuchs [4]) and was extended to the three-dimensional case in the 1990s (e.g. Tu and Fuchs [5]).

The overlapping grid (also known as the 'chimera' grid) approach is a good alternative for cases where the boundaries are not affected by the fluid. For cases where the boundary conditions are determined by the interaction between the object and the surrounding flow (and fluid), the current approach is preferable. For the sake of comparison, however, we consider here only solid objects.

The use of momentum sources to describe boundaries, also referred to as 'fictitious domain' methods, can be traced to different early attempts in the 1970s by Peskin [6]. He applied the basic approach for studying the flow of blood past an elastic (initially two-dimensional) heart-valve leaflet. Since then different variants of this family of methods has been proposed. Some particularly active individuals in the field have been Peskin [6–8], Zaleski [9,10], Trygvasson [11] and Shyy [12,13]. This short list is not exclusive, and further references may be found in the references given herein. The main differences among the different approaches lie in the implementation of the technique rather than on the basic principles. Thus, the basic approach is generally the same but methods for determining the magnitude of the sources differ. The idea is to determine the effect an object has on the fluid by adding source terms to the momentum equations. Glowinski *et al.* [14–16] studied flow around cylindrical shapes using a finite element approach. Two-dimensional flow in pipes and cavities, with Reynolds numbers up to 1000, were studied by Ionkin and Churbanov [17]. Saiki and Biringen [18] obtained results that correlate well with measurements for flows around cylinders, both stationary and oscillating, using the method by Goldstein *et al.* [19]. The approach we use differs from the method by Goldstein *et al.* [19] in the way the body forces are computed. It is believed that our approach has the potential to incorporate into a multi-grid solver. The approach allows also locally refined girds to be combined with the body force method.

All of the studies mentioned above concern primarily two-dimensional flow or low Reynolds numbers or both. The goals of this work have been to (a) propose a new algorithm for determining the body forces, (b) compare the accuracy of the basic method with two alternative approaches, (c) apply different types of boundary conditions and their interaction with the local spatial resolution, (d) demonstrate the applicability of the method, (e) study the applicability of the method for LES of high-Reynolds number flows. Here, we consider both stationary and moving three-dimensional objects at low as well as higher Reynolds number. We also consider the interplay between boundary conditions and the local spatial resolution. To determine the numerical order of accuracy of the method, we consider the Stokes flow around a sphere. We have also studied the case of an oscillating sphere and compared, and the motion of arbitrarily shaped blades placed in a cylindrical (non-axisymmetric) tank. For the last case we compare our results with experimental data from the literature.

2. MATHEMATICAL FORMULATION

2.1. Governing equations and boundary conditions

The equations governing isothermal, incompressible flow of a Newtonian fluid can be written as

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

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu \frac{\partial}{\partial x_j}\frac{\partial u_i}{\partial x_j} + F_i \tag{2}$$

where F_i is a source term.

The system of Equations (1) and (2) is well posed if the d-conditions are imposed on all the boundaries, where d is the dimension of the problem. In addition, one must also satisfy global mass conservation. In the case of solid walls, one applies usually the no-slip condition, which implies that the local fluid velocity equals the velocity of the boundary at the same point. For non-solid object boundaries, the stresses have to be in balance. This type of boundary condition can be given implicitly and hence has to be determined as part of the solution. At inflow and outflow, the velocity vector and its gradient respectively are often assumed. The same number of conditions has to be imposed, even when other expressions are used. Other types of conditions, such as for a boundary with surface tension, could be applied equally.

The body forces F_i in our cases vanish normally. However, we shall replace some of the boundaries by a force distribution on the surface of the boundary. In this case, the body forces F_i are computed on the boundaries so as to satisfy the local boundary conditions. Thus, F_i do not vanish only on certain surfaces. This singularity has, of course, an impact on the numerics involved in solving the equations.

2.2. Turbulent flows

To model the turbulent flows presented here we have used LES. LES is based on spatial filtering of the equations of motion rather than on the time averaging used in traditional turbulence modelling. The space filtering of a function $f(x_i, t)$ is defined as

$$\overline{f(x_i, t)} = \int_{-\infty}^{\infty} G(x_i - x_i')f(x_i', t)\,\mathrm{d}x_i' \tag{3}$$

where G is a filter function.

The space filtered equations for the conservation of mass and momentum for an incompressible Newtonian fluid can be written, using summation convention, as

$$\frac{\partial \overline{u_i}}{\partial x_i} = 0 \tag{4}$$

$$\frac{\partial \overline{u_i}}{\partial t} + \overline{u_j}\frac{\partial \overline{u_i}}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \bar{p}}{\partial x_i} + \nu\frac{\partial}{\partial x_j}\frac{\partial \overline{u_i}}{\partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} + \overline{F_i} \tag{5}$$

where

$$\tau_{ij} = \overline{u_i u_j} - \overline{u_i}\,\overline{u_j}$$

where $\overline{F_i}$ is a source term which, in this work, is used to invoke the effects of solid boundaries. The term τ_{ij} is the sub-grid scale (SGS) stress tensor, which reflects the effect of the unresolved scales on the resolved scales. This term is often expressed as a function of the filtered rate of strain, e.g. the Smagorinsky model [20]. In this paper, however, an implicit model is used, i.e. the numerical truncation error acts as an SGS term. The rationale of using implicit models is as follows: LES requires high numerical resolution such that the flow quantities under consideration have an asymptotic behaviour (which does not imply that all quantities have similar properties). In a turbulent field, which is more or less is in equilibrium, there exists an energy cascade by which energy is extracted from the large-scale motion through smaller and smaller scales. At the smallest scales (for which the viscous and the inertial forces balance each other), the kinetic dissipates into heat. Any SGS model must account for these dissipative effects. Without this role, there will be a continuous build-up of kinetic energy at the smallest resolved scales. Therefore, neglecting this effect would not lead to a catastrophic result. SGS models also have other roles (such as accounting for the interaction between the unresolved and resolved scales). Also, having adequate spatial resolution implies that the SGS has to approach asymptotically a monotone behaviour. In particular, if the spatial resolution improves, the SGS model would be less and less important (it should vanish, or be of the order of the truncation errors for DNS). The truncation errors of the discrete system act dissipatively. These terms do not have any further physical role so as to account for SGS interactions. Among these roles one may mention the effects of the unresolved scales on the resolved one in the average and instantaneous sense (inducing back-scatter). This part of the SGS can be handled explicitly by different SGS models, such as 'dynamic' models (e.g. Held and Fuchs [21] and Olsson and Fuchs [22,23]).

In several of the cases presented here, the flow is laminar. The above discussion should, therefore, be limited to the cases where the wake is turbulent. In laminar flows all the scales of the flow are resolved and the filter function G in Equation (3) is identical to the Dirac δ function. This statement is correct on the differential equation level. However, all discrete approximations to the continuous variables imply filtering [24]. Thus, a consistent way of handling the singular force terms is by considering their spatial average, even in laminar (or DNS) cases.

3. NUMERICAL METHODS

3.1. Flow solver

The incompressible Navier–Stokes equations are discretized on a system of locally refined Cartesian grids (e.g. Fuchs and Zhao [25]). The dependent variables are defined on a staggered grid. This arrangement has the advantage that the system requires three boundary conditions (e.g. the velocity vector) on all boundary points. The different terms of the momentum and continuity equations are approximated by finite differences. Basically, one may use finite differences of any order. However, for higher orders (i.e. more than two) additional boundary conditions have to be specified. For some of the Reynolds numbers under consideration, the grid is unable to resolve the smallest scales, and hence one has to add 'numerical viscosity' so as to adjust the local scale to the grid size. This can be achieved directly (i.e. implicitly) by using an 'upwind' finite difference scheme for the convective terms. Here, we use upwind finite differences of first- or third-order accuracy. The lower-order scheme (first-order for the convective terms and second-order for the others) implies that the low-order terms dominate, leading to a high level of numerical dissipation. Using directly higher-order (third- and fourth-order approximations respectively) leads to a less robust solver with a considerably slower convergence rate. To combine numerical efficiency with higher-order accuracy, we introduce the higher-order terms as a 'single-step' defect correction [26]. One can show that, for smooth problems, this procedure is adequate to maintain the theoretical accuracy of the high-order scheme.

The time integration is done by a three-level implicit scheme. In each time step, the system of equations is solved iteratively using a multi-grid solver. The relaxation scheme within the multi-grid solver comprises of pointwise relaxation of the momentum equations coupled with a pointwise smoothing of the continuity equation. At the latter step, both the velocity vector and the pressure are corrected so that the residuals of the momentum equations shall not be changed as the continuity equation is satisfied. This approach is equivalent to an approximate diagonalization of the system of equations [25,26].

The basic numerical procedure of the technique used here is composed of the following steps:

1. Spatial discretization of the boundary surface.
2. Determine the boundary conditions (using the dependent variables on the full computational grid): 'interpolation' step.
3. Determine the defect to the surface force.
4. Distribute the surface force to the surrounding space.

Items 1–4 may be done in different ways, which of course has an effect on the numerical accuracy and efficiency of the scheme. In the following we elaborate some of the options that we have employed.

3.2. Boundary conditions

Objects are modelled by adding momentum sources to the Navier–Stokes equations (5). The magnitude of these sources is determined iteratively based on the defect in satisfying the boundary conditions. First, the surface of the boundary is discretized. On these nodes the desired boundary conditions have to be satisfied. Of course, the surface nodal points do not in general coincide with the discretization of the (three-dimensional) computational domain. Thus, in order to satisfy the boundary conditions, the appropriate variables have to be found on the surface nodal points. For a solid object, we apply no-slip conditions. For a stationary object, this implies that the velocity on the surface should vanish. This velocity, at the discrete surface points has to be determined by some interpolation to those points. For non-solid objects, stress boundary conditions may be specified. In such situations, the variables that are computed on the boundary surface will be the stresses themselves and not the velocity components as in the previous case.

As stated above, after discretizing the boundary surface, one has to determine if the boundary conditions are satisfied. Information has to be transferred from the computational grid to the surface grid. This can be achieved by different types of interpolations. It should be noted that it is not self evident that a higher-order interpolation scheme is to be preferred. For example, for a solid object, the velocity has to vanish on the surface. When the object is replaced by a body force, the velocity field shall have piecewise continuous gradients. The jump in the stress is directly related to the strength of the force distribution. In this case, a higher-order polynomial interpolation may result in oscillatory behaviour (Gibb's effect). For this reason we have studied several options for interpolating data to the boundary surface. The fluid velocity (u_i^f) in surface nodes has to be determined. In this work we have studied two ways of doing this. The most straightforward approach is using the Lagrange interpolation formula (in multiple dimensions). The procedure was adopted from a three-dimensional Lagrangian interpolation algorithm by Ericsson and Fuchs [27]

$$u_i^{\mathrm{f}} = \sum_{r=1}^{n+1} \sum_{s=1}^{n+1} \sum_{t=1}^{n+1} \left(\prod_{k=1_{k \neq r}}^{n+1} \frac{\xi - \xi_k}{\xi_r - \xi_k} \right) \left(\prod_{k=1_{m \neq s}}^{n+1} \frac{\eta - \eta_m}{\eta_s - \eta_m} \right) \left(\prod_{k=1_{p \neq t}}^{n+1} \frac{\zeta - \zeta_p}{\zeta_t - \zeta_p} \right) u_{i_{rst}} \tag{6}$$

The second approach is to average the velocities on the given computational grids. One may argue that if one would employ the 'interpolation' and 'distribution' operators that are the inverse of each other (in some sense) then there would be minimal error committed in these operations. By this argument, if one would use a Gaussian 'distribution' step, one could argue for using also a Gaussian averaging for the 'interpolation' step. Thus, the Gaussian interpolation scheme for the boundary velocity (u_i^{f}) would be

$$u_i^{\mathrm{f}} = \int_{\xi-1}^{\xi+2} \int_{\eta-2}^{\eta+1} \int_{\zeta-2}^{\zeta+2} u_i G_F \, \mathrm{d}\xi \, \mathrm{d}\eta \, \mathrm{d}\zeta \tag{7}$$

where ξ_j is the boundary position normalized with the mesh spacing and G_F is a Gaussian distribution function

$$G_F = \frac{1}{(\sigma\sqrt{2\pi})^3} \exp\left(\frac{-(\xi^2 + \eta^2 + \zeta^2)}{2\sigma^2} \right) \tag{8}$$

The variance, σ, is of the order of the computational grid size.

The next step in the algorithm (Step 3) is to compute the components of F_i, such that the boundary conditions are satisfied. For the solid (stationary object) this implies that (u_i^{f}) vanishes. We argue that, near the solid boundary, the dominating term is the viscous term in the momentum equations. Thus, a change (error) in the boundary velocity (δu_i^{f}) has to be balanced by a change in the body force $\Delta \tilde{F}_i$ on the surface. This implies that

$$\tilde{F}_i^n = \tilde{F}_i^{n-1} + \Delta F_i^n \tag{9}$$

$$\Delta F_i^n = \alpha \frac{u_i^{\mathrm{f}} - u_i^{\mathrm{b}}}{h^2} \tag{10}$$

where u_i^{b} is the specified boundary velocity, n is the iteration number (not to be confused with the time step), $\Delta F_i = \tilde{F}_i^{n-1}$ and α is a constant added to ensure both numerical stability as well as optimal convergence.

The final step of the algorithm (Step 4) is to 'distribute' the surface forces $\tilde{F}_i$ to the computational domain F_i. The consistent way of distribution would be if the surface δ function is averaged (distributed) in the same way as the discrete approximation. Thus, for second-order finite differences one could use a spatial 'top-hat' function. On the other hand, as it is argued above, one would like to have 'interpolation' and 'distribution' operators that are the inverses on each other. For the time being we use, therefore, the Gaussian distribution (7) and (8).

4. FLOW AROUND A SPHERICAL OBJECT

4.1. Creeping flow—Stokes' solution

Consider creeping fluid motion ($Re \ll 1$) around a solid spherical object. For this particular flow case, an exact solution was first derived by Stokes [28]. Using the spherical polar co-ordinate system shown in Figure 1, the velocity components can be written as

$$u_r = U\cos\theta\left(1 + \frac{a^3}{2r^3} - \frac{3a}{2r}\right) \tag{11}$$

$$u_\theta = U\sin\theta\left(-1 + \frac{a^3}{4r^3} - \frac{3a}{4r}\right) \tag{12}$$

where a is the sphere radius and U is the free stream velocity.

In order to achieve acceptable numerical accuracy, the boundary velocity must be determined in a proper manner. Two approaches have been studied in this work: Lagrangian interpolation from values outside the boundary (6) and spatial averaging, Equations (7) and (8), with a Gaussian weight function using velocity values on both sides of the boundary. The properties of these approaches were studied by using Stokes' exact solution for the velocities. The computational domain (shown in Figure 1) is a cubic box with the side length $2D_{sp}$, centred in this box is a spherical body with diameter D_{sp}. Four levels of grid resolutions are used with 16^3, 32^3 and 128^3 grid points respectively. We compare the accuracy of the following methods.

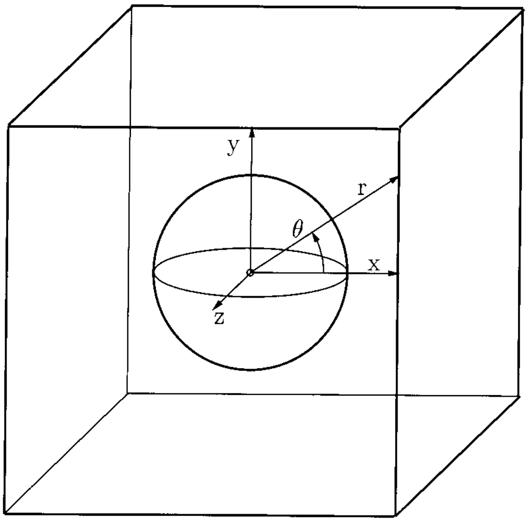

Figure 1. Geometry and co-ordinate systems of the Stokes flow simulations.

(a) Computational cells that belong to a solid object are excluded, i.e. not updated by the solver. This is referred to as 'blocking'.
(b) The 'force' method with Gaussian 'interpolation' (using Equations (7) and (8)). This method is referred to as the 'Gaussian' method.
(c) The 'force' method with Lagrange interpolation formulation (Equation (6)). This is denoted the 'Lagrangian' method in the following.

The different computed cases are given in Table I. The results with the different methods have been compared with the exact solution (Equation (12)). The exact Stokes solution has been applied on the outer boundaries and the Reynolds number has been set to 0.01 in all the simulations.

4.1.1. Numerical accuracy. The Gaussian method using $5 \times 5 \times 5$ cells cube (Case 1) was compared with the Lagrangian method using $2 \times 2 \times 2$ points (Case 2) and $3 \times 3 \times 3$ points (Case 3) respectively. Figure 2 shows maximum- and L_2-norms of the error in boundary velocity. As can be seen in this figure, Case 3 results in a lower error level than the other two methods, especially at higher resolutions.

The maximum- and L_2-norms of the errors in the velocity components in the x- and y-directions are plotted in Figure 3. It is evident that cell-'blocking' gives better performance than the Gaussian averaging. However, the interpolation approach gives over all the smallest errors and the solution order is higher (Table II).

Covergence histories of the boundary velocity for the Gaussian and Lagrangian approaches using 64^3 grid points are shown in Figure 4. The Gaussian approach exhibits faster convergence than the Lagrangian and it is also computationally less expensive. It is also clear that the convergence rate is far from optimal. For time-dependent flows, in which one has to take small time-steps (for any reason), the convergence rate of the basic method is not a serious problem. In other situations, the convergence rate of the basic method (cf. Goldstein *et al.* [18]) is also poor. In Section 7.1 we outline the multi-grid implementation of our method which we believe will remedy the slow convergence of the basic algorithm.

Table I. Computed cases of creeping flow past a sphere.

Grid size	Boundary method
16^3	Blocking
36^3	Blocking
64^3	Blocking
128^3	Blocking
16^3	Gaussian
32^3	Gaussian
64^3	Gaussian
128^3	Gaussian
16^3	Lagrangian
32^3	Lagrangian
64^3	Lagrangian

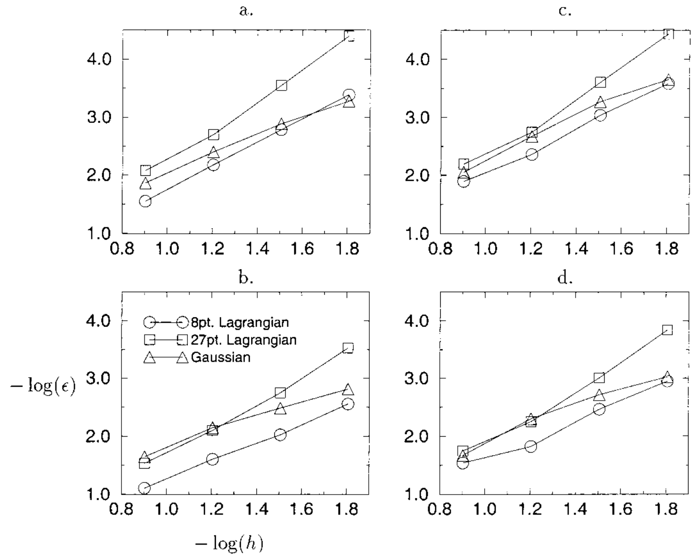

Figure 2. Error in boundary velocity using the Stokes' velocity field; (a) L_2-norm x-direction, (b) max-norm x-direction, (c) L_2-norm y-direction, (d) max-norm y-direction.

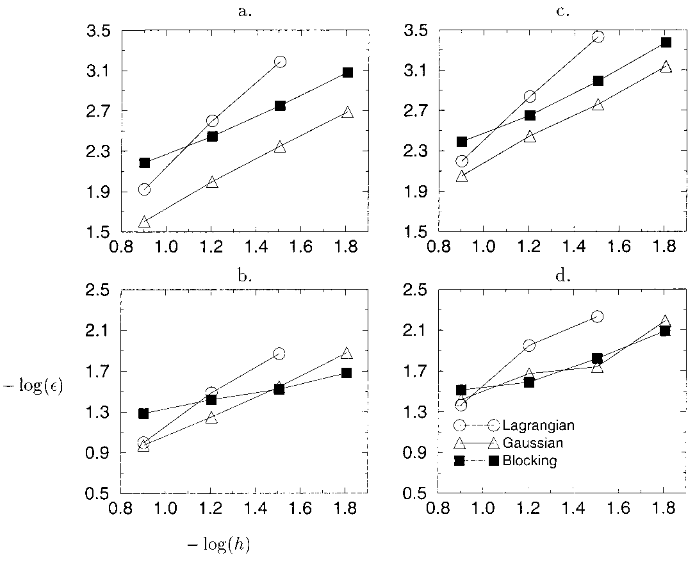

Figure 3. Maximum- and L_2-norms of velocity errors; (a) L_2-norm x-direction, (b) max-norm x-direction, (c) L_2-norm y-direction, (d) max-norm y-direction.

Table II. Solution order.

Boundary method	U		V	
	Max-norm	L_2-norm	Max-norm	L_2-norm
Blocking	0.44	0.99	0.64	1.09
Gaussian	1.00	1.20	0.85	1.20
Lagrangian	1.45	2.10	1.43	2.05

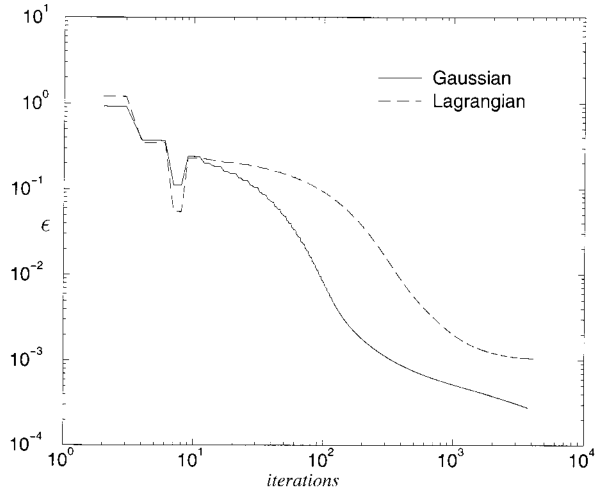

Figure 4. Convergence history (L_2-norm) of the boundary velocity for the Lagrangian and the Gaussian approaches.

4.2. *Low-Reynolds number flow*

Simulations of the flow around a sphere in the Reynolds number range 1–1000 have been performed. The Gaussian interpolation approach was used since it is considerably faster than the Lagrangian one. However, for $Re = 100$, the other methods were also computed for comparative reasons. The wake length and drag coefficient were calculated and compared with measurement data and results from calculations by Shirayama [29] and Pruppacher *et al.* [30]. The geometry and the extent of the local grids are shown in Figure 5. The size of the computational domain is $24D_{sp} \times 16D_{sp} \times 16D_{sp}$. Two global multi-grid levels were used in addition to the four levels of local refinements around the sphere. The size of the grids and the grid resolution are presented in Table III. It has to be pointed out that the three-digit 'accuracy' of the calculated recirculation length is a sub-grid accuracy obtained by interpolation and therefore the last digits contain the effects of several numerical errors.

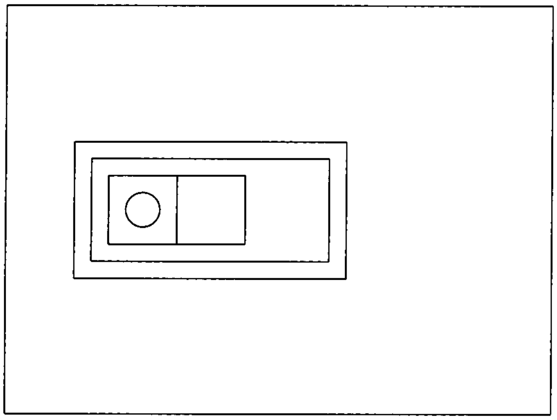

Figure 5. A schematic representation of the locally refined grid system.

Table III. Grid configuration.

Grid number	Type	Size	Number of grid points
1	Global	$24D_{sp}\times 12D_{sp}\times 12D_{sp}$	$24\times 12\times 12$
2	Global	$24D_{sp}\times 12D_{sp}\times 12D_{sp}$	$48\times 24\times 24$
3	Local	$8D_{sp}\times 4D_{sp}\times 4D_{sp}$	$32\times 16\times 16$
4	Local	$7D_{sp}\times 3D_{sp}\times 3D_{sp}$	$56\times 24\times 24$
5	Local	$4D_{sp}\times 2D_{sp}\times 2D_{sp}$	$64\times 32\times 32$
6	Local	$4D_{sp}\times 2D_{sp}\times 2D_{sp}$	$64\times 64\times 64$

4.2.1. Results. The velocity field at $Re = 100$ is shown in Figure 6, note the symmetric recirculation zone. This zone is reported to first appear at a Reynolds number of about 20 and it remains stable up to about $Re = 275$ [29]. Hence, for $Re = 1000$, the flow in the wake lacks symmetry and tends to be turbulent (Figure 7(a)). However, in a time-averaged sense, the flow is still symmetric as can be seen in Figure 7(b). Comparing the length of the recirculation zone for some Reynolds numbers and approaches to describing the body fairly good agreement with results by Shirayama [29] and Pruppacher *et al.* [30] is obtained; although our results show a somewhat shorter recirculation zone. Also, the length is only marginally affected by the type of boundary description (Table IV).

The drag on the sphere ($\mathscr{F}_D$) has been studied in terms of drag coefficient, $C_D = (8\mathscr{F}_D)/(\rho U^2\pi D_{sp}^2)$. To calculate the force one may study momentum balance in a control volume Ω around the sphere. By integrating the steady Navier–Stokes equations in this volume, the force can be obtained

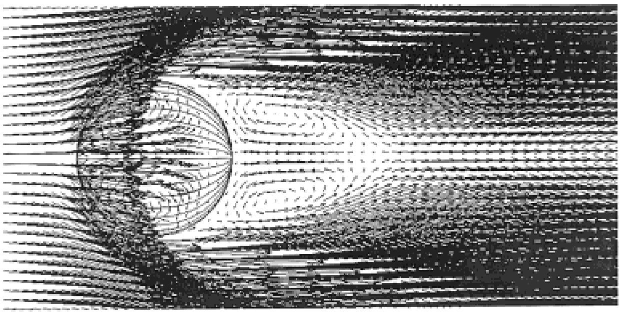

Figure 6. Velocity vectors at the sphere centre plane ($Re = 100$).

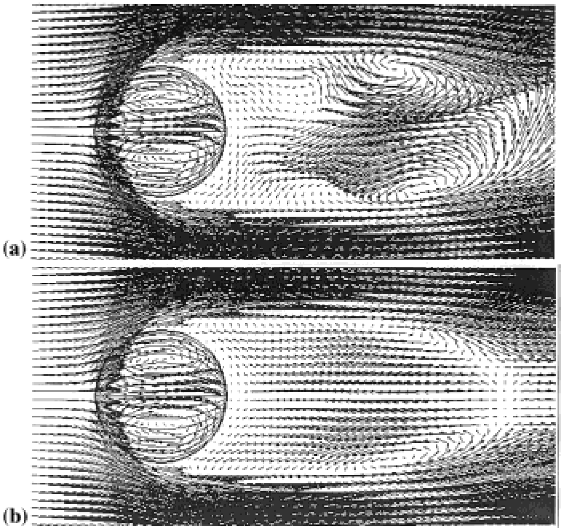

Figure 7. Velocity field at the sphere centre plane; (a) instantaneous, (b) time-averaged ($Re = 1000$).

Table IV. Length of the recirculation zone normalized with the sphere diameter.

Re	Gaussian	Lagrangian	Blocking	Shirayama [29]	Pruppacher *et al.* [30]
50	0.386	—	—	0.384	—
100	0.766	0.714	0.736	0.844	0.93
1000	2.05	—	—	—	—

$$\mathscr{F}_i = \iiint_\Omega \left(\rho u_j \frac{\partial u_i}{\partial u_j} + \frac{\partial p}{\partial x_i} - \mu \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right) \mathrm{d}\mathscr{V} \tag{13}$$

Using Gauss' theorem, this can be rewritten as

$$\mathscr{F}_i = \iint_\Gamma \left(\rho u_i u_j n_j + p n_i \delta_{ij} - \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) n_j \right) \mathrm{d}\mathscr{S} \tag{14}$$

where Γ is the surface of the control volume and n is the direction normal to the surface.

This way of calculating the drag is quite straightforward. However, it is more convenient to compute the body forces directly by using the source terms already calculated. The force that the fluid exerts on the sphere is obtained by integrating the body forces over the sphere surface

$$\mathscr{F}_i = \iint_\gamma F_i \,\mathrm{d}\mathscr{S} \tag{15}$$

Table V shows that there is very little difference between the ways of calculating the drag; we have therefore used Equation (15) wherever possible. The discrepancy between the two methods can be attributed to numerical integration errors.

Comparing the drag coefficients obtained from the simulations using the source term approach at $Re = 100$ with the 'blocking' method, one sees no significant difference, nor is there any discrepancy with Shirayama's [29] simulations (Table VI). The trend is the same for other Reynolds numbers as can be seen in Figure 8. Figure 8 shows the drag coefficients for several Reynolds numbers compared with measurements presented by White [31] and the simulations by Shirayama [29].

Table V. Drag coefficient calculated with Equations (14) and (15).

Re	Equation (14)	Equation (15)
1	29.1	26.1
10	4.454	4.451
50	1.630	1.620
100	1.113	1.107

Table VI. Drag coefficient at $Re = 100$.

Re	C_D
Gaussian	1.107
Lagrangian	1.104
Blocking	1.108
Shirayama [27]	1.104

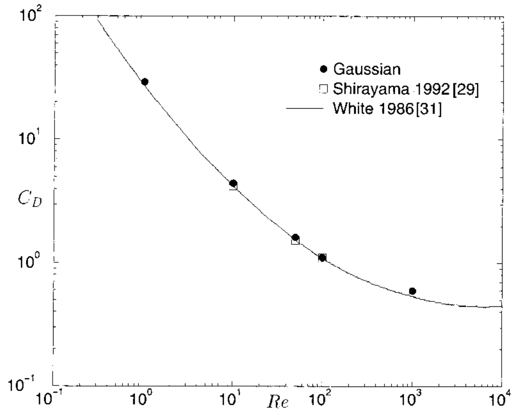

Figure 8. Drag coefficient at some Reynolds numbers computed by the Gaussian averaging scheme.

4.2.2. Moving boundaries. The advantages of using source terms over blocking of cells to describe the boundaries becomes evident if one considers a non-stationary boundary, i.e. a boundary moving relative to the main flow or even driving the main flow (e.g. a propeller). Consider a sphere moving perpendicular to a parallel flow, with a velocity of its centre given by

$$V_y = AU_0 \cos(\omega t) \tag{16}$$

where U_0 is the free stream velocity, $A = 0.15$ and $\omega = 0.6$.

Using blocking for this case, the grid will have to be regenerated at each instant and the moving of the blocked cells can give rise to discontinuities in the velocity gradients and consequently convergence problems. The source term approach will generate a much smoother motion, as the location of the boundary is independent of the grid. The error is also strongly periodic, which indicates that it is dependent on the magnitude of the boundary velocity. Figure 10(a)–(h) shows the sphere during the period of motion, note the asymmetric and lagging wake, as one would expect intuitively.

Next consider the errors in applying the no-slip boundary conditions. Since finite precision arithmetic is used the estimated error is $O(\epsilon_0/h^2)$, where ϵ_0 is the machine accuracy and h is the mesh spacing near the boundary. This relation stems from the viscous terms in the momentum equations. The convergence history to machine accuracy, for the stationary case, is plotted in Figure 9. The convergence rate is rather poor and further improvements in the algorithm are proposed in Section 7.1. For the time-dependent case, reduction in each time step is rather small (depending on the time step). However, when time step is not too large only a limited number of iterations are required.

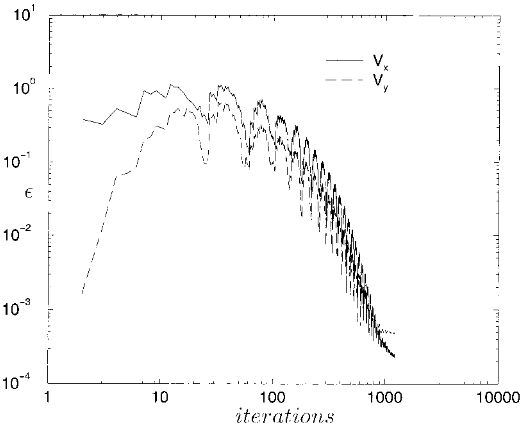

Figure 9. L_2-norm of errors in boundary velocity for a stationary sphere ($Re = 50$).

5. BOUNDARY CONDITIONS AND SPATIAL RESOLUTION

An often-encountered problem is the implementation of appropriate boundary conditions. This issue is less pronounced with body fitted grids, since for any viscous flow no-slip conditions are applied on solid walls. Using the method presented in this paper one might experience difficulties with imposing no-slip conditions if the size of the object in one direction is of the order of the mesh spacing, at least at high Reynolds numbers, which is the case for the impeller blades in Section 6. However, such boundary conditions are meaningful if the boundary layer associated with the viscous effects is resolved. If, on the other hand, the boundary layer is not fully resolved it is inconsistent using no-slip conditions. Under such circumstances one could equally apply slip conditions, as one would do directly when viscous effects are neglected. As we shall see, when we apply a singular force distribution, the 'thickness' of the force 'layer' is of the order of one mesh spacing for low Reynolds numbers. This situation may be improved, by special treatment so as to achieve a 'sub-grid' resolution of the singular force surface thickness (Section 7.2). With this aspect in mind we have considered both applying the no-slip conditions and alternatively slip conditions at a solid surface.

To demonstrate the difficulties with thin objects we first consider a well-resolved case: a thin stationary disc at $Re = 40$ with the same set-up as for the sphere (Figure 5). We apply either the usual no-slip conditions or slip conditions (which in this case means that the flow normal to the surface vanishes). Thus, with slip boundary conditions, the flow along both sides of the disc surface is in parallel. Figure 11(a) and (b) shows the velocity vector of flow around the disc no-slip and slip conditions respectively. The drag coefficients calculated with Equation (15) give $C_D = 1.97$ in the no-slip case and $C_D = 2.15$ in the slip case, as compared with 1.8, which is the value given in the literature (e.g. White [31]). Studying the velocity parallel to the disc surface one can see that, in order to achieve zero velocity on the surface, the velocities on

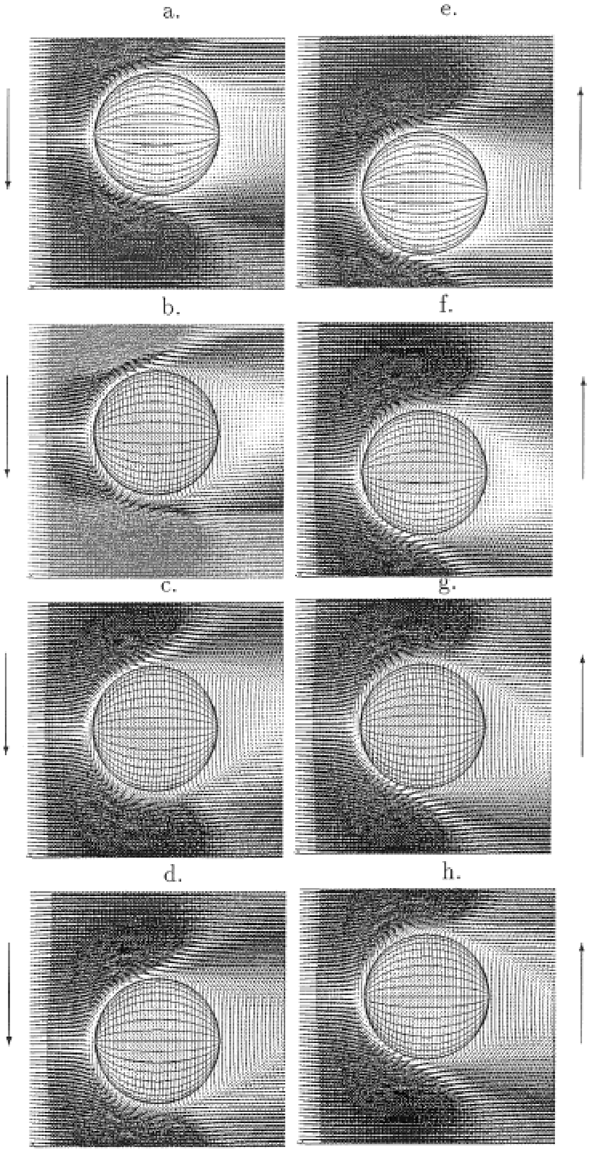

Figure 10. Velocity fields at several instants for a sphere performing a sinoudial motion perpendicular to the main flow ($Re = 50$). The arrows indicate the direction of motion.

either side of the surface must have opposite directions. This is clearly visible in Figure 12 and it will lead to a *de facto* thickness of the plate equal to the grid size. However, if we instead consider the high-Reynolds number case of a rotating impeller, the result will be quite different

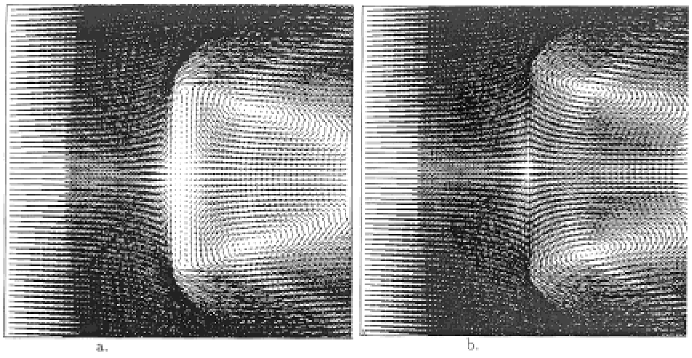

Figure 11. Velocity field at the disc centre plane ($Re = 40$); (a) no-slip, (b) slip.

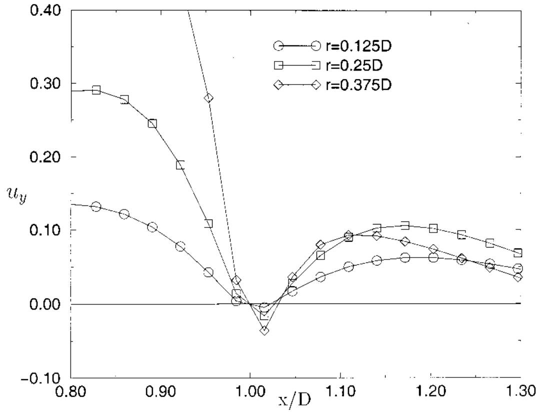

Figure 12. Velocity parallel to the disc at three radial distances, parallel to the axis of the disc. The disc is located at $x/D = 1$.

(as shown in Section 6) due to the fact that the boundary layer is not resolved. When, on the other hand, the thickness of the disc is resolved, there are no problems in implementing no-slip conditions.

6. IMPELLER STIRRED TANKS

The source term approach has also been applied on the description of the impeller in a stirred tank. This is a case with a relatively high Reynolds number and several moving boundaries.

6.1. Geometrical set-up

The simulations were performed in a water-filled cylindrical tank with a single six-bladed Rushton turbine (6RT) and a six-bladed Scaba turbine (6SRGT) (Figure 13; Table VII). The blades are centred both radially and axially in the tank. The geometrical outline and grid configuration of the tank are indicated in Figure 14. The outline is the same as the one used in the simulations by Revstedt *et al.* [32] and in the experimental work by Stoots and Calabrese [33]. Simulations were made at a turbine speed (N) of 2.75 rps, which correspond to a Reynolds number ($Re = (\rho\pi N D l_a)/\mu$) of 3800. The computational grid consists of three global multi-grid levels and two levels of local refinement in the impeller region. On the finest global level the grid consists of 32^3 cells. Here, we have only studied the flow in the immediate vicinity of the impeller, thereby justifying the coarse grid on the global level. The mesh spacing is then $0.023 \cdot D$ in the impeller region, which is in the order of the Taylor micro-scale as measured by Costes and Couderc [34].

Since the blade thicknesses are not resolved spatially, we assumed these to be grid independent (i.e. infinitely thin). Therefore, we are faced with the problem of applying no-slip conditions. Of course, no-slip conditions can be applied if the blade thickness are resolved or are assumed to be directly grid dependent. The later situation is believed to be more problematic as compared with using slip conditions (in this high-Reynolds number case). No-slip conditions are set on the tank walls and on the baffles. The cylindrical shape is obtained by blocking out cells. The hub and shaft of the turbine have been neglected.

6.2. Results

The velocity field relative to the impeller has been studied in order to get a qualitative as well as quantitative picture of our boundary description. The phase average of a function $g(\tau)$ is defined as

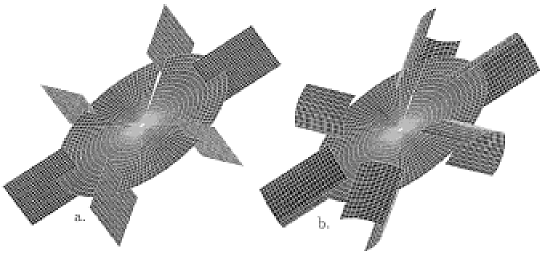

Figure 13. Turbine geometries and surface grid; (a) Rushton turbine (6RT), (b) SCABA 6SRGT.

Table VII. Tank dimensions with notations as in Figure 14.

	T	H	B	C	D	l_a	l_p	D_p
6RT	0.444 m	T	$0.1T$	$0.5T$	$0.33T$	$0.2D$	$0.25D$	$0.5D$
6SRGT	0.444 m	T	$0.1T$	$0.5T$	$0.36T$	$0.15D$	$0.28D$	$0.5D$

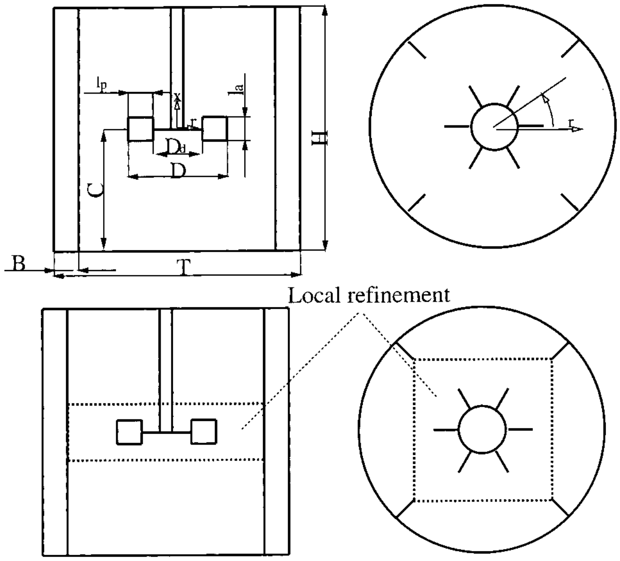

Figure 14. Tank geometry, co-ordinate system and grid configuration.

$$\hat{g}(\tau) = \frac{1}{N_s} \sum_{n=0}^{N_s} g(\tau + n\mathscr{T}) \tag{17}$$

where $\mathscr{T}$ denotes the period and N_s is the total number of samples. Due to spatial symmetry, the period $\mathscr{T}$ is specified as the time it takes for the impeller to turn 60°. Data was sampled along the positive y-axis at time intervals corresponding to an impeller movement of 1°. We hereby obtain data in a cylindrical co-ordinate system without having to interpolate from the Cartesian grid.

Figure 15 shows the phase averaged velocity as a function of angular position behind the blade just off the impeller tip scaled with the tip velocity. As mentioned above, additional complications can occur when imposing no-slip conditions on an infinitely thin surface. It is clear that the use of no-slip conditions here leads to a strong flow towards the turbine centre just behind the blade, which is not present in the measurements by Stoots and Calabrese [33]. Slip conditions on the other hand give a larger velocity in the radial direction than the one observed experimentally. This is probably due to the fact that the slip condition allows the generation of a too powerful recirculation zone. This zone is visible when considering the velocity in the tangential direction (Figure 15). The effect of the boundary condition can also be seen by normalizing the velocity by the mean azimuthal velocity. The counterparts of Figure 15 are depicted in Figure 16. The improvement of the current computations as compared with our previous results (Revestedt *et al.* [32]) is evident.

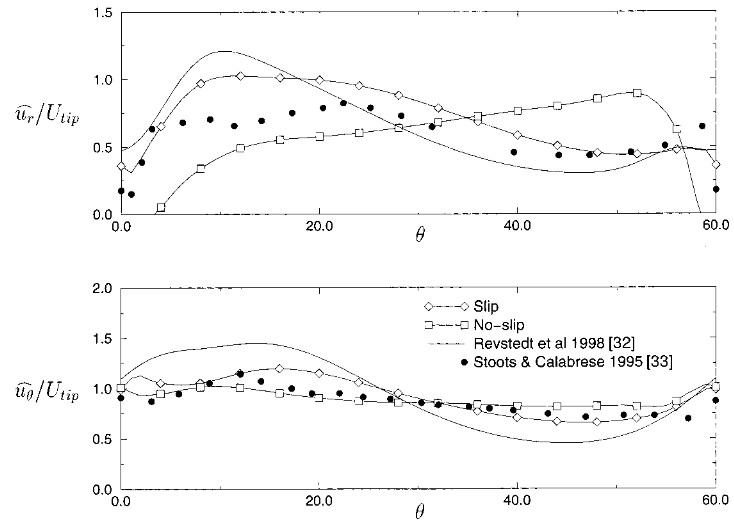

Figure 15. Velocity components for the 6RT scaled with the tip velocity as a function of angle just off the impeller tip ($r/D = 0.505$) at blade mid-height.

An important and often studied parameter for stirred mixers is the turbine power consumption. It is customary to give power consumption of an impeller in the form a dimensionless quantity, the power number P_0

$$P_0 = \frac{P}{\rho N^3 D^5} \tag{18}$$

where P denotes the power, N is the revolutionary speed and D is the impeller diameter. Experimental studies by Amanullah *et al.* [35] and Saito *et al.* [36] show that under non-aerated conditions, the power required by the 6SRGT is 60–65 per cent less than what is required for the Rushton turbine. Here we have calculated the power number in the following manor:

$$P_0 = \frac{\omega}{\rho n^3 D^5} \sum_{m=1}^{M} (F_\theta r)_m \tag{19}$$

where M is the total number of boundary nodes. Our calculation shows a power reduction of 73 per cent for the 6SRGT as compared with the 6RT.

Figure 17 shows the phase averaged, relative to the turbine, velocity fields in an x–θ plane for the 6RT and the 6SRGT turbines respectively. The strong recirculation zone and the trailing vortex pair, first detected by van't Riet and Smith [37], of the 6RT are clearly visible. In the wake of the 6SRGT blade, the recirculation is much less pronounced, which is the main reason for the considerably lower power input as noted above.

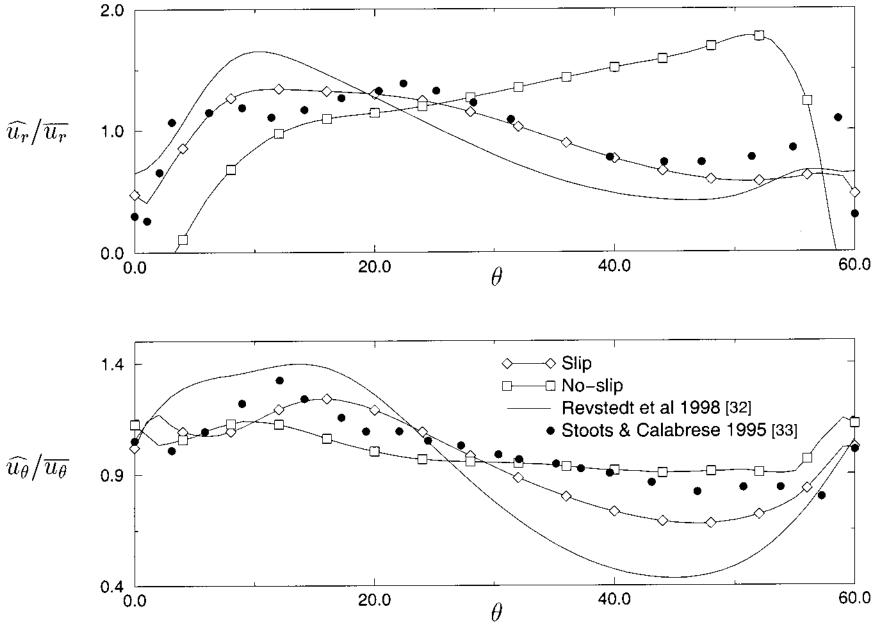

Figure 16. Velocity components for the 6RT scaled with the time-averaged velocity as a function of angle just off the impeller tip ($r/D = 0.505$) at blade mid-height.

7. DISCUSSION

Some important aspects of the 'force' methods have been demonstrated. The advantages of the methods are clear for geometries that are flow- and time-dependent. If only time-dependence exists, one may use other methods (such as overlapping grids) with equally good results. For general cases, we believe that the current approach can be much more attractive once the two main issues demonstrated in this paper are resolved. These two issues are related to numerical efficiency and accuracy respectively.

In the following two sections we propose methods for resolving these issues. The implementation of these remedies is underway, and hence no numerical results are available yet.

7.1. Improved computational efficiency

Handling non-Dirichlet conditions in an iterative process may lead to a considerable reduction in computational efficiency. This happens often due to the fact that updating the boundary conditions implies that the residuals of the system become non-smooth (containing large amplitude, high-frequency error components). The smoothing process requires that the boundary data are spread out over the whole domain (for elliptical problems, such as the one we treat here). The error transport rate is rather poor if only the finest grid (with the largest

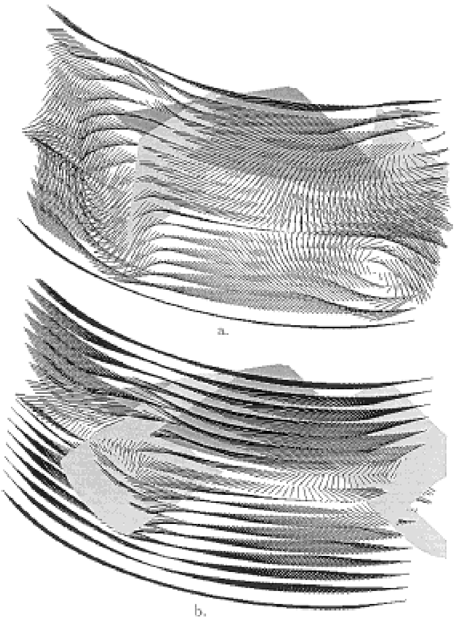

Figure 17. Velocity field in a x–θ plane; (a) 6RT, (b) SCABA (impellers are rotating from left to right) 6SRGT.

number of nodes) is used. Multi-grid methods can easily remedy this situation. The multi-grid implementation that we propose has the following steps. After a given grid is relaxed (including Steps 1–4 in Section 3.1), a coarse grid problem for the boundary force is defined. The boundary 'equation' on the given grid (superscript k) is defined as

$$\Delta F_i^k = R_i^k \tag{20}$$

R_i^k vanishes on the finest grid and on all grids when the boundary conditions are satisfied. The 'error' on the finest grid (M) is

$$R_i^M = \alpha \frac{u_i^{\mathrm{f}} - u_i^{\mathrm{b}}}{h^2} \tag{21}$$

The 'boundary equation' on the coarser grid (k) is defined through the 'error' in the finer grid R_i^{k+1}

$$R_i^k = I_k^{k+1} R_i^{k+1} + \alpha \frac{u_i^{\mathrm{f}} - u_i^{\mathrm{b}}}{h^2} \tag{22}$$

where the velocities are taken on the coarse grid. The coarse grid force is given by

$$\tilde{F}_i^k = I_k^{k+1}\tilde{F}_i^{k+1} \tag{23}$$

The operator I_k^{k+1} stands for a (linear) interpolation of the values on the surface, belonging to a finer surface grid to give a value on the corresponding coarse grid surface cell.

In the full multi-grid calculations, one has to satisfy Equation (20) during the relaxation process. Once the coarsest grid has been fully solved, the correction to the body force (i.e. the change in $\tilde{F}_i^k$ after and before relaxations) is interpolated and added to the existing force on the surface node points.

One also should note that the 'distribution' steps of the algorithm (i.e. Steps 2 and 4 in Section 3.1) have to be corrected for the change in the grid. This can be done in a manner similar to that of the 'error' (23).

The main advantage of this scheme is that the corrections of the boundary force are distributed to the whole field at a high rate on the coarse grids and hence leading to an overall faster convergence.

7.2. Improved numerical accuracy

The surface force equals the force that the boundary excerpts on the fluid. This force balances the stress in the fluid at the surface. Thus, if the flow equations are also valid *through* the boundary, the pressure and/or the velocity gradients are piecewise continuous. Of course, a polynomial approximation over a discontinuity to a piecewise function always yields 'overshoot' and hence low accuracy.

This problem has been recently addressed by Li [38] and Leveque and Li [39]. In these papers, the authors introduce a non-isotropic interpolation so as to allow to an improved (yet less than three) accuracy.

The problem under consideration is similar to that one encounters with shocks. For that problem Ekstrand and Fuchs [40] have introduced an approach by which one subtracts the shock-jump from the dependent variables, leaving only the smooth part for a polynomial approximation. A similar approach could be easily introduced for the body force. By subtracting from the velocity vector a (local) piecewise continuous component (which is related to the local body force), one can use higher-order finite differences for maintaining higher-order accuracy (in principle of any desired order), without the need to adjust the computational stencil coefficients to the problem.

8. CONCLUDING REMARKS

A method for describing complex boundaries on a Cartesian grid has been developed and applied to several flow situations. Good agreement with experimental results and simulations using body fitted grids were achieved.

The way in which the boundary velocity is determined is important for the order accuracy of the solution. Lagrangian interpolation was shown to give higher-order accuracy than taking a Gaussian average. However, the Gaussian average gives faster and more stable convergence of the boundary, furthermore one sees very little difference between the approaches in terms of length of the recirculation zone, drag coefficient, etc. Therefore, the Gaussian average is probably more feasible to use in any practical flow situation.

Considering very slender solid bodies, e.g. the turbine blades, one needs either to develop a different way to distribute the source terms to be able to apply no-slip conditions on the surfaces or, which is the natural solution, to introduce adequate spatial resolution.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support for this work by the EU through the Biotechnology Programme of Framework IV. This work is a part of the project 'Bioprocess Scale-up Strategy', project no. BIO-CT95-0028.

REFERENCES

1. Fletcher CAJ. *Computational Techniques for Fluid Dynamics*, vol. I and II. Springer: Berlin, 1991.
2. Gullbrand J, Bai X-S, Fuchs L. Large eddy simulation of turbulent reacting flows using Cartesian grid and boundary corrections. AIAA Paper No. 98-3317, 1998.
3. Ferziger JH, Peric M. *Computational Methods for Fluid Dynamics*. Springer: Berlin, 1996.
4. Fuchs L. Calculation of flow fields using overlapping grids. In *Notes on Numerical Fluid Mechanics*, vol. 29, Wesseling P (ed.). Vieweg: Braunschweig, 1990; 138–147.
5. Tu JY, Fuchs L. Overlapping grids and multigrid methods for three-dimensional unsteady flow calculations in IC engines. *International Journal for Numerical Methods in Fluids* 1992; **15**: 693–714.
6. Peskin CS. Numerical analysis of blood flow in the heart. *Journal of Computational Physics* 1977; **25**: 220–252.
7. Mayo AA, Peskin CS. An implicit numerical methods for fluid dynamics problems with immersed elastic boundaries. *Contemporary Mathematics* 1993; **141**: 261–277.
8. Tu C, Peskin CS. Stability and instability in the computation of flows with moving immersed boundaries: a comparison of three methods. *SIAM Journal on Scientific and Statistical Computing* 1992; **131**: 361–1376.
9. Lafaurie B, Mantel T, Zaleski S. Direct Navier–Stokes simulation of the near-nozzle region. In *ILASS-Europe-98*, Yule AJ (ed.). UMIST: Manchester, 1998; 54–59.
10. Laufarie BN, Nardone C, Scardovelli, Zaleski S, Zanetti G. Modelling merging and fragmentation in multiphase flows with SURFER. *Journal of Computational Physics* 1994; **113**: 134–147.
11. Esmaeeli A, Tryggvason G. Direct numerical simulations of bubbly flows. Part 1. Low Reynolds number arrays. *Journal of Fluid Mechanics* 1998; **377**: 313–345.
12. Shyy W, Udaykumar HS, Choi D. Structured moving grid and geometric conservation laws for fluid flow computation. *Numerical Heat Transfer Part A* 1998; **34**: 369–397.
13. Shyy W, Udaykumar HS, Rao MM, Smith RW. *Computational Fluid Dynamics with Moving Boundaries*. Taylor & Francis: Washington, DC, 1995.
14. Glowinski R, Pan T-W, Periaux J. A fictitious domain method for Dirichlet problem and applications. *Computer Methods in Applied Mechanics and Engineering* 1994; **111**: 283–303.
15. Glowinski R, Pan T-W, Periaux J. A fictitious domain method for external incompressible viscous flow modeled by Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1994; **112**: 133–148.
16. Glowinski R, Pan T-W, Periaux J. Fictitious domain method for the simulation of Stokes flow past a moving disk. In *Computational Fluid Dynamics '96*, Desideri JA, *et al.* (eds). Wiley: New York, 1996; 64–70.
17. Ionkin AA, Churbanov AG. Numerical simulation of viscous flows in complex domains usind a fictitious region method. In *Computational Fluid Dynamics '96*, Desideri JA, *et al.* (eds). Wiley: New York, 1996; 834–838.
18. Saiki EM, Biringen S. Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method. *Journal of Computational Physics* 1996; **123**: 450–465.
19. Goldstein D, Handler R, Sirovich L. Modeling a no-slip flow boundary with an external force filed. *Journal of Computational Physics* 1993; **105**: 354–366.
20. Smagorinsky J. General circulation experiments with the primitive equations. *Monthly Weather Review* 1963; **91**: 99–165.

21. Held J, Fuchs L. Large eddy simulations of compressible separated flow around a NACA 0012 wing section. AIAA Paper No. 97-1931, 1997.
22. Olsson M, Fuchs L. Large eddy simulation of the proximal region of spatially developing circular jet. *Physics of Fluids* 1996; **8**: 2125–2137.
23. Olsoon M, Fuchs L. Large eddy simulations of a forced semiconfined circular impinging jet. *Physics of Fluids* 1998; **10**: 476–486.
24. Aldama AA. *Filtering Techniques for Turbulent Flow Simulations*. Springer: Berlin, 1990.
25. Fuchs L, Zhao H-S. Solution of three-dimensional viscous incompressible flows by a multi-grid method. *International Journal for Numerical Methods in Fluids* 1984; **4**: 539–555.
26. Fuchs L. Defect-corrections and higher numerical accuracy. In *Proceedings of the GAMM Workshop on 'Efficient Solvers for Elliptic Systems', Notes on Numerical methods in Fluid Mechanics*, vol. 10, Hackbusch W (ed.). Vieweg: Braunschweig, 1984; 52–56.
27. Ericsson M, Fuchs L. Modelling of turbulent flow in an IC-engine configuration. In *Computational Mechanics '95*, Atluri SN, Yagawa G, Cruse TA (eds). Springer: Berlin, 1995; 983–988.
28. Stokes GG. On the effect of the internal friction of fluids on the motion of pendulums. *Transactions of the Cambridge Philosophical Society* 1851; **9**: 8–106.
29. Shriayama S. Flow past a sphere: topological transitions of the vorticity field. *AIAA Journal* 1992; **30**: 349–358.
30. Pruppacher HR, Le Clair BP, Hamielec AE. Some relations between drag and flow pattern of viscous flow past a sphere and cylinder at low and intermediate Reynolds number. *Journal of Fluid Mechanics* 1970; **44**: 781–790.
31. White FM. *Fluid Mechanics*. McGraw-Hill: New York, 1986.
32. Revstedt J, Fuchs L, Tr(g(rdh C. Large eddy simulations of the turbulent flow in a stirred reactor. *Chemical Engineering Science* 1998; **53**: 4041–4053.
33. Stoots CM, Calabrese RV. Mean velocity field relative to a Rushton turbine blade. *AIChE Journal* 1995; **41**: 1–11.
34. Costes J, Couders JP. Study by laser Doppler anemometry of the turbulent flow induced by a Rushton turbine in a stirred tank: influence of the size of the units—II. Spectral analysis and scales of turbulence. *Chemical Engineering Science* 1988; **43**: 2765–2772.
35. Amanuallah A, Serrano-Carreon L, Castro B, Galindo E, Nienow AW. The influence of impeller type in pilot scale xanthan fermentations. *Biotechnology and Bioengineering* 1998; **57**: 98–108.
36. Saito F, Nienow AW, Chatwin S, Moore IPT. Power, gas dispersion and homogenisation characteristics of SCABA SRGT and Rushton turbine impellers. *Journal of Chemical Engineering of Japan* 1992; **25**: 281–287.
37. van't Riet K, Smith JM. The trailing vortex system produced by Rushton turbine agitators. *Chemical Engineering Science* 1975; **30**: 1096–1105.
38. Li Z. A fast iterative algorithm for elliptic interface problems. *SIAM Journal on Numerical Analysis* 1998; **35**: 230–254.
39. Leveque RJ, Li Z. Immersed interface methods for Stokes flow with elastic boundaries or surface tension. *SIAM Journal on Scientific Computing* 1997; **18**: 709–735.
40. Ekstrand P, Fuchs L. Adaptive solution of transonic flows. In *Computational Mechanics '95*, Atlurui SN, Yagawa G, Cruse TA (eds). Springer: Berlin, 1995; 1071–1076.