Journal of Computational Physics **162,** 164–185 (2000)
doi:10.1006/jcph.2000.6531, available online at http://www.idealibrary.com on IDEAL®

Vortex Methods with Spatially Varying Cores

Georges-Henri Cottet,* Petros Koumoutsakos,† and Mohamed Lemine Ould Salihi*

*LMC-IMAG, Universite Joseph Fourier, Grenoble, France; †Institute of Computational Sciences, ETH, Zürich CH-8092, Switzerland, and CTR, NASA Ames 202A-1, Moffett Field, California 94035
E-mail: petros@eniac.ethz.ch

Received September 7, 1999; revised March 30, 2000

The accuracy of vortex methods employing smooth vortex particles/blobs is determined by the blob size, which can be viewed as a mollifier of the vorticity field. For computational efficiency, this core size needs to be spatially variable as particles are used to discretize different parts of the flow field, such as the boundary layer and the wake in bluff body flows. We derive here a consistent approximation for the viscous Navier–Stokes equations using variable size vortex particles. This derivation is based on the implementation of mappings that allow the consistent formulation of the diffusion and convection operators of the Navier–Stokes equations in the context of vortex methods. Several local mappings can be combined giving the capability of "mesh-embedding" to vortex methods. It is shown that the proposed variable method offers a significant improvement on the computational efficiency of constant core size methods while maintaining the adaptive character of the method. The method is ideally suited to flows such as wakes and shear layers and the validity of the approach is illustrated by showing results from cylinder flows and wall-vortex interactions. Using this scheme, previously unattainable simulations of cylinders undergoing rotary oscillations at high Reynolds numbers reveal an interesting mechanism for drastic drag reduction.

Key Words: vortex methods; variable filters; domain decomposition.

1. INTRODUCTION

Vortex methods are based on the particle discretization of the Lagrangian form of the vorticity-velocity formulation of the Navier–Stokes equations. The computational elements are vorticity carrying particles that are being convected with the velocity of the flow field. The vorticity field can always be reconstructed by a linear superposition of the individual vorticity fields carried by the particles. In smooth vortex methods—as opposed to point vortex methods—each particle is associated with a smooth core function, or *vortex blob* which is in particular used in the velocity evaluation by a mollified Biot–Savart law [1].

0021-9991/00 $35.00

The Lagrangian form of vortex methods avoids the explicit discretization of the convective term in the Navier–Stokes equations and the associated stability constraints. The particle positions are modified according to the local flow map, making the method self-adaptive. This adaptation comes at the expense of the regularity of the particle distribution as particles move in order to adapt to the gradients of the flow field. However, the regularity of the particle distribution is important for the convergence of the method. This has been demonstrated by both theoretical results and detailed numerical studies [12]. Particle regularity is best guaranteed by remeshing the particle locations on a regular grid [4] and precludes the use of the random walk method for the simulation of diffusion in direct numerical simulations. We will here consider vortex methods where particle circulations are redistributed among particles to account for diffusion, through the so-called particle strength exchange (PSE) algorithm [4, 6].

Besides particle regularity, another critical factor for the accuracy in the computation of both particle diffusion and velocities is an overlaping condition which imposes that the size of the core surrounding each particle exceeds the inter-particle spacing. This overlapping requirement imposes a servere restriction on the overall adaptivity of vortex methods. For example, in bluff body flows the boundary of the body is the source of vorticity in the flow and it is important to discretize adequately the boundary layer region near the surface of the body. This requirement dictates the size of the particle cores. However, for constant size vortex blobs, as the vorticity gradients decay on the wake, it is clear that the flow is discretized using unnecessarily large numbers of computational elements. This deficiency of constant size vortex methods clearly detracts from the adaptive character of the method and its capability to accurately resolve vorticity gradients while remaining computationally efficient.

This issue has been addressed extensively in the context of grid based methods, in cases where the vorticity gradients are well defined, such as in bluff body flows. The adaptivity in grid based methods is obtained by spatially adjusting the grid size and/or using techniques such as B-splines and embedded meshes [13]. In vortex methods Hou [10] first introduced a variable size vortex method for the Euler equations and he demonstrated its convergence. However, the straightforward implementation of Hou's method in the context of viscous vortex methods [11] is not possible due to the moment properties required by the scheme of particle strength exchange for the simulation of diffusion. Instead, as shown in this paper, variable vortex blobs can be incorporated by introducing a mapping between the spatially varying physical domain and a uniform mapped domain. The formulation of the diffusion operator is further carried out in the mapped domain where the mapped coordinates reside on a regular equispaced grid and discretized on particles through generalized PSE formulas. In this context variable size vortex methods are related to LES formulations with spatially varying filters as introduced by Ghosal and Moin [7].

The mappings required in the formulation of variable size vortex methods can be global or local mappings. In a domain decomposition approach, the use of local mappings is easily extended to combinations of several local mappings, thus providing a vortex method with mesh-embedding capabilities.

The paper is organized as follows. In Section 2 we outline the governing Navier–Stokes equations and their discretization using vortex methods. In Section 3 we present the formulation for variable size vortex methods and in Section 4 we discuss implementation issues, such as the use of the method along with fast N-body solvers. In Section 5 we present results from the application of the method in two dimensions and we compare its performance with vortex methods employing constant size vortex blobs.

2. GOVERNING EQUATIONS AND VORTEX METHODS

The evolution of the vorticity field in a three-dimensional, incompressible, viscous flow is described by the Navier–Stokes equations. These equations may be expressed in a velocity-vorticity ($\mathbf{u}$, $\boldsymbol{\omega} = \nabla \times \mathbf{u}$) formulation as

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\mathbf{u} \cdot \nabla)\boldsymbol{\omega} - (\boldsymbol{\omega} \cdot \nabla)\mathbf{u} - \nu \Delta \boldsymbol{\omega} = 0. \tag{1}$$

The velocity field is obtained by solving the Poisson equation

$$\Delta \mathbf{u} = -\nabla \times \boldsymbol{\omega}. \tag{2}$$

In two dimensions, the vorticity is orthogonal to the flow plane, so that the stretching term disappears and the vorticity equation reduces to a convection-diffusion equation.

The Navier–Stokes equations can be expressed in a Lagrangian formulation as

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{u}(\mathbf{x}_p, t) \tag{3}$$

$$\frac{\mathrm{d}\boldsymbol{\omega}_p}{\mathrm{d}t} = [\nabla \mathbf{u}(\mathbf{x}_p, t)]\,\boldsymbol{\omega}_p + \nu \Delta \boldsymbol{\omega}(\mathbf{x}_p), \tag{4}$$

where $\mathbf{x}_p$, $\boldsymbol{\omega}_p$ denote the locations and the vorticity carried by the fluid elements. These two equations can be solved by simultaneously updating the locations and strengths of the vorticity carrying fluid elements. Alternatively, they may be solved in sub-steps by employing a viscous splitting algorithm where in the first sub-step the fluid elements are advanced with the local flow velocity and their circulations are updated to take into account the vorticity stretching. Diffusion acts at these new locations to modify the vorticity field of the flow.

Vortex methods are based on this Lagrangian description and use particles as vorticity carrying computational elements. The method amounts to tracking these particles and their circulations. This requires one to compute velocities and velocity gradients at particle locations. In this paper, we will essentially focus on the totally grid free implementation of vortex methods, where velocity calculations are based on the Biot–Savart law

$$\mathbf{u} = \int \mathbf{K}(\mathbf{x} - \mathbf{y}) \times \boldsymbol{\omega}\, d\mathbf{y} + \mathbf{U}_0(\mathbf{x}, t). \tag{5}$$

In the above equation $\mathbf{U}_0(\mathbf{x}, t)$ is the solution of the homogenous Eq. (2), and

$$\mathbf{K}(\mathbf{z}) = \begin{cases} -\frac{1}{2\pi}\mathbf{z}/|\mathbf{z}|^2 & \text{in two dimensions} \\ \frac{1}{4\pi}\mathbf{z}/|\mathbf{z}|^3 & \text{in three dimensions.} \end{cases}$$

This formula is differentiated to obtain velocity gradients. In point vortex methods the approximation of the vorticity field is

$$\boldsymbol{\omega}(\mathbf{x}) = \sum_p v_p \boldsymbol{\omega}_p \delta(\mathbf{x} - \mathbf{x}_p), \tag{6}$$

where $\mathbf{x}_p$, v_p, and $\boldsymbol{\omega}_p$ respectively represent the locations, volumes, and vorticities of the particles. In vortex blob methods, particles are assigned a finite core size ε that is used

to compute the velocity field. More precisely, the core shape of the particles is obtained through a cut-off function ζ satisfying $\int \zeta(\mathbf{x})\, d\mathbf{x} = 1$ and the vortex-blob approximation is given by

$$\boldsymbol{\omega}_\varepsilon(\mathbf{x}) = \sum_p v_p \boldsymbol{\omega}_p \zeta_\varepsilon(\mathbf{x} - \mathbf{x}_p) \tag{7}$$

with

$$\zeta_\varepsilon(\mathbf{x}) = \varepsilon^{-d} \zeta\left(\frac{\mathbf{x}}{\varepsilon}\right)$$

(here and in the sequel d is the number of space dimensions). The regularization implied in this formula is motivated by the fact that the particle representation (6) would lead to singular evaluations of the velocity in (5). For a thorough discussion of the point and smooth vortex methods and their numerical analysis we refer to [4] and the references therein.

The numerical implementation of (3)–(4) is completed by discretizing the contribution of the diffusion. Essentially two classes of techniques are presently available for that part of the equations. One is the random walk method of Chorin [1]. The other class consists of resampling schemes, originating with the scheme of Cottet and Mas-Gallic [2]. In this paper we will be concerned with the particle strength exchange method [6]. This resampling scheme is based on integral approximations of the diffusion operator that can be written in the general form

$$\Delta\boldsymbol{\omega} \approx \varepsilon^{-2} \int \eta_\varepsilon(|\mathbf{x} - \mathbf{y}|)[\boldsymbol{\omega}(\mathbf{x}) - \boldsymbol{\omega}(\mathbf{y})]\, d\mathbf{y}, \tag{8}$$

where

$$\eta_\varepsilon(\mathbf{x}) = \varepsilon^{-d} \eta\left(\frac{\mathbf{x}}{\varepsilon}\right)$$

and η is a spherically symmetric function satisfying the normalization condition

$$\int x_i^2 \eta(\mathbf{x})\, d\mathbf{x} = 2 \tag{9}$$

for $i = 1, \ldots, d$. High order approximations can be obtained through the choice of suitable functions η. In practice, one chooses a function that has compact support or is decaying rapidly at infinity. The discretization of the integral in the right-hand side of (8) thus only involves a few particles. This scheme is rather straightforward to implement and adds a marginal numerical cost to the velocity evaluations. It has been shown to yield high resolution particle schemes for viscous flows [11]. Finally it can be easily extended to handle spatially varying diffusion coefficients, a flexibility which will prove useful in the sequel. The vortex blob method defined so far can be summarized by the following system of ordinary differential equations for the particle locations and vorticities

$$\frac{d\mathbf{x}_p}{dt} = \sum_{q=1}^{N} v_q K_\varepsilon(\mathbf{x}_p - \mathbf{x}_q) \times \boldsymbol{\omega}_q + \mathbf{U}_0(\mathbf{x}_p, t) \tag{10}$$

$$\frac{d\boldsymbol{\omega}_p}{dt} = \left[\sum_{q=1}^{N} v_q \nabla K_\varepsilon(\mathbf{x}_p - \mathbf{x}_q) \times \boldsymbol{\omega}_q\right] \boldsymbol{\omega}_p + \nu\varepsilon^{-2} \sum_{q=1}^{N} v_q [\boldsymbol{\omega}_q - \boldsymbol{\omega}_p] \eta_\varepsilon(|\mathbf{x}_p - \mathbf{x}_q|). \tag{11}$$

These equations must be supplemented by initial conditions which depend on how particles are initialized.

3. VORTEX METHODS WITH VARIABLE BLOB SIZES

The parameter ε used in the definition of the blobs as well as in the diffusion approximation defines the minimal scales that can be resolved by the method. It plays the role of the grid size in grid based Eulerian methods. As such it is natural to allow this parameter to vary with time and space. Here we are concerned with the case of a spatially variable blob size and we first consider its implication in velocity calculations.

3.1. *Variable Core Size for the Biot–Savart Integral*

A variable blob method would consist of defining a function $\varepsilon(\mathbf{x}) \ll 1$ and modifying (7) into

$$\omega_\varepsilon^h(\mathbf{x}) = \sum_p \alpha_p \zeta_{\varepsilon(\mathbf{x}_p)}\left(\mathbf{x} - \mathbf{x}_p^h\right). \tag{12}$$

This leads to the velocity formula

$$\mathbf{u}^h(\mathbf{x}) = \sum_p \alpha_p \mathbf{K}_{\varepsilon(\mathbf{x}_p)}\left(\mathbf{x} - \mathbf{x}_p^h\right). \tag{13}$$

The convergence of the method based on this formulas for the particle velocity evaluations has been proved for two-dimensional flows by T. Hou [10], under the assumption that there exists a smooth function f such that

$$\varepsilon(\mathbf{x}) = \varepsilon f(\mathbf{x}) \qquad \text{with } 0 \le C \le f(\mathbf{x}) \le C', \tag{14}$$

for some constants C, C'. The numerical analysis of particle methods generally distinguishes the contribution to the discretization error resulting, on the one hand, from the sampling of the continuous vorticity onto a discrete set of particles, and, on the other hand, from the regularization involved in the velocity evaluations.

Let us only outline here the error analysis for the regularization error produced on a smooth vorticity field ϕ. This regularization error can be expressed as

$$E = \int \phi(\mathbf{y})\zeta_{\varepsilon(\mathbf{y})}(\mathbf{x} - \mathbf{y})\, d\mathbf{y} - \phi(\mathbf{x}).$$

For simplicity, let us assume that the cut-off ζ has a compact support, of size unity. As it is classical in particle methods, we assume that this cut-off has order r in the sense that

$$\int \zeta(\mathbf{x})\, d\mathbf{x} = 1 \tag{15}$$

$$\int \mathbf{x}^{\mathbf{i}} \zeta(\mathbf{x})\, d\mathbf{x} = 0 \qquad \text{if } |\mathbf{i}| \le r - 1 \tag{16}$$

$$\int |\mathbf{x}|^r |\zeta(\mathbf{x})|\, d\mathbf{x} < \infty, \tag{17}$$

where $\mathbf{i} = (i_1, i_2)$ is a multi-index, $|\mathbf{i}| = i_1 + i_2$, and $\mathbf{x}^{\mathbf{i}} = x_1^{i_1} x_2^{i_2}$. We first set

$$\mathbf{z} = \frac{\mathbf{x} - \mathbf{y}}{\varepsilon(\mathbf{y})}.$$

We can write

$$\frac{\partial z_i}{\partial y_j} = \frac{1}{\varepsilon(\mathbf{y})}\left[\delta_{ij} - \varepsilon z_i \frac{\partial f}{\partial y_j}\right]$$

and a Taylor expansion of f around $\mathbf{x}$ yields

$$\frac{\partial z_i}{\partial y_j} = \frac{1}{\varepsilon(\mathbf{y})}\left[\delta_{ij} + \sum_{k=1}^{r} \varepsilon^k \mathbf{z}^k a_k\right] + O(\varepsilon^r).$$

This first shows that, for ε small enough, $\mathbf{y} \to \mathbf{z}$ defines a one-to-one smooth mapping and that the Jacobian of this mapping can be evaluated as

$$\varepsilon(\mathbf{y})^2 \left[1 + \sum_{k=1}^{r} \varepsilon^k \mathbf{z}^k b_k\right] + O(\varepsilon^r).$$

As a result

$$\int \zeta_{\varepsilon(\mathbf{y})}(\mathbf{x} - \mathbf{y})\, d\mathbf{y} = 1 + O(\varepsilon^r)$$

and we can rewrite E as

$$E = \int (\phi(\mathbf{y}) - \phi(\mathbf{x}))\zeta_{\varepsilon(\mathbf{y})}(\mathbf{x} - \mathbf{y})\, d\mathbf{y} + O(\varepsilon^r).$$

If ϕ is of class C^r, its Taylor expansion gives

$$E = \sum_{|\boldsymbol{\alpha}| \le r} c_{\boldsymbol{\alpha}} \|\phi\|_{r,\infty} \int \mathbf{z}^{\boldsymbol{\alpha}} \zeta(\mathbf{z}) \varepsilon(\mathbf{y})^{|\alpha|}\, d\mathbf{z} + O(\varepsilon^r).$$

Upon expanding f again around $\mathbf{x}$, one finally obtains

$$E = \sum_{1 \le |\boldsymbol{\alpha}| \le r-1} d_{\boldsymbol{\alpha}} \|\phi\|_{r,\infty} \varepsilon^{|\alpha|} \int \mathbf{z}^{\boldsymbol{\alpha}} \zeta(\mathbf{z})\, d\mathbf{z} + O(\varepsilon^r).$$

and, in view of the conditions (16) and (17), $E = O(\varepsilon^r)$.

It follows that, for inviscid flows, the vortex method with smoothly varying blob size has the same convergence properties as for uniform blobs.

3.2. *Diffusion with Variable Core Sizes*

We are now concerned with the implementation of a PSE formula similar to (8) with variable values of ε.

Let us first observe that (8) can be derived starting from a straightforward Taylor expansion of ω around $\mathbf{x}$,

$$\omega(\mathbf{y}) = \omega(\mathbf{x}) + (\mathbf{y}-\mathbf{x})\cdot\nabla\omega(\mathbf{x}) + \sum_{i,j}(x_i - y_i)(x_j - y_j)\frac{\partial^2\omega}{\partial x_i \partial x_j} + \cdots.$$

By the symmetry properties of η, the first order terms and the cross terms involving the second order derivates of ω drop out in the right-hand side of (8). The normalization condition (9) then leads to the desired approximation. If we now follow the numerical analysis outlined in Subsection 3.1, we notice that a straightforward use of a variable value of ε in the right-hand side of (8) would bring nonvanishing terms from a combination of first order terms in the Taylor expansions of ω and ε. To avoid this inconsistency, we need to assume a mapping from the physical coordinates, with variable-size blobs, to a coordinate system where blobs have a uniform size.

For simplicity we present first the analysis for the one-dimensional case and then we generalize it for arbitrary mappings. Particular mapping examples are described in Section 5.

3.2.1. The 1-D case. We will denote by x, y locations in the physical space, with variable blob size, and by $\hat{x}, \hat{y}$ locations in the mapped coordinates, where the grid size is uniform. We will assume that the mapping is given by the formulas

$$x = f(\hat{x}), \qquad \hat{x} = g(x), \qquad \omega(x) = \hat{\omega}(\hat{x}).$$

Writing derivatives in the mapped coordinates yields

$$\frac{d^2\omega}{dx^2} = h(\hat{x})\frac{d}{d\hat{x}}\left[h(\hat{x})\frac{d\hat{\omega}}{d\hat{x}}\right],$$

where $h(\hat{x}) = g'(x)$. Next, we use the following integral approximation, the proof of which relies on Taylor expansions at order 2 similar to those in [4] (proof of Eq. 8),

$$\frac{d}{d\hat{x}}\left[h(\hat{x})\frac{d\hat{\omega}}{d\hat{x}}\right] \simeq \varepsilon^{-3}\int \frac{h(\hat{x})+h(\hat{y})}{2}[\hat{\omega}(\hat{x}) - \hat{\omega}(\hat{y})]\eta\left(\frac{\hat{x}-\hat{y}}{\varepsilon}\right)d\hat{y}.$$

In the above formula, the kernel η satisfies the moment properties (9) and ε is a *constant* blob size. This leads to the following PSE scheme for the diffusion equation in one dimension,

$$\frac{d\omega_p}{dt} = \nu\varepsilon^{-3}h(\hat{x}_p)\sum_q \hat{v}_q \frac{h(\hat{x}_p)+h(\hat{x}_q)}{2}[\omega_q - \omega_p]\eta\left(\frac{\hat{x}_p - \hat{x}_q}{\varepsilon}\right), \tag{18}$$

where the $\hat{v}_q$ denote the volumes of the mapped particles. In this formula, vorticity is exchanged in a range of order ε around $\hat{x}_p$ in the mapped coordinate. In the physical coordinates this range becomes $\varepsilon h(\hat{x}_p)$ which corresponds to the desired variable blob size. Notice also that the volumes of the physical and mapped particles are related through the Jacobian of the mapping

$$\hat{v}_q = v_q h(\hat{x}_q)$$

which establishes that the scheme in Eq. (18) is indeed conservative.

3.2.2. General case. We now derive general formulas for PSE schemes using variable blob size. We assume that the computational domain Ω, where we wish to use variable blob size, is mapped through a mapping denoted by $\mathbf{F}$ to a domain $\hat{\Omega}$ with uniform blobs. We will use the notations

$$x = \mathbf{F}(\hat{\mathbf{x}}); \qquad u(x) = \hat{u}(\hat{\mathbf{x}}).$$

We consider the inverse mapping $\mathbf{G} = \mathbf{F}^{-1}$ and set, for $i,\ j$ in $[1, d]$ where d is the dimension,

$$a_{ij} = \frac{\partial G_i}{\partial x_j}.$$

With these notations, we may write

$$\Delta_{\mathbf{x}} u = \sum_{i,j,k} a_{ji} \frac{\partial}{\partial \hat{x}_j} \left(a_{ki} \frac{\partial \hat{u}}{\partial \hat{x}_k} \right). \tag{19}$$

If $J = \det[a_{ij}]$ denotes the Jacobian determinant of the mapping $\mathbf{F}$, we observe that, for $i \in [1, d]$,

$$\sum_{j=1}^{d} \frac{\partial}{\partial \hat{x}_j} (a_{ji} J) = 0. \tag{20}$$

This results from the fact that if ϕ is any vector-valued test function defined in Ω and vanishing on its boundary, we have

$$0 = \int_{\Omega} \operatorname{div}_{\mathbf{x}} \phi \, d\mathbf{x} = \sum_j \int_{\hat{\Omega}} a_{ji} \frac{\partial \hat{\phi}_i}{\partial \hat{x}_j} J^{-1} \, d\hat{\mathbf{x}}.$$

By the divergence theorem, this implies

$$0 = \sum_j \int_{\hat{\Omega}} \frac{\partial}{\partial \hat{x}_j} (a_{ji} J^{-1}) \hat{\phi}_i \, d\hat{\mathbf{x}}$$

which proves our claim. Hence we can rewrite (19) as

$$\Delta_{\mathbf{x}} u = J \sum_{j,k} \frac{\partial}{\partial \hat{x}_j} \left(b_{jk} \frac{\partial \hat{u}}{\partial \hat{x}_k} \right) \tag{21}$$

with $b_{jk} = J^{-1} \sum_i a_{ki} a_{ji}$, or, equivalently,

$$\Delta_{\mathbf{x}} u = J \operatorname{div}_{\hat{\mathbf{x}}} [B \nabla_{\hat{\mathbf{x}}} \hat{u}], \tag{22}$$

where B is the matrix with entries b_{ij}.

It now remains to derive integral approximations in the mapped coordinates for the differential operator in the right-hand side of (22). In [6], general formulas of the form

$$\operatorname{div}[B \nabla u](\mathbf{x}) \simeq \varepsilon^{-2} \sum_{i,j=1}^{d} \int \psi_{ij}^{\varepsilon}(\mathbf{x} - \mathbf{y}) M_{ij}(\mathbf{x}, \mathbf{y}) [u(\mathbf{y}) - u(\mathbf{x})] \, d\mathbf{y} \tag{23}$$

are given (here and in the rest of this discussion we drop the notations $\hat{}$, bearing in mind that we are now brought back to a situation where we can use uniform blob size). In this formula, the $M_{ij}(\mathbf{x}, \mathbf{y})$ are symmetric functions and ψ_{ij}^{ε} are cut-off functions. These functions are related to each other and to the matrix B through general conditions that are detailed in [6]. For our purpose it will be enough to focus on the following particular choice. We choose ψ_{ij} under the form

$$\psi_{ij}(\mathbf{x}) = x_i x_j \theta(|\mathbf{x}|)$$

and the spherically symmetric cutoff θ is normalized such that

$$\int x_i^4 \theta(\mathbf{x})\, d\mathbf{x} = d + 2.$$

Then a second order approximation of the form (23) is obtained if one chooses

$$M_{ij}(\mathbf{x}, \mathbf{y}) = m_{ij}\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right)$$

with $m = [m_{ij}]$ given in terms of B through

$$m = B - \frac{1}{d+2}\operatorname{tr} B.$$

As written above, the approximation is second order; higher order approximations can be obtained by requiring additional moment conditions for θ [6].

The PSE scheme for the Navier–Stokes equation which immediately follows from these formulas is given by

$$\begin{aligned}\frac{d\boldsymbol{\omega}_p}{dt} = {} & \nu\varepsilon^{-4} J(\mathbf{x}_p) \sum_{q,i,j} \hat{v}_q \left(\hat{x}_p^i - \hat{x}_q^i\right)\left(\hat{x}_p^j - \hat{x}_q^j\right)\theta^{\varepsilon}(\hat{\mathbf{x}}_p - \hat{\mathbf{x}}_q) \\ & \times \left[b_{ij} - \frac{1}{d+2}\sum_i b_{ii}\delta_{ij}\right]\left(\frac{\hat{\mathbf{x}}_p + \hat{\mathbf{x}}_q}{2}\right)(\boldsymbol{\omega}_q - \boldsymbol{\omega}_p). \end{aligned} \tag{24}$$

Since the volumes of the particles in the physical and mapped spaces are related through

$$v_p J(\mathbf{x}_p) = \hat{v}_p$$

we observe, as in the 1D example, that the scheme automatically satisfies the conservation property

$$\frac{d}{dt}\left(\sum_p v_p \boldsymbol{\omega}_p\right) = 0.$$

The PSE scheme in Eq. (25) relies on the explicit knowledge of a global mapping in the computational domain. We will see in Subsection 4.3 below a strategy, more suitable to deal with complex geometries, enabling us to combine several local mappings.

4. IMPLEMENTATION ISSUES

In the implementation of viscous vortex methods there are two issues that need to be addressed in order to achieve fast and accurate computations. These are:

- the remeshing of the particle locations
- the use of fast codes for the velocity evaluation.

These issues have been addressed in detail in the past [12] for the case of uniform blobs. The use of variable tools does not pose any additional difficulties as remeshing can be performed in the mapped uniform space and fast multipole codes are well suited to the concept of variable blob sizes.

4.1. *Remeshing for Variable Size Blobs*

In all particle simulations of inviscid or high Reynolds flow, it is critical to maintain some regularity in the Lagrangian grid. The convergence of vortex methods is conditioned by the overlapping at all times of the blobs as shown by numerical analysis and benchmark simulations.

In a variable blob method, regridding must be applied in the mapped coordinates, and at a frequency which prevents particles living in the low resolution areas from traveling too far in the high resolution areas between two regriddings. This is not a drastic constraint: in a vortex code, the time-step is typically scaled by the inverse of the maximum vorticity, and remeshing at every time step with a third or fourth interpolation formula introduces only a marginal numerical discrepancy [5, 12].

4.2. *Fast Multipole Codes and Variable Blobs*

In vortex methods we can compute the velocity field using the Biot–Savart law thus explicitly enforcing the far field boundary condition. The straightforward implementation of the method has a nominal cost that scales with the square of the number of computational elements. Fast multipole algorithms reduce the computational cost of the algorithm to scale linearly with the number of particles by a hierarchical clustering of the particles into a tree data structure. In the implementation of the fast multipole algorithm as it is discussed in [12] clusters are formed by recursively subdividing the rectangular domain encompassing all particles. This process terminates when less than a certain number of particles resides in each rectangle. A particle interacts directly with all particles residing in non-divisible clusters found within a certain range. The interactions with particles in the other clusters are accounted for via the multipole expansions that consider the particles as point vortices—a just approximation a few diameters away from the particle cores. One can see then that as the clustering of particles is determined by the number kept in each cell, when we maintain a certain regularity—via remeshing—on the locations of the variable size particles, the clusters formed will maintain the point-vortex attributes required by the multipole algorithm. Hence the implementation of fast codes in the physical space with variable blobs does not pose any additional constraints.

4.3. *Variable Blobs, Local Mappings, and Domain Decomposition*

As we will see below, variable blob methods based on a global mapping are a useful tool for a number of wall-bounded flows (e.g., channel flows, flows past an airfoil), for which

such a mapping is available. For more complex geometries or for domains involving several bodies, global mappings are not always available and it is desirable to have a method which could rely on local mappings and their superposition. Typically, in a configuration involving several obstacles, around each obstacle a fine grid would be related to one local mapping. Local particle solvers, with variable blobs associated to each mapping, must then be linked through domain decomposition techniques.

Domain decompositions algorithms are distinguished in two broad classes, depending on whether they are based on matching or overlaping sub-domains. Matching sub-domains impose strong geometrical constraints on the grids defined on each sub-domain. By contrast, the flexibility added by overlapping sub-domains has already been useful to derive domain decomposition algorithms that combine Eulerian and Lagrangian solvers [3, 15].

In a domain decomposition algorithm involving particle solvers, the coupling between the sub-domains must be done at two levels: interface conditions must be supplied to compute particle velocities and to update particle strengths.

Given a vorticity field, the determination of the velocity amounts to the solution of a Poisson equation, and the Schwarz alternating method is a natural way to enforce the right interface conditions. In a vortex code the method iterates between the boundary source terms that must be added to the Biot–Savart law in each sub-domain. The fact that variable blobs, corresponding to different mappings, are being implemented, does not introduce any difficulty provided the overlapping zone has a width exceeding the blob sizes in this area. We refer to [15] for details on the implementation of this method in the context of a particle-grid domain decomposition.

Once the velocities are evaluated on each sub-domain, it remains to update particle locations and circulations. Since vortex methods are based on explicit time-discretization of the vorticity convection-diffusion equation, the vorticity transfer from one domain to another is simply achieved by interpolating the particle strengths in the overlapping zone.

To be more specific let us consider the example sketched in Fig. 1. In this example, Ω_1 and Ω_2 are two domains with non-uniform grid-spacing, connected with a domain Ω_3 with uniform spacing of Fig. 1. Assume that at time t_n the vorticity is known as

$$\omega^j(\mathbf{x}) = \sum_p \alpha_p^j \delta\left(\mathbf{x} - \mathbf{x}_p^j\right)$$

for $j = 1, 2, 3$, respectively, in Ω_1, Ω_2, Ω_3. The algorithm to update the particle strength (assuming that the velocities are known everywhere) proceeds as follows:

- Particles $\mathbf{x}_p^1$ and $\mathbf{x}_p^2$ lying in $\Omega_1 \cap \Omega_3$ and $\Omega_2 \cap \Omega_3$, respectively, are remeshed on the uniform mesh Ω_3; we thus obtain a new distribution $\tilde{\omega}^3(\mathbf{x}) = \sum_p \tilde{\alpha}_p^3 \delta(\mathbf{x} - \mathbf{x}_p^3)$ in Ω_3 (note that $\tilde{\alpha}_p^3 = \alpha_p^3$ if $\mathbf{x}_p^3 \in \Omega_3 - (\Omega_1 \cup \Omega_2)$).
- Similarly particles $\mathbf{x}_p^3$ in $\Omega_1 \cap \Omega_3$ (resp. $\mathbf{x}_p^2$ in $\Omega_2 \cap \Omega_3$) carrying circulations α_p^3 are remeshed on the grid points of Ω_1 (resp. Ω_2), giving new distributions $\tilde{\omega}^j(\mathbf{x}) = \sum_p \tilde{\alpha}_p^j \delta(\mathbf{x} - \mathbf{x}_p^j)$, $j = 2, 3$.
- For $j = 1, 2, 3$, particles $\mathbf{x}_p^j$ with circulations $\tilde{\alpha}_p^j$ are removed and exchange circulations through the PSE scheme with variable blobs according to the mapping defined in the corresponding sub-domain.

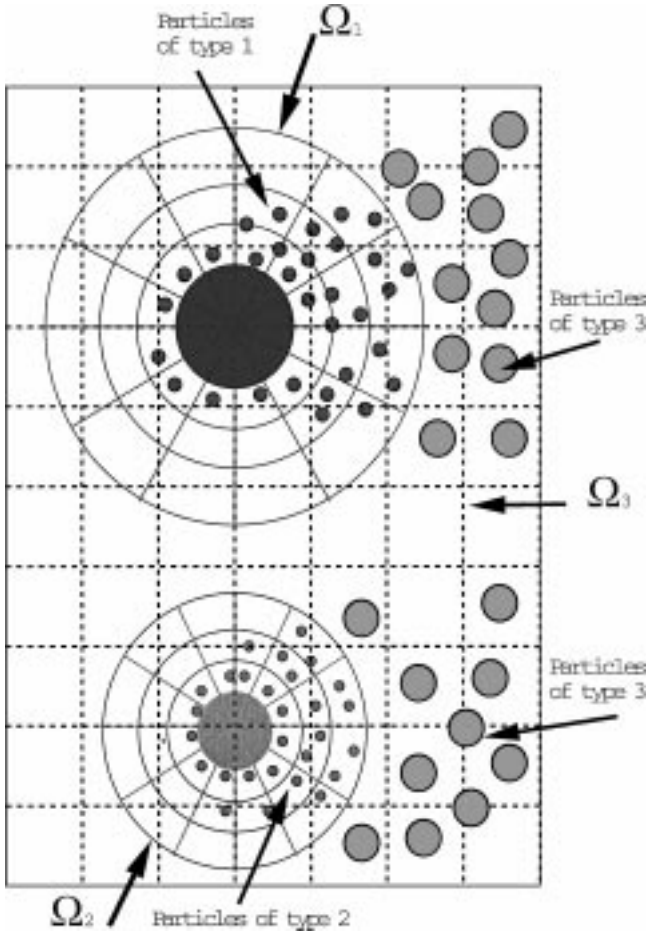

FIG. 1. Decomposition of particle locations.

At the end of these steps, the vorticity has been updated in all 3 domains. Provided that the overlapping width exceeds the core radius of the PSE kernel, this procedure allows a consistent transfer of vorticity through diffusion. As a matter of fact, in absence of convection the procedure just outlined is very reminiscent to what a time explicit finite-difference solver would do under the same circumstances. The transfer of vorticity through convection is implicity done through the advection of particles which enter domains $\Omega_i - \Omega_j$ from the overlapping zone.

Note that this strategy requires remeshing at every time-step, which makes it crucial to use a non-dissipative remeshing technique. As we mentioned in Subsection 4.1, third or fourth order interpolation provides smooth remeshing formulas which can be repeated at every time-step without adding significantly to the truncation error of the overall scheme.

5. RESULTS

To illustrate the variable blob techniques we will focus on applications to two-dimensional flows. Implementation of variable blobs for three-dimensional bluff-body flows in currently under way.

5.1. *Flow Past Cylinders*

Vortex methods with variable cores are well suited to simulations of bluff-body flows as small size particles are needed to resolve the vorticity gradients near the surface of the body, while larger blobs would be suitable to discretize the smoothly varying vortices in the wake of the flow. In order to demonstrate the validity of the method we employ here an exponentially varying mapping and (i) we present results comparing the present method with the results obtained from benchmark studies of flows past impulsively started cylinders using non-variable blobs, (ii) we calculate the Strouhal number and the average drag coefficient for long time simulations at $\mathrm{Re} = 200$, and (iii) finally we present preliminary results from the application of the method to simulation of the flow of high Reynolds numbers past a cylinder performing rotary oscillations.

For the polar mapping employed in these simulations we will denote by r, θ locations in the physical space, with non-uniform grid size, and by $\hat{r}$, $\hat{\theta}$ locations in the mapped coordinates, where the grid size is uniform. Here we consider an explicit mapping given by the formulas

$$r = e^{\hat{r}} \tag{25}$$

$$\theta = \hat{\theta}. \tag{26}$$

The Laplacian operator in physical space may be expressed as

$$\nabla^2 \boldsymbol{\omega} = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \boldsymbol{\omega}}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 \boldsymbol{\omega}}{\partial \theta^2} \tag{27}$$

$$= \frac{1}{r}\frac{\partial}{\partial \hat{r}}\frac{\partial \hat{r}}{\partial r}\left(r\frac{\partial \hat{r}}{\partial r}\frac{\partial \hat{\boldsymbol{\omega}}}{\partial \hat{r}}\right) + \frac{1}{r^2}\frac{\partial^2 \hat{\boldsymbol{\omega}}}{\partial \hat{\theta}^2} \tag{28}$$

$$= \frac{1}{r^2}\left(\frac{\partial^2 \hat{\boldsymbol{\omega}}}{\partial \hat{r}^2} + \frac{\partial^2 \hat{\boldsymbol{\omega}}}{\partial \hat{\theta}^2}\right). \tag{29}$$

Hence the task now is to approximate the modified Laplacian operator with an integral operator. However, this can be formulated easily because in the $(\hat{r}, \hat{\theta})$ space we are dealing with a uniform grid. Hence the strengths of the computational elements in the physical space may be updated as

$$\frac{\mathrm{d}\alpha_p}{\mathrm{d}t} = \frac{\nu}{r_p^2 \hat{\varepsilon}^2} \sum_{q=1}^{N} \frac{1}{r_q^2}[\alpha_q v_p - \alpha_p v_q]\hat{\Lambda}_{\hat{\varepsilon}}(|\hat{\mathbf{R}}_p - \hat{\mathbf{R}}_q|), \tag{30}$$

where for the case of a Gaussian core function we have that

$$\hat{\Lambda}_{\hat{\varepsilon}}(|\hat{\mathbf{R}}_p - \hat{\mathbf{R}}_q|) = \frac{1}{\pi \hat{\varepsilon}^2} e^{\frac{-(\hat{r}_p - \hat{r}_q)^2 + (\hat{\theta}_p - \hat{\theta}_q)^2}{2\hat{\varepsilon}^2}}. \tag{31}$$

In Fig. 2 we show the computational elements for the simulation of impulsively started flow at $\mathrm{Re} = 200$, past a circular cylinder using variable and constant blob sizes. The relative placement and the area discretized in the two cases can be further appreciated by examining a close up of Fig. 2 in Fig. 3. The savings on computational elements and hence in computational efficiency are one order of magnitude up to the time $\mathrm{T} = 15$ as shown in Fig. 4. Longer time simulations increase these savings as more and more particles are entering the wake of the cylinder. The savings of one order of magnitude in computational

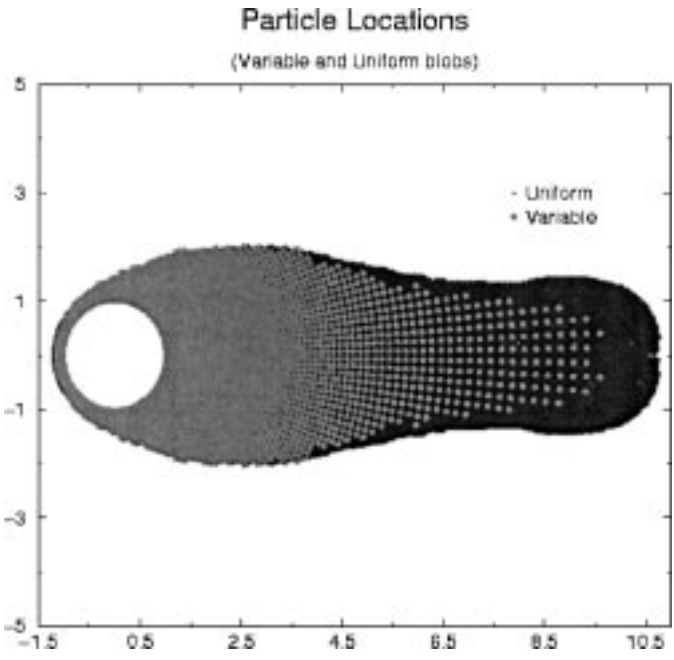

FIG. 2. Impulsively started cylinder, Re = 200, T = 10.0—particle locations.

time are achieved while maintaining the accuracy of the method as it is demonstrated in Fig. 5 showing the flow drag coefficient and in Fig. 6 showing vorticity contours for this impulsively started cylinder.

Longer time simulations were also conducted using the variable size vortex method. In Fig. 7 we show the particle locations in the wake of a circular cylinder at Re = 200

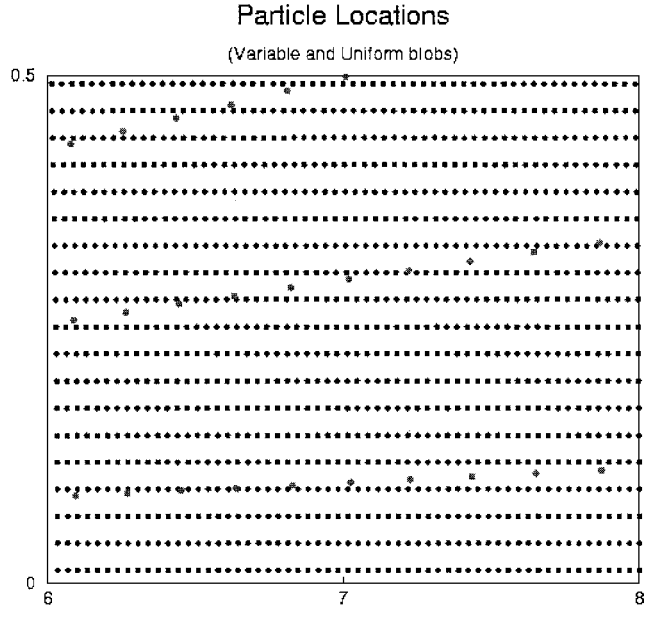

FIG. 3. Close-up of particle locations shown in Fig. 2.

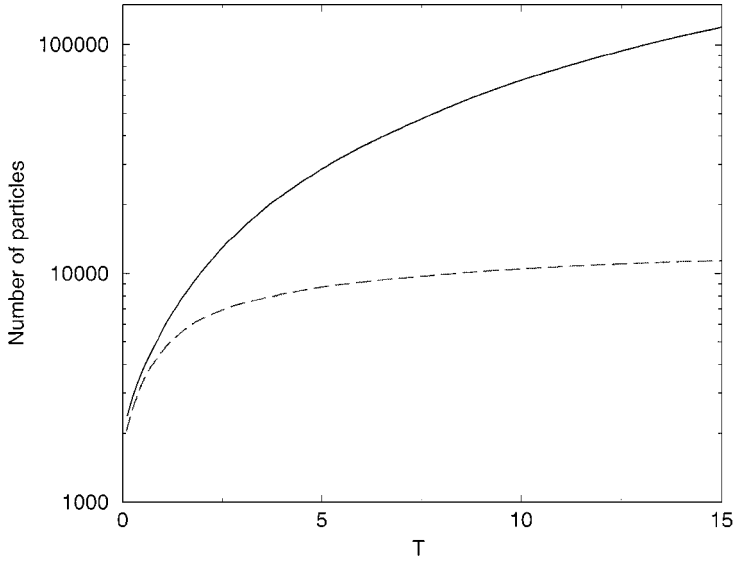

FIG. 4. Number of particles as a function of time for Re = 200. Variable size blobs (dashed line) and constant size blobs (solid line).

while in Fig. 8 we show the vorticity contours of the flow field. Calculations of the average drag coefficient ($C_d \approx 1.35$) and the Strouhal frequency ($St \approx 0.19$) are in good agreement with results reported in the literature for related two-dimensional simulations [9].

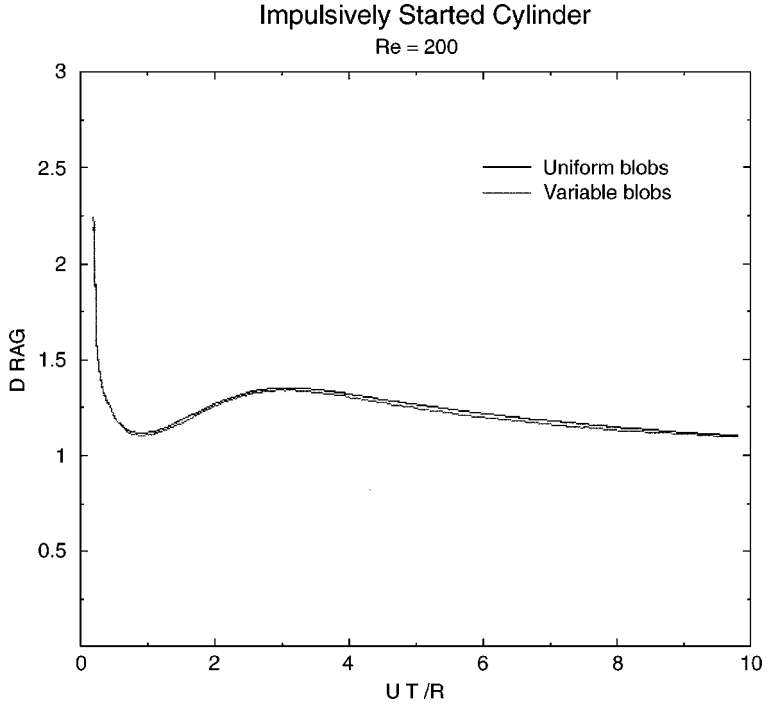

FIG. 5. Comparison of drag coefficient computed with uniform and variable. size particles. Impulsively started cylinder, Re = 200.

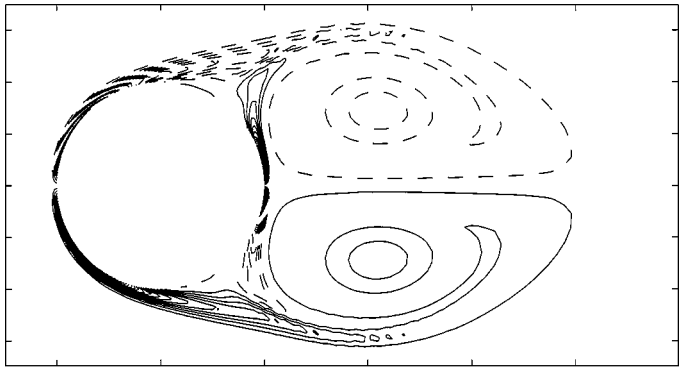

FIG. 6. Vorticity contours at T = 15, as computed by uniform (top half) and variable (bottom half) size vortex methods. Impulsively started cylinder, Re = 200.

Finally using the computational efficiency provided by variable vortex methods we have conducted long time calculations of a cylinder undergoing rotary oscillations. For Re = 2000, the results of the 2D simulations showed a drastic drag reduction for certain rotational frequencies. This drag reduction is attributed to the modification of the shedding mechanisms from the surface of the cylinder. As shown in Fig. 9 the cylinder rotation results in the ejection of bipolar vortex structures that result in high drag reduction. Further investigations and extensive two- and three-dimensional studies of these phenomena are the subject of ongoing research.

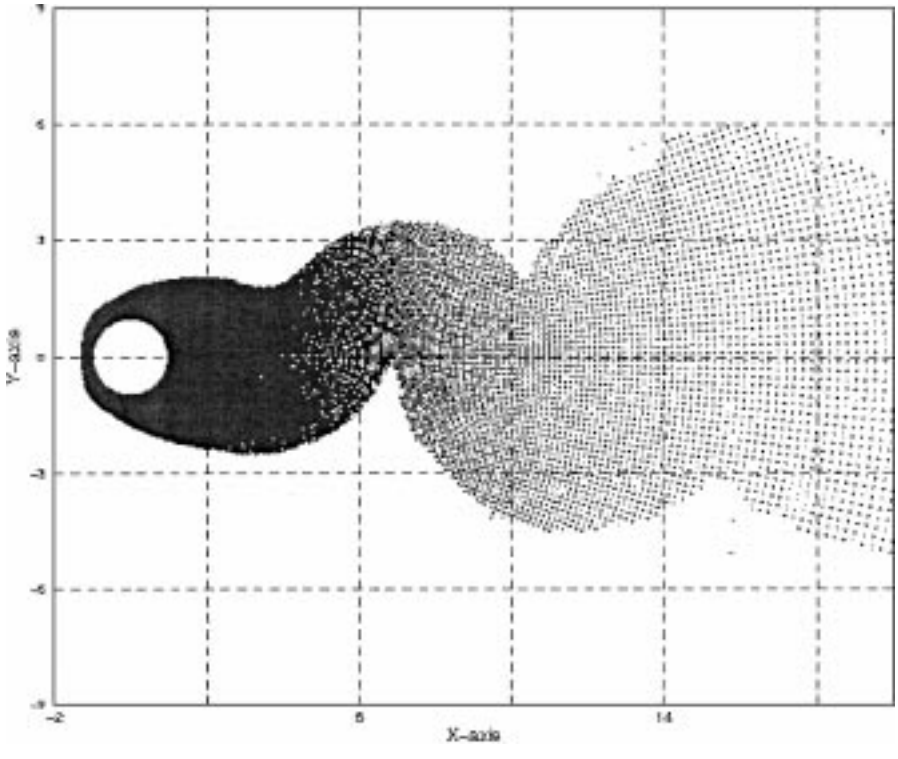

FIG. 7. Particle locations for a vortex wake (at Re = 200) discretized by variable cores.

FIG. 8. Vorticity contours for a vortex wake at Re = 200.

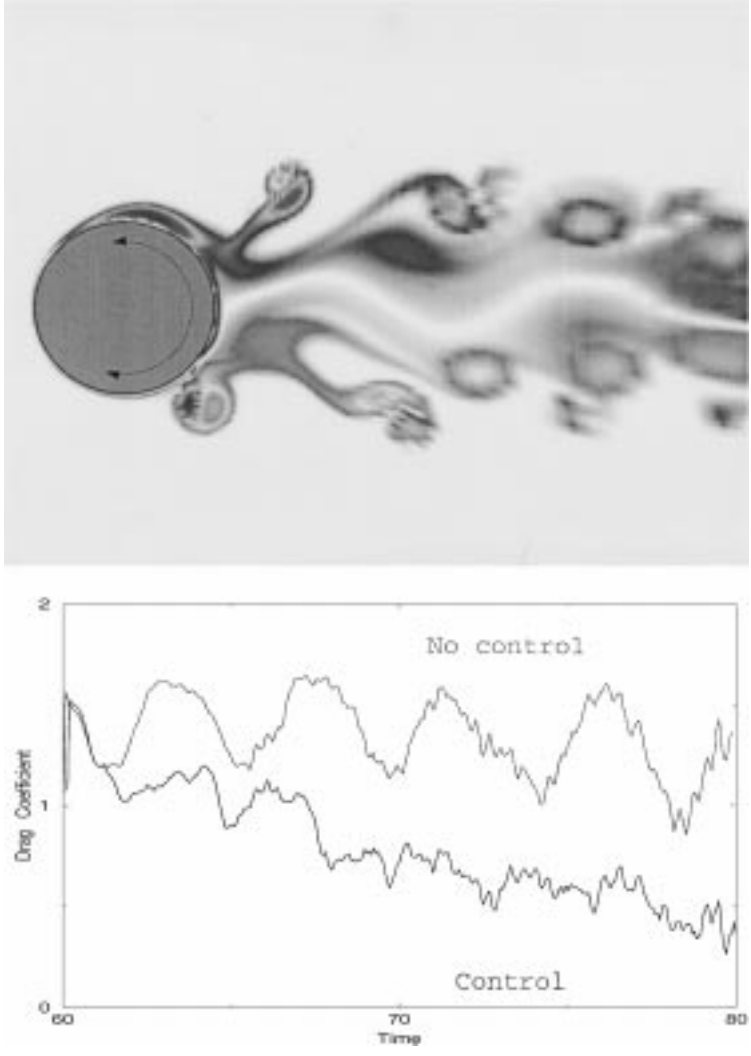

FIG. 9. Vorticity contours and drag coefficient for a cylinder performing rotary oscillations at Re = 2000.

5.2. *Vortex-Wall Interaction: Rebound of a Dipole*

We now consider the classical case of a dipole impinging on a flat wall. Systematic comparisons have shown that vortex methods with uniform blobs give results of the same accuracy, for a given mesh size, as centered finite-difference schemes [15]. We wish here to allow grid refinement near the wall, along both directions. Although this geometry appears simpler than the cylinder, it turns out that the mapping is more complex to handle, as the diffusion operator in the mapped coordinates does not reduce to a diagonal matrix. The geometry consists of a dipole in a half-plane ($x, y > 0$) and the wall is at $y = 0$; the mapping used in our simulations is, with the notations introduced in Subsection 3.2.2,

$$x = \hat{x} f'(\hat{y}) \tag{32}$$

$$y = f(\hat{y}) \tag{33}$$

with

$$f(\hat{y}) = \lambda \left[\exp\left(\frac{\hat{y}}{\lambda}\right) - 1 \right],$$

where λ is a positive parameter. As a result, the grid undergoes a stretching in both directions with a stretching factor of 2 at $y = \lambda$. Straightforward calculations yield

$$B(x, y) = \begin{bmatrix} 1 + \frac{\hat{x}^2}{\lambda^2} & -\frac{\hat{x}}{\lambda} \\ -\frac{\hat{x}}{\lambda} & 1 \end{bmatrix}.$$

The PSE formula (24) was then implemented with a cut-off $\theta(r) = 1/(1 + r^5)$. Velocity evaluations were done by a Biot–Savart law with core size reflecting the volume expansion of the grid, that is,

$$\varepsilon(x, y) = \varepsilon |f'(\hat{y})|^2.$$

In Figs. 10, 11, we compare the results obtained by the variable-blob method with $d\hat{x} = d\hat{y} = 1/256$, and $\lambda = 0.25$ or $\lambda = \infty$ (in the latter case, the blobs have a uniform size). We also show in Fig. 12 the results in the case a low resolution is used everywhere ($\lambda = \infty$, $d\hat{x} = d\hat{y} = 1/128$). The initial condition is a Lamb dipole of unit circulation [14] and the Reynolds number is 400. These results show that, while it is important to have good resolution at the wall to capture accurately the vorticity created there (the last picture in Fig. 11 shows that an insufficient resolution results in a delay in the dynamics), a lower resolution away from the wall does not affect the quality in the results. To conclude this example, let us mention that an extension of a finite-difference scheme to the refined grid would not be straightforward (see [13] for a method in the spirit of finite differences based on B-splines to allow refinement in all directions near the wall).

5.3. *The Case of Multiple Bodies: Local Mappings and Domain Decomposition*

Let us finally give an example where several local mappings are used in order to allow particle refinement near several obstacles. We consider the case of the flow past two cylinders. Around each cylinder particles are remeshed on a polar grid with linear stretching in the radial direction, along the same lines as in Subsection 5.1. These grids extend to 1

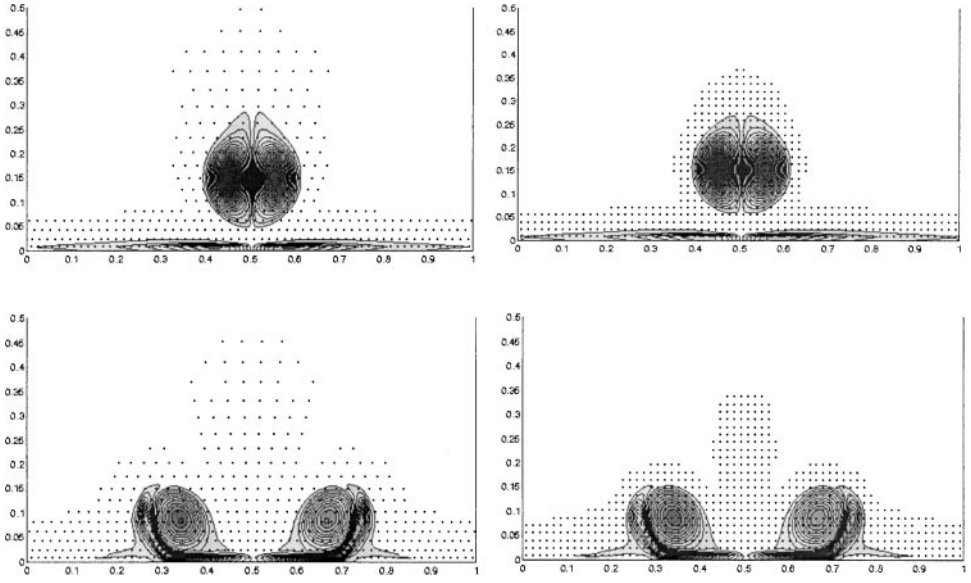

FIG. 10. Successive stages of a vortex dipole impinging on a wall: T = 1.0, 2.0. Right column, uniform blobs, fine grid; left column, variable blobs, coarse to fine grid ratio 2 from $y = 0$ to $y = 0.25$. Particle locations are indicated by dots (for clarity only one-fourth of the particles is shown).

cylinder radius. In between the cylinders, particles are regridded on a Cartesian uniform mesh, with mesh size approximately corresponding to the mesh size of the polar grid in the outer layer (see Fig. 13).

In these calculations we were only interested in the interaction of the wakes with the bodies; Fig. 14 shows that the coupling techniques described in Subsection 4.3 do allow a smooth transfer of vorticity between the sub-domains. Note that for calculations over

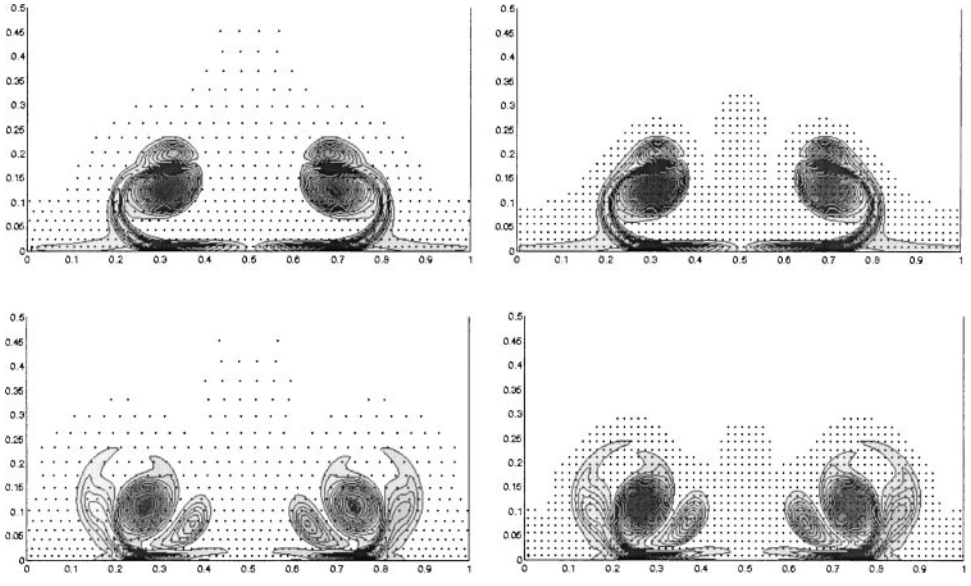

FIG. 11. Successive stages of a vortex dipole impinging on a wall: T = 4.0, 6.0. For description see Fig. 10.

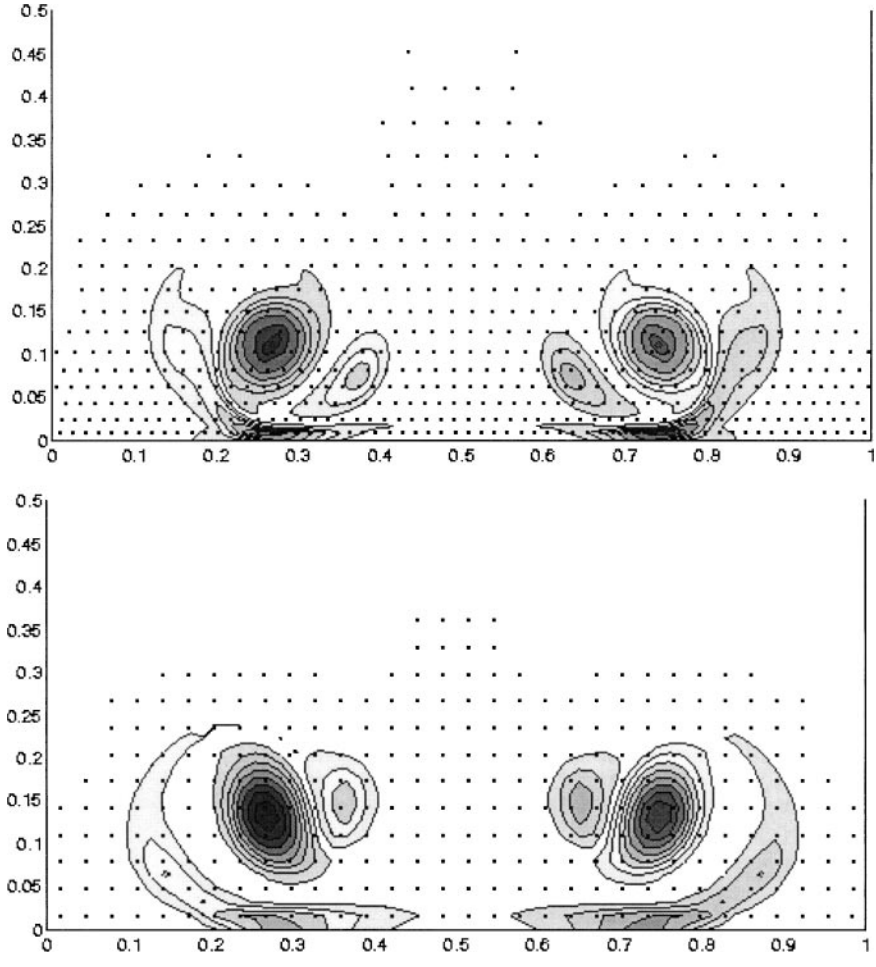

FIG. 12. Rebound of a vortex dipole impinging on a wall at T = 6.0. Bottom, uniform blobs, coarse grid; top, variable blobs, coarse to fine grid ratio 2 from $y = 0$ to $y = 0.25$; right column, uniform blobs, coarse grid.

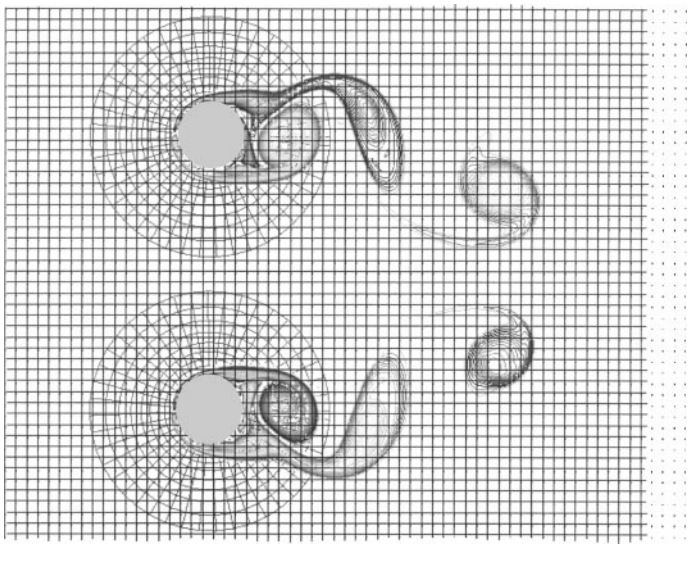

FIG. 13. Flow around two asymmetrically placed cylinders. Grids used for regridding the particles.

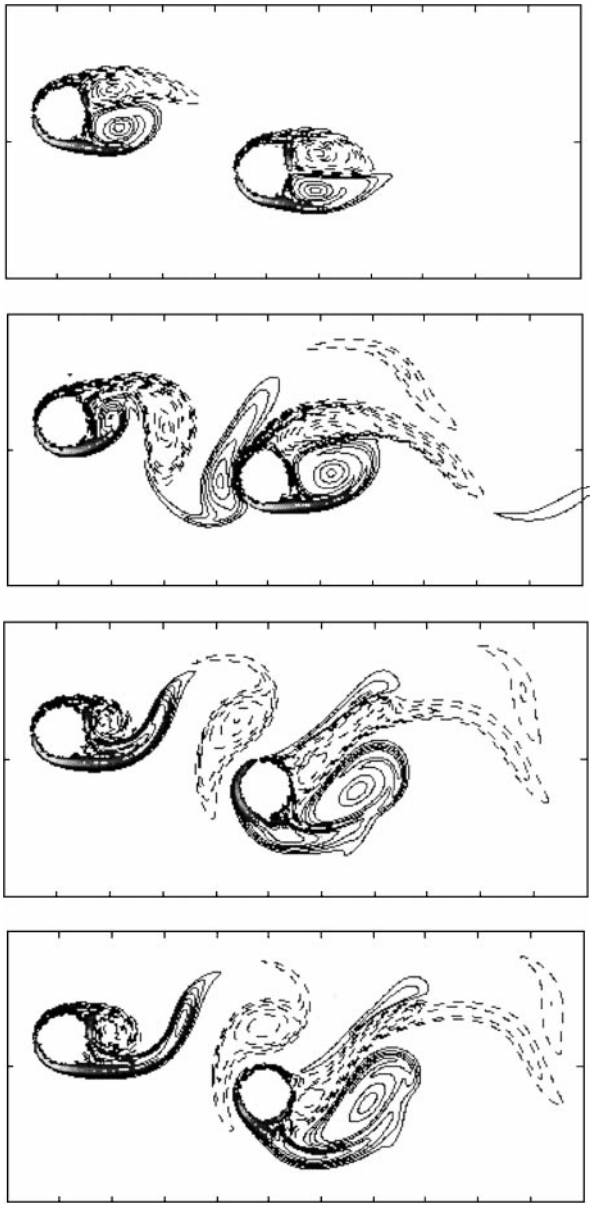

FIG. 14. Flow around two asymmetric cylinders. Successive stages of the vorticity contours.

longer times, with wakes extending far behind the second cylinder, it would be natural and straightforward to implement a third mapping allowing lower resolution down-stream.

6. CONCLUSIONS

In this paper we have presented a new methodology that aims at enhancing the adaptive character of vortex methods. The governing equations for the evolution of vortex particles have been derived for particles of varying blob size. We have also presented the formulation for extending this methodology to flows around multiple bodies.

The results presented herein demonstrate the applicability of this method and the savings that can be obtained in computations using variable size vortex methods. We believe that in the present formulation vortex methods offer an interesting alternative to grid based methodologies. The adaptivity and the Lagrangian character of vortex methods is retained while it is complemented by capabilities such as mesh-embedding.

Present research is aimed at combining effectively in three dimensions grid based and particle methods for different parts of the domain. Variable size vortex methods are a key aspect of this approach.

REFERENCES

1. A. J. Chorin, Numerical study of slightly viscous flow, *J. Fluid Mech*. **57**, 785 (1973).
2. G.-H. Cottet and S. Mas-Gallic, Une méthode de décomposition pour une équation de type convection-diffusion combinant résolution explicite et méthode particularie, *C. R. Acad. Sci. Paris* **297**, 133 (1983).
3. G.-H. Cottet, Particle-grid domain decomposition methods for the Navier–Stokes equations exterior domains, in *Lectures in Applied Mathematics* (Amer. Math. Soc., Providence, 1991), Vol. 28, p. 100.
4. G.-H. Cottet and P. Koumoutsakos, *Vortex Methods: Theory and Practice* (Cambridge Univ. Press, Cambridge, UK, 1999).
5. G.-H. Cottet, M. L. Ould Salihi, and M. El Hamraoui, Multi-purpose regridding in vortex methods, in *ESAIM Proceedings*, in press.
6. P. Degond and S. Mas-Gallic, The weighted particle method for convection-diffusion equations, *Math. Comp*. **53**, 485 (1989).
7. S. Ghosal and P. Moin, The basic equations for large eddy simulation of turbulent flows in complex geometry, *J. Comput. Phys*. **73**, 24 (1995).
8. L. Greengard and V. Rokhlin, A fast algorithm for particle simulation, *J. Comput. Phys*. **118**, 325 (1987).
9. R. Henderson, Nonlinear dynamics and pattern formation in turbulent wake transition, *J. Fluid Mech*. **352**, 65 (1997).
10. T. Y. Hou, Convergence of a variable blob vortex method for the Euler and Navier–Stokes equations, *SIAM J. Numer. Anal*. **27**, 1387 (1990); and private communication.
11. P. Koumoutsakos and A. Leonard, High resolution simulations of the flow around an impulsively started cylinder using vortex methods, *J. Fluid Mech*. **296**, 1 (1995).
12. P. Koumoutsakos, Inviscid axisymmetrization of an elliptical vortex, *J. Comput. Phys.* **138**, 821 (1997).
13. A. G. Kravchenko, P. Moin, and R. Moser, Zonal embedded grids for numerical simulations of wall-bounded flows, *J. Comput. Phys.* **127**, 412 (1996).
14. P. Orlandi, Vortex dipole rebound from a wall, *Phys. Fluids A* **1429**, 75 (1990).
15. M. L. Ould Salihi, G.-H. Cottet, and M. El Hamraoui, Blending finite-difference and vortex methods for incompressible flow simulations, submitted for publication.