AN ADAPTIVE WAVELET COLLOCATION METHOD FOR FLUID-STRUCTURE INTERACTION AT HIGH REYNOLDS NUMBERS[*]

NICHOLAS K.-R. KEVLAHAN[†] AND OLEG V. VASILYEV[‡]

Abstract. Two mathematical approaches are combined to calculate high Reynolds number incompressible fluid-structure interaction: a wavelet method to dynamically adapt the computational grid to flow intermittency and obstacle motion, and Brinkman penalization to enforce solid boundaries of arbitrary complexity. We also implement a wavelet-based multilevel solver for the Poisson problem for the pressure at each time step. The method is applied to two-dimensional flow around fixed and moving cylinders for Reynolds numbers in the range $3 \times 10^1 \leq Re \leq 10^5$. The compression ratios of up to 1000 are achieved. For the first time it is demonstrated in actual dynamic simulations that the compression scales like $Re^{1/2}$ over five orders of magnitude, while computational complexity scales like Re. This represents a significant improvement over the classical complexity estimate of $Re^{9/4}$ for two-dimensional turbulence.

1. Introduction. One of the most practically important problems in aerodynamics is calculating moderate to high Reynolds number incompressible flow around solid obstacles of arbitrary shape. This problem arises in aerodynamics (e.g., turbulent flow over the wings and fuselage of airplanes), in off-shore drilling (e.g., water flow around riser tubes transporting oil from the sea bed to the surface), and in the wind engineering of buildings. In each case the primary difficulty arises from the need to calculate turbulent or transitional flow with boundary conditions on complicated domains. In addition, it may be important to allow for the obstacle to move or deform in response to the applied fluid forces (this motion in turn affects the flow).

In this paper we combine two mathematical approaches to calculate turbulent flow in complex domains. The first technique, the adaptive wavelet collocation method (AWCM), tackles the problem of efficiently resolving a high Reynolds number flow in complicated geometries (where grid resolution should depend on both time and location). The second technique, Brinkman penalization, addresses the problem of efficiently implementing solid boundaries of arbitrary complexity.

The adaptive wavelet method is appropriate for turbulence since the wavelets (which are localized in both space and scale) adapt the numerical resolution naturally to the intermittent structure of turbulence at small scales. The wavelet method thus allows turbulent flows to be calculated with a greatly reduced number of modes while controlling the L^∞-norm error. Furthermore, the computational cost of the algorithm

is independent of the dimensionality of the problem and is $O(\mathcal{N})$, where $\mathcal{N}$ is the total number of wavelets actually used. Wavelet methods for differential equations may be divided into two classes: adaptive wavelet Galerkin methods (AWGMs), for which much rigorous analysis has been done (e.g., [11, 13, 33]), and AWCMs, for which numerical results are more advanced (e.g., [19, 20, 44, 39, 40, 42, 43]). We use the AWCM since it is more straightforward to implement for nonlinear equations and can be modified for a wide class of partial differential equations. The AWCM is implemented using second-generation wavelets, which allows the order of the method to be varied easily (we generally use a sixth-order method). Another advantage of the wavelet collocation approach is that it is straightforward to apply in both two and three dimensions.

Parallel to the development of efficient wavelet codes for turbulence, we have been investigating the use of the Brinkman penalization to simulate the presence of arbitrarily complex solid boundaries [25]. This technique, originally proposed by Arquis and Caltagirone [3], allows boundary conditions to be enforced to a specified precision without changing the numerical method (or grid) used to solve the equations. The main advantage of this method, compared to other penalization-type methods, is that the error can be controlled via a penalization parameter [1]. It has been shown that the solution of the penalized equations converges to the exact solution in the limit as the penalization parameter tends to zero [2]. Because this volume penalization is simple and cheap to calculate, it is well suited to moving obstacles. The AWCM then allows the computational grid to follow the obstacle, thus avoiding accelerating reference frames, or large areas of fine grids.

In previous work [23, 24, 40] we investigated AWCM with penalization of the two-dimensional vorticity equation (i.e., the curl of the Brinkman penalization). Schneider and Farge [32] have also developed such an approach, using vaguelettes. This approach works well for moderate Reynolds numbers ($Re < 10^3$), but is inaccurate for larger Reynolds numbers. We concluded that the error was due to the fact that in the vorticity formulation the penalization is a δ-function surface penalization. This singular forcing (similar to Peskin's approach [30]) is stabilized only by a sufficiently large viscosity. We make some comments on the difference between vorticity and velocity penalization in section 5.4.

In this paper we combine the AWCM with Brinkman penalization to solve the incompressible Navier–Stokes equations in primitive variable form. Due to the fact that the wavelet collocation method does not utilize divergence-free wavelets, we use a standard split-step method in time, where the first step produces a nonsolenoidal velocity field. This intermediate velocity is then made divergence-free by solving a Poisson equation for the pressure. It is well known that solving the Poisson equation is the most computationally intensive part of split-step methods. Thus, in order to solve the Poisson equation efficiently, we developed a multilevel elliptic solver that makes use of the adapted multiscale wavelet collocation grid. The wavelet multilevel elliptic solver will be described in detail in a companion article [41]. The entire numerical method therefore has complexity $O(\mathcal{N})$, where $\mathcal{N}$ is the number of active wavelets or grid points. The resulting method is very general: it can be applied to flow over a large range of Reynolds numbers with or without obstacles of arbitrary shape and motion, and with a variety of boundary conditions. It is easy to modify the spatial order of the method and, since the grid adapts dynamically, there is no need to custom-design meshes for particular problems. The AWCM is described in section 2, while the Brinkman penalization and time scheme are explained in sections 3 and 4.

In section 5 we present results from the AWCM combined with Brinkman penalization. By calculating the flow around a tightly packed periodic array of cylinders, we are able to show that the number of wavelets scales like $Re^{1/2}$ and the overall computational complexity scales like Re for $3 \times 10^1 \leq Re \leq 10^5$. This represents a significant improvement over the usual complexity estimate of $Re^{9/4}$ and confirms the usefulness of adaptive wavelet techniques for high Reynolds number flow. The wavelet/Brinkman method is also applied to fixed cylinders at $Re = 100$ and $Re = 3000$ and to the case where the cylinder motion is coupled to the fluid forces. Because of the large intermittency of these flows we typically obtain compression ratios of between 270 and 1000.

Finally, we summarize the main results of this research in section 6, and make some remarks on the extension of the method to three-dimensional flows. The high compressions achieved and the fact that the number of grid points scales like $Re^{1/2}$ over many orders of magnitude suggest that our method is promising for high Reynolds number turbulence.

2. Adaptive wavelet collocation method. The numerical method is formally derived by evaluating the governing partial differential equations at collocation points, which results in a system of nonlinear ordinary differential-algebraic equations describing the evolution of the solution at these collocation points. The use of a collocation method means that the calculation of nonlinear terms can be done more simply than with wavelet Galerkin methods [4, 7, 19, 22, 28]. However, Cohen, Dahmen, and DeVore [11] and Dahmen, Schneider, and Xu [13] have recently developed an improved AWGM to evaluate nonlinear functions of wavelet expansions, and this technique could prove useful for the advection term of the Navier–Stokes. In order for the algorithm to resolve all the structures appearing in the solution and yet be efficient in terms of minimizing the number of unknowns, the computational grid should adapt dynamically in time to reflect the local structure of the solution; i.e., high resolution computations should be carried out only in those regions with sharp gradients.

In this section we briefly review the dynamically adaptive wavelet collocation method (for more details, see [44, 39]). In particular, we will sketch efficient wavelet-based procedures for dynamic grid adaptation and calculation of spatial derivatives.

2.1. Grid adaptation. Grid adaptation occurs naturally in wavelet methods. To illustrate the algorithm, let us consider a function $f(\mathbf{x})$, defined on a closed n-dimensional rectangular domain Ω. We use tensor product second-generation wavelets [36, 37], which are a generalization of biorthogonal wavelets [14, 12] and are more easily applied to functions defined on domains more general than R^n. Second-generation wavelets form a Reisz basis for $\mathbf{L}^2$ space, with the wavelets being local in both space and frequency and having many vanishing polynomial moments, but without the translation and dilation invariance of their biorthogonal cousins. Despite the loss of two fundamental properties of wavelet bases, second-generation wavelets retain many of the useful features of biorthogonal wavelets, including the existence of a fast transform. Second-generation wavelets are constructed on a set of grids,

$$(2.1) \qquad \mathcal{G}^j = \left\{ \mathbf{x}_{\mathbf{k}}^j \in \Omega : \mathbf{k} \in \mathcal{K}^j \right\}, \quad j \in \mathcal{J},$$

where $\mathbf{k} = (k_1, \ldots, k_n)$ and grid points $\mathbf{x}_{\mathbf{k}}^j = (x_{1,k_1}^j, \ldots, x_{n,k_n}^j)$ are constructed as a tensor product of uniformly or nonuniformly spaced one-dimensional grids. The only restriction is that each individual set of one-dimensional grids is nested ($x_{m,k_l}^j = x_{m,2k_l}^{j+1}$, $m = 1, \ldots, n$), which guarantees the nestedness of the grids, i.e., $\mathcal{G}^j \subset \mathcal{G}^{j+1}$.

The procedure of constructing n-dimensional scaling functions $\phi_{\mathbf{k}}^j(\mathbf{x})$ and a family of n-dimensional wavelets $\psi_{\mathbf{l}}^{\mu,j}(\mathbf{x})$, $\mu = 1, \ldots, 2^n - 1$, on an n-dimensional dyadic grid is described in [44]. Once wavelets and scaling functions are constructed, a function $f(\mathbf{x})$ can be decomposed as

$$(2.2) \qquad f(\mathbf{x}) = \sum_{\mathbf{k} \in \mathcal{K}^0} c_{\mathbf{k}}^0 \phi_{\mathbf{k}}^0(\mathbf{x}) + \sum_{j=0}^{+\infty} \sum_{\mu=1}^{2^n-1} \sum_{\mathbf{l} \in \mathcal{L}^{\mu,j}} d_{\mathbf{l}}^{\mu,j} \psi_{\mathbf{l}}^{\mu,j}(\mathbf{x}).$$

For functions which contain isolated small scales on a large-scale background, most wavelet coefficients are small; thus we retain a good approximation even after discarding a large number of wavelets with small coefficients. Intuitively, the coefficient $d_{\mathbf{l}}^{\mu,j}$ is small unless the $f(\mathbf{x})$ has variation on the scale of j in the immediate vicinity of wavelet $\psi_{\mathbf{l}}^{\mu,j}(\mathbf{x})$. In fact, the error incurred by ignoring coefficients with magnitude lower than ϵ is $O(\epsilon)$. More precisely, if we rewrite (2.2) as a sum of two terms composed respectively of wavelets whose amplitude is above and below some prescribed threshold ϵ, i.e., $f(\mathbf{x}) = f_{\geq}(\mathbf{x}) + f_{<}(\mathbf{x})$, then it can be shown [15, 44, 39] that

$$(2.3) \qquad |f(\mathbf{x}) - f_{\geq}(\mathbf{x})| \leq C_1 \epsilon \leq C_2 \mathcal{N}^{-p/n},$$

where $\mathcal{N}$ is the number of significant wavelet coefficients, p is the order of the wavelets, and n is the dimensionality of the problem.

In order to realize the benefits of the wavelet compression, we need to be able to reconstruct $f_{\geq}(\mathbf{x})$ from the subset of $\mathcal{N}$ grid points. Note that every wavelet $\psi_{\mathbf{l}}^{\mu,j}(\mathbf{x})$ is uniquely associated with a collocation point. Consequently, the collocation point should be omitted from the computational grid if the associated wavelet is omitted from the approximation. This procedure results in a set of nested adaptive computational grids $\mathcal{G}_{\geq}^j \subset \mathcal{G}^j$, such that $\mathcal{G}_{\geq}^j \subset \mathcal{G}_{\geq}^{j+1}$ for any $j < J - 1$, where J is the finest level of resolution present in approximation $f_{\geq}(\mathbf{x})$.

When solving the evolution equations an additional criterion for grid adaptation should be added. The computational grid should consist of grid points associated with wavelets whose coefficients are significant or could become significant during a time step. In other words, at any instant in time, the computational grid should include points associated with wavelets belonging to an *adjacent zone* of wavelets for which the magnitude of their coefficients is greater than an a priori prescribed threshold ϵ. This point is discussed in greater detail in section 2.3.

2.2. Calculation of spatial derivatives on an adaptive grid. When solving partial differential equations numerically, it is necessary to obtain derivatives of a function from its values at collocation points. In this section we describe an efficient procedure for calculating spatial derivatives [39, 44] which takes advantage of the multiresolution wavelet decomposition and the fast wavelet transform and uses finite difference differentiation. In other words, we make wavelets do what they do well: compress and interpolate. Finite differences do the rest: differentiate polynomials.

The differentiation procedure is based on the interpolating properties of second-generation wavelets. We recall that wavelet coefficients $d_{\mathbf{l}}^{\mu,j}$ measure the difference between the approximation of the function at the $j + 1$ level of resolution and its representation at the j level of resolution. Thus if there are no points in the immediate vicinity of a grid point $\mathbf{x}_{\mathbf{k}}^j$, i.e., $|d_{\mathbf{m}}^{\mu,j}| < \epsilon$ for all the neighboring points, and points $\mathbf{x}_{(2k_1 \pm 1, \ldots, 2k_n \pm 1)}^{j+1}$ are not present in $\mathcal{G}_{\geq}^{j+1}$, then there exists some neighborhood of $\mathbf{x}_{\mathbf{k}}^j$, where the actual function is well approximated by a wavelet interpolant based on $c_{\mathbf{m}}^j$.

Thus differentiating this local polynomial gives us the value of the derivative of the function at that particular location. Let us denote by $\mathcal{D}_{\geq}^{j}$ a collection of such points at j level of resolution. Then the procedure for finding derivatives at all grid points consists of the following steps. First, knowing the values of a function on an adaptive computational grid $\mathcal{G}_{\geq}$, perform wavelet transform. Next, recursively reconstruct the function starting from the coarsest level of resolution. On each level of resolution j find derivatives of the function at grid points that belong to $\mathcal{D}_{\geq}^{j}$. At the end of the inverse wavelet transform we have derivatives of the function at all grid points. The computational cost of calculating spatial derivatives is roughly the same as the cost of forward and inverse wavelet transforms.

2.3. Numerical algorithm. The three basic steps of the numerical algorithm are then as follows (bold symbols denote n-dimensional vectors $\mathbf{u} \equiv (u_1, \ldots, u_n)$ and $\mathbf{k} \equiv (k_1, \ldots, k_n)$):

1. Knowing the values of the solution $\mathbf{u}_{\mathbf{k}}^{J}(t)$, we compute the values of wavelet coefficients corresponding to each component of the solution using the fast wavelet transform. For a given threshold ϵ we adjust $\mathcal{G}_{\geq}^{t+\Delta t}$ based on the magnitude of the wavelet coefficients, assigning a value $d_{\mathbf{k}}^{\mu,j} = 0$ for the new grid points. As described below, we then add nearest neighbors in position and scale to account for the change in the solution from t to $t + \Delta t$.
2. If there is no change between computational grids $\mathcal{G}_{\geq}^{t}$ and $\mathcal{G}_{\geq}^{t+\Delta t}$ at time t and $t + \Delta t$, we go directly to step 3. Otherwise, we compute the values of the solution at the collocation points $\mathcal{G}_{\geq}^{t+\Delta t}$, which are not included in $\mathcal{G}_{\geq}^{t}$.
3. We integrate the resulting system of ordinary differential equations to obtain new values $\mathbf{u}_{\mathbf{k}}^{J}(t + \Delta t)$ at positions on the irregular grid $\mathcal{G}_{\geq}^{t+\Delta t}$ and go back to step 1.

With such an algorithm the grid of collocation points adapts dynamically in time to follow local structures that appear in the solution. Note that by omitting wavelets with coefficients below a threshold parameter ϵ we automatically control the L^{∞}-norm error of approximation (the L^{∞}-norm error of the reconstruction is bounded by ϵ). Thus the wavelet collocation method has another important feature: active control of the accuracy of the solution. The smaller ϵ is chosen to be, the smaller the error of the solution is. (Note that ϵ is properly normalized for each variable so that for a variable f, $\epsilon_f = \epsilon ||f||_{\infty}$.) In typical applications the normalized value of ϵ varies between 10^{-2} and 10^{-5}. As the value of ϵ increases, fewer grid points are used in the solution.

The method is made truly dynamical by allowing for the change in the solution over one time step by adding nearest neighbor wavelets (i.e., grid points) in position and scale. Adding nearest neighbors in position corresponds to a CFL criterion of one. Since the scales are dyadic, adding nearest neighbors in scale means allowing for the creation of scales twice as small via the quadratic nonlinearity. Because grid adaptation is very cheap, it is done at each time step, which means the grid closely follows the strong gradients of the flow. Without the addition of nearest neighbors the method would be equivalent to an nonlinear wavelet compression of the solution at each time step (a sort of "image" compression). In our method we adapt only on the velocity field (which produces good results), but we could also adapt on other quantities (such as scalar concentration). A typical dynamically adapted grid is shown at each scale in Figure 5.5. Note how the finest grids are needed only in the boundary layer.

Refinement of the grid based on nonlinear wavelet filtering is the basis of the coherent vortex simulation (CVS) method [16, 33] and stochastic coherent adaptive large eddy simulation (SCALES) method [18]. In CVS, it is proposed that large values of ϵ (e.g., $\epsilon = 10^{-2}$) could be used if the effect of the neglected wavelets is modelled stochastically. This should be especially straightforward for two-dimensional turbulence since Farge, Schneider, and Kevlahan [16] showed that the neglected modes have Gaussian statistics. In SCALES, it is proposed to use even larger values of ϵ, so only the most energetic part of the coherent vortices is simulated in the resolved field. The effect of the discarded wavelets is then modelled using the dual stochastic-deterministic model [18]. In this paper we do not use such subgrid-scale models and therefore must use a relatively small threshold, $\epsilon = 10^{-4}$.

3. Brinkman penalization. Let us consider a viscous incompressible fluid governed by the Navier–Stokes equations:

$$(3.1) \qquad \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} + \mathbf{U}) \cdot \nabla \mathbf{u} + \nabla P = \nu \Delta \mathbf{u},$$

$$(3.2) \qquad \nabla \cdot \mathbf{u} = 0,$$

where $\mathbf{u}$ is the velocity, P is the pressure, and $\mathbf{U}$ is an imposed mean flow. We consider here the case where the fluid occupies the complement in the plane R^2 of a set of obstacles O_i. The problem is solved in a rectangular domain $\Omega = [L_1, L_2] \times [M_1, M_2]$ containing all the obstacles O_i. To these equations are added appropriate external boundary conditions (inflow, outflow, and side), which are discussed further below.

On the surface of the obstacles the velocity must satisfy the no-slip condition,

$$(3.3) \qquad \mathbf{u} + \mathbf{U} = \mathbf{U}_o \quad \text{on } \partial O_i \ \forall i,$$

where $\mathbf{U}_o$ is the velocity of the obstacle. To model the effect of the no-slip boundary conditions on the obstacles O_i without explicitly imposing (3.3), we follow Angot, Bruneau, and Fabrie [1] by replacing (3.1)–(3.3) by the following set of L^2-penalized equations:

$$(3.4) \qquad \frac{\partial \mathbf{u}_\eta}{\partial t} + (\mathbf{u}_\eta + \mathbf{U}) \cdot \nabla \mathbf{u}_\eta + \nabla P_\eta = \nu \Delta \mathbf{u}_\eta - \frac{1}{\eta} \chi(\mathbf{x}, t)(\mathbf{u}_\eta + \mathbf{U} - \mathbf{U}_o),$$

$$(3.5) \qquad \nabla \cdot \mathbf{u}_\eta = 0,$$

where $\mathbf{U}_o$ is the obstacle's velocity. Note that (3.4)–(3.5) are valid in the entire domain Ω: the last term on the right-hand side of (3.4) is a volume penalization of the flow inside the obstacle. Here $0 < \eta \ll 1$ is a penalization coefficient, and χ denotes the characteristic (or mask) function

$$(3.6) \qquad \chi(\mathbf{x}, t) = \begin{cases} 1 & \text{if } \mathbf{x} \in O_i, \\ 0 & \text{otherwise.} \end{cases}$$

Angot [2] proved that the solution of the penalized equations (3.4)–(3.5) converges to that of the Navier–Stokes equations (3.1)–(3.2) with the correct boundary conditions (3.3) as $\eta \to 0$. More precisely, the upper bound on the global error of the L^2-penalization was shown to be [2]

$$(3.7) \qquad \|\mathbf{u} - \mathbf{u}_\eta\|_{H^1(\Omega)} = O(\eta^{1/4}).$$

In the specific case of impulsively started flow over a plane, Kevlahan and Ghidaglia [25] showed that the error is actually lower: $O(\eta^{1/2})$. It seems reasonable that this is the sharp estimate in the general case as well.

This volume penalization has been implemented in a finite difference code [26] for two-dimensional flow around a cylinder and was found to give good results. In fact, we find that the actual numerical error is slightly better, $O(\eta)$. It is important to note that η is an arbitrary parameter, independent of the spatial or temporal discretization, and thus the boundary conditions can be enforced to any desired accuracy by choosing η appropriately. This property distinguishes the Brinkman method from other penalization schemes and allows the error to be controlled precisely. We have found that $\eta = 10^{-4}$ gives drag curves correct to about 1%.

Another advantage of the Brinkman penalization is that the force $\mathbf{F}_i$ acting on an obstacle O_i can be found by simply integrating the penalization term over the volume of the obstacle:

$$(3.8) \qquad \mathbf{F}_i = \frac{1}{\eta} \int_{O_i} (\mathbf{u} + \mathbf{U} - \mathbf{U}_o)\, d\mathbf{x}.$$

Thus, the calculation of lift and drag on an obstacle can be made simply, accurately, and at low cost. This is helpful when calculating fluid-structure interaction, where the force must be updated at each time step.

Although it is a very flexible and simple method, Brinkman penalization does have two drawbacks. First, the large factor $1/\eta$ means that the term is stiff and must be solved implicitly. This is not difficult since the penalization term is simply a mask times the velocity. A stiffly stable time scheme for the penalized equations is described in the following section. The second drawback is that since the penalized equations are defined over the whole domain they must also be solved inside the obstacle. This adds some extra computational work, but since the flow penetrates to a skin-depth of only $O(\eta^{1/2})$ [25] there are few grid points inside the obstacle. The sparseness of the grid inside the obstacle is clear in Figure 5.1(b).

4. Time integration and Poisson solver. Our goal is to solve the following system of equations describing coupled fluid-structure interaction:

$$(4.1) \qquad \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} + \mathbf{U}) \cdot \nabla \mathbf{u} + \nabla P = \nu \Delta \mathbf{u} - \frac{1}{\eta} \chi(\mathbf{x}_o)(\mathbf{u} + \mathbf{U} - \mathbf{U}_o),$$

$$(4.2) \qquad \nabla \cdot \mathbf{u} = 0,$$

$$(4.3) \qquad m \frac{d^2 \mathbf{x}_o}{dt^2} + b \frac{d\mathbf{x}_o}{dt} + k\mathbf{x}_o = \mathbf{F}(t),$$

where the third equation models the motion of the obstacle's center of mass $\mathbf{x}_o$ as a damped harmonic oscillator forced by the fluid-induced force $\mathbf{F}$ (calculated using (3.8)). The obstacle equation is integrated in time using a second-order Adams–Bashforth scheme.

For the purposes of illustration, let us consider the first-order time integration of the penalized Navier–Stokes equations (4.1)–(4.2) (in practice, we use the similar second-order method mentioned at the end of this section). The time scheme is based on a split-step where a nonsolenoidal velocity field is calculated in the first step and is made divergence-free using a pressure projection in the second step. To calculate the velocity $\mathbf{u}(\mathbf{x}, t_{n+1}) \equiv \mathbf{u}_{n+1}$ given the velocity at the previous time step $\mathbf{u}_n$, we

first solve for the nonsolenoidal velocity $\mathbf{u}_*$:

$$\mathcal{L}\mathbf{u}_* \equiv \left(\frac{1}{\Delta t} + \frac{1}{\eta}\chi(\mathbf{x}_o)\right)\mathbf{u}_* + (\mathbf{u}_n + \mathbf{U}) \cdot \nabla\mathbf{u}_* + \nu\Delta\mathbf{u}_*$$

(4.4)
$$= \frac{\mathbf{u}_n}{\Delta t} - \frac{1}{\eta}\chi(\mathbf{x}_o)(\mathbf{U} - \mathbf{U}_o),$$

where $\mathcal{L}$ is a linear operator. Note that the Laplacian and penalization terms are discretized implicitly (as these are stiff terms), while the advection term is discretized semi-implicitly. The semi-implicit discretization allows the use of a fine grid in the boundary layer, without having to use an unreasonably small time step because of the CFL limit (we typically enforce a CFL criterion of one). The linear system (4.4) is solved using the conjugate gradient method BI-CGSTAB [38]. Note that the vorticity is correct after the first step: $\omega_{n+1} = \omega_* = \nabla \times \mathbf{u}_*$. Since the vorticity is correct at this stage, we calculate the force using $\mathbf{u}_*$ in (3.8). This avoids errors induced by the slight smoothing at the boundary caused by correcting $\mathbf{u}_*$ in the pressure projection step below.

The presence of the Laplacian operator means that $\mathbf{u}_*$ is differentiable after this step, even though the penalization term is discontinuous [25]. If we did not use a split-step, the Poisson problem for the pressure would involve the divergence of the penalization term which would formally introduce δ-function singularities at the edge of the obstacle. The split-step method is thus a natural choice when using Brinkman penalization.

In the second step we correct $\mathbf{u}_*$ by making it divergence-free using the following pressure projection:

(4.5)
$$\mathbf{u}_{n+1} = \mathbf{u}_* - \Delta t \nabla P_{n+1},$$

where P_{n+1} satisfies

(4.6)
$$\nabla \cdot \nabla P_{n+1} = \frac{1}{\Delta t}\nabla \cdot \mathbf{u}_*.$$

The combination of (4.5) and (4.6) ensures that the velocity at t_{n+1} is divergence-free, i.e., $\nabla \cdot \mathbf{u}_{n+1} = 0$. All derivatives are calculated to sixth-order accuracy using wavelets on the adapted grid, as described in section 2.2. Because we use a collocation scheme, pressure and velocity are given at the same grid points. This is the simplest method for nonuniform grids, but it means that we have to be careful when solving (4.6). In particular, to avoid the odd-even decoupling instability associated with the nonstaggered grid, we define the Laplacian operator as the inner product of a downwind gradient operator and an upwind divergence operator. Griebel and Koster [19] independently developed a similar upwind/downwind decomposition in their Galerkin approach for the incompressible Navier–Stokes equations.

The Poisson equation (4.6) is solved using a multilevel method with V-cycles [9, 35]. The grids on multiple levels are provided by the adaptive wavelet multiresolution, which produces a natural adaptive method for the Poisson equation. Wavelets are also used to interpolate between the grids at multiple levels. This is the first time an AWCM multilevel solver using second-generation wavelets has been developed, and it is briefly described in the appendix and fully in [41]. Note that since it is based on the wavelet transform, the elliptic solver also has complexity $O(\mathcal{N})$. An early AWCM using the Deslaurier–Dubuc interpolating bases was proposed by Bertoluzza [6]; the AWGM has also been applied to solve problems involving elliptic operators [10].

The split-step time scheme is stiffly stable, and can be solved with either periodic or nonperiodic boundary conditions. In the latter case, the consistent boundary conditions for (4.6) are $\nabla P \cdot \mathbf{n} = \mathbf{u} \cdot \mathbf{n}$, where $\mathbf{n}$ is the outward normal vector of the computational domain. The actual velocity boundary conditions used are described in section 5 below.

In practice, we use a similar second-order time scheme which uses the second-order backward difference formula for the time derivative and a suitable second-order semi-implicit discretization of the convective term. To reduce the effect of the numerical boundary layer at the edge of the computational domain when using nonperiodic boundary conditions, we use the rotational form of the split-step method proposed by Guermond and Shen [21].

The following section presents results for two-dimensional flow around fixed and moving cylinders for $3 \times 10^1 \le Re \le 10^5$. The results are compared with other numerical simulations and experimental results, and the scaling of number of grid points and computational complexity with Reynolds number is measured.

5. Results.

5.1. Scaling with Reynolds number. We begin with a series of simulations of the impulsively started flow through a tightly packed cylinder array over a large range of Reynolds numbers, $3 \times 10^1 \le Re \le 10^5$. To model a large array of cylinders (such as found in the tube bundle of a heat exchanger) we consider one periodic cell where the ratio of cylinder separation to diameter $P/D = 1.5$. The Reynolds number is defined in the usual way as $Re = UD/\nu$, where $U = |\mathbf{U}|$ is the imposed mean flow, which is at an angle of $45°$ to the array. We ensure that the boundary layer is fully resolved by setting the smallest grid spacing $\Delta x_{\min} = Re^{-1/2}/6$, i.e., six points across the boundary layer or Taylor scale $\lambda = Re^{-1/2}$. Note that in principle we do not need to limit the maximum resolution (this is determined automatically by the threshold ϵ). We do so here simply to conserve computational resources, since the smallest scales at the solid boundary are associated with the penalization and are not physical. Each simulation is performed until $t = 1$, by which time the boundary layer is completely formed and trailing vortices begin to appear. The goal of this investigation is to determine how $\mathcal{N}$ (number of active grid points) and overall complexity scale with Reynolds number. We will verify the claim that adaptive wavelet techniques are well suited to calculating intermittent high Reynolds number flows.

Figure 5.1 shows the vorticity and adapted grid at $Re = 10^4$. The vorticity is very smooth, and shows no sign of instability or Gibb's oscillation associated with the discontinuity at the surface of the obstacle. The grid is fine only in the thin boundary layer of width $\lambda \approx Re^{-1/2} \approx 0.01$. For an L^∞-error of 10^{-4} we require only 66862 grid points out of a maximum of 869^2, which corresponds to a compression ratio of 12. In particular, there are few points in the interior of the cylinder beyond the skin-depth of thickness about 0.05. This result demonstrates that our approach gives smooth vorticity and a sparse grid even at relatively large Reynolds numbers.

It was claimed [16] that the wavelet-based CVS method should perform well at high Reynolds numbers, because the wavelets would automatically concentrate the grid points on the intermittent structures of the flow. Since turbulence becomes increasingly intermittent at high Reynolds numbers, CVS should become increasingly efficient as Reynolds number increases. We now check this claim by measuring how $\mathcal{N}$ (number of active wavelets or grid points), compression ratio, time step, and overall computational complexity scale with Reynolds number. Figure 5.2(a) shows that the number of active grid points and the compression ratio scale like $\lambda \propto Re^{1/2}$, while
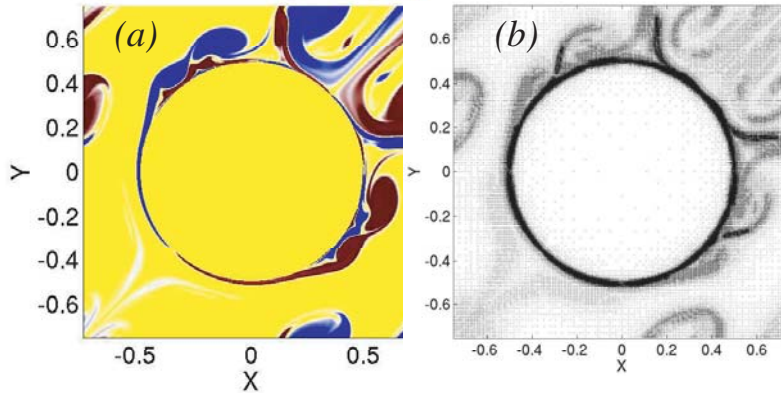
FIG. 5.1.   *Two-dimensional periodic cylinder array at* $Re = 10^4$, $t = 3.5$.   (a) *Vorticity.*
(b) *Adapted grid.*

Figure 5.2(d) shows that computational complexity scales like Re. These results support the original claim, since intermittency scales roughly like λ. Note that the overall complexity of the wavelet calculation increases much more slowly than the usual classical scaling estimate of $Re^{9/4}$ for two-dimensional turbulence based on the Kolmogorov scale, or $Re^{3/2}$ based on λ. Furthermore, the scaling law is constant over five orders of magnitude. However, for $Re > 10^6$ (when the boundary layer itself becomes turbulent) the scaling law may change. This will be the subject of future investigation.

5.2. Fixed cylinder at $Re = 100$. We now consider the case of an isolated cylinder in a large domain of size $[-20, 40] \times [-15, 15]$ (cylinder diameter $D = 1$). To mimic an isolated cylinder we employ periodic boundary conditions in the spanwise direction, and decaying Robin boundary conditions at inflow ($\partial\mathbf{u}/\partial x = \mathbf{u}$) and outflow ($\partial\mathbf{u}/\partial x = -\mathbf{u}$). We checked that increasing the spanwise dimension did not significantly affect the results. To avoid instabilities associated with small fluctuations, we use a uniform grid of resolution λ and width 1 at the inflow and outflow (this buffer zone increases the number of grid points by less than 10%). The finest resolution is $\Delta x = \lambda/6$, which corresponds to a finest grid of 3584×1792. As in the previous section, the flow is started impulsively at $t = 0$. The step-size is chosen to satisfy a CFL criterion of one.

Figure 5.3 shows the vorticity and grid at $t = 150$ when the von Karman vortex street is well established. Note that the grid follows the vortices as they are shed downstream and that vortices exit the computational domain cleanly with no reflection or distortion. The companion figures show a close-up of the vorticity and computational grid. Again, the vorticity is smooth, and there are few points inside the obstacle. The grid also coarsens quickly away from the boundary layer. Figure 5.4 shows that most points are near the Taylor scale λ (within an order of magnitude). This is an indication that λ is the most dynamically important scale for this flow.

To get a better idea of how the adaptive wavelet algorithm distributes points in position and scale we have plotted the grid at each scale in Figure 5.5. This figure shows clearly that the finest scales ($\lambda/6$) are required only in the boundary layer, and that the grid gradually coarsens with the diffusion of vorticity as the vortices move downstream. Figure 5.6 shows the evolution of compression as the von Karman
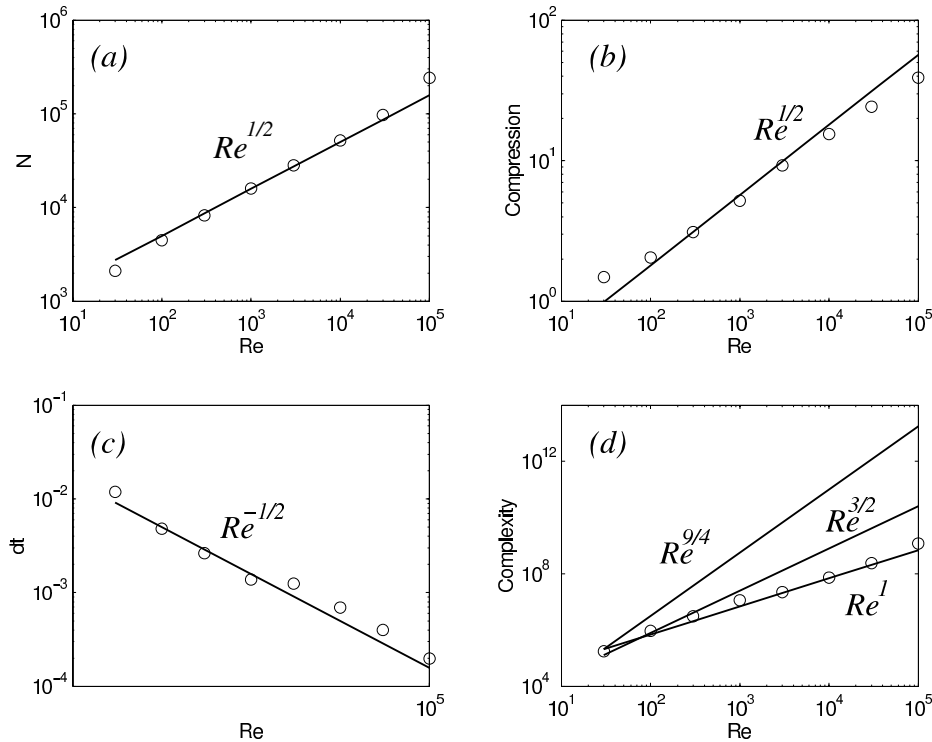
FIG. 5.2. *Scaling for tightly packed cylinder array.* (a) *Number of active grid points.* (b) *Compression ratio.* (c) *Time step.* (d) *Complexity compared with classical scaling based on the Kolmogorov scale $\eta = Re^{-3/4}$ (which gives $Re^{9/4}$ in two dimensions), and the Taylor scale λ (which gives $Re^{3/2}$ in two dimensions).*

vortex street develops. Initially, the compression is very high, 470, before dropping to a quasi-stationary level of about 270 when the vortex shedding is well developed. The adaptive wavelet collocation code is about four times slower per grid point than the FFTW-based spectral code used in previous work [25], so a compression factor of 270 represents an acceleration of about 68 times compared with the spectral method.

Finally, in Figure 5.7 we plot the lift and drag for the cylinder calculated using (3.8). The average drag during the shedding phase is $C_D = 1.35$, the lift amplitude is $C_L = 0.27$, and the Strouhal number is $St = 0.168$. Shiels, Leonard, and Roshko [34] used their highly accurate vortex method to investigate the same flow and found $C_D = 1.33, C_L = 0.3$, and $St = 0.167$, which are close to the present results. Persillon and Braza [29] used a finite-volume scheme and found slightly different numbers: $C_D = 1.25, C_L = 0.37$, and $St = 0.164$. Williamson's laboratory experiments [45] gave $St = 0.164$. Thus, the present method gives results for fluid forces and vortex shedding frequency that are consistent with other numerical and laboratory results.

5.3. Moving cylinder at $Re = 100$. We now modify the simulations of the previous section by allowing the cylinder to move in response to fluid forces. We model the interaction using the harmonic oscillator equation (4.3) with zero damping ($b = 0$), nondimensional frequency $k^* = k/(1/2\rho U^2) = 8.74$, and mass $m^* = m/(1/2\rho D^2) - \pi/2 = 5$ ($\pi/2$ is the normalized added-mass associated with the volume of fluid displaced by the cylinder). The parameters were chosen to match a
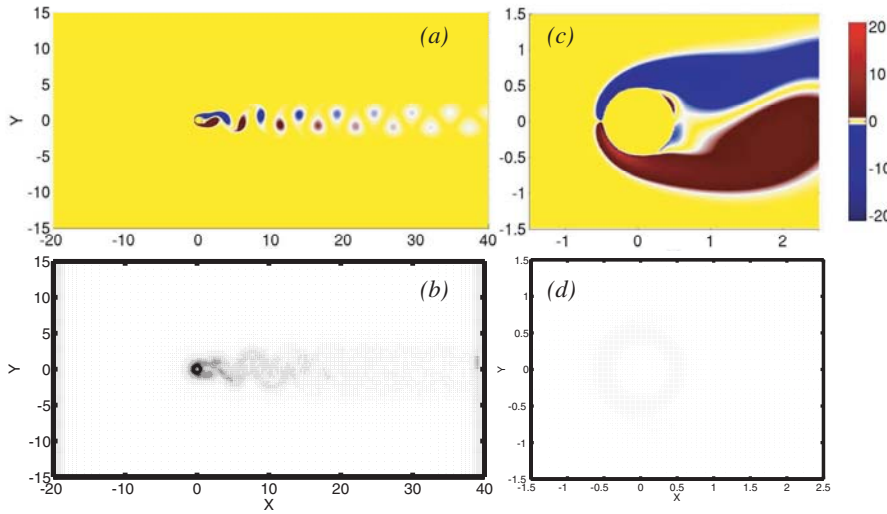
FIG. 5.3. *Two-dimensional fixed cylinder at* $Re = 100$, $t = 150$. (a) *Vorticity.* (b) *Adapted grid.* (c) *Close-up of vorticity.* (d) *Close-up of grid.*
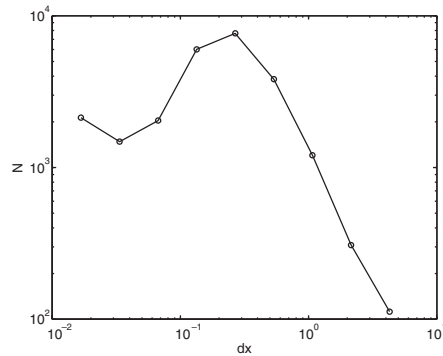

FIG. 5.4. *Number of grid points as a function of grid size for fixed cylinder at* $Re = 100$. *The grid size* $\Delta x = Lx/(14 \times 2^{j-1})$, *where* j *is the scale. Note that most grid points are near the Taylor scale* $Re^{-1/2} = 0.1$.

calculation performed by Shiels, Leonard, and Roshko [34]. The results in this section demonstrate the flexibility of our combined adaptive wavelet collocation-Brinkman penalization method: the cylinder can be moved by simply updating the mask function χ, without the need for accelerating reference frames or expensive remeshing. We find that it is only slightly more expensive to calculate a moving cylinder than a fixed cylinder.

The vorticity and grid for the moving cylinder are presented in Figure 5.8. Note that the cylinder is in its upward phase and that the grid has followed the movement of the cylinder as well as the advection of the vorticity. Figure 5.9 shows the lift, drag, and oscillation amplitude as a function of time. The oscillation amplitude $A = 0.42$, lift amplitude $C_L = 0.81$, and the Strouhal frequency $St = 0.189$ are reasonably close
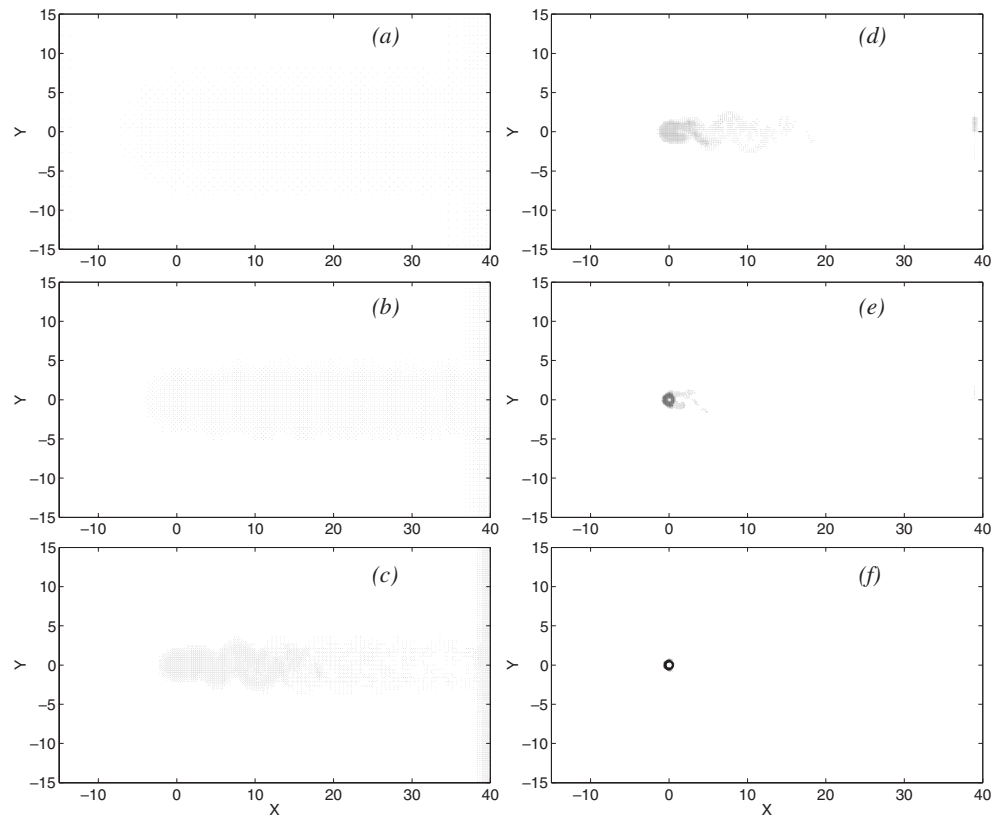
FIG. 5.5. *Computational grid at different scales* 2^{-j}. (a) $j = 4$, *2224 points.* (b) $j = 5$, *4836 points.* (c) $j = 6$, *7794 points.* (d) $j = 7$, *2977 points.* (e) $j = 8$, *1810 points.* (f) $j = 9$, *2261 points.*

to the vortex method values of $A = 0.57$, $C_L = 0.83$, $St = 0.194$ in Shiels, Leonard, and Roshko [34]. However, our average drag coefficient is much lower: $C_D = 1.74$ compared to $C_D = 2.26$. Since the lift coefficients are very close (within 2.5%), it is possible that they may have modified their definition of drag in the case of a moving cylinder.

This section has shown the usefulness of our method for investigating fluid-structure interaction. The calculation of moving obstacles is just as simple and essentially as fast as calculating fixed obstacles. It would be straightforward to apply the method to different shapes or multiple moving obstacles by a suitable modification of the mask function χ. One could similarly allow the obstacles to rotate or deform in response to fluid forces.

5.4. Fixed cylinder $Re = 3000$. Our final example is the impulsively started flow around a cylinder at $Re = 3000$. This example shows the accuracy of our method in calculating the singular start-up flow and the sensitivity of our AWCM to strong gradients in the flow. Since we are interested only in the initial flow, we use doubly periodic boundary conditions and a large domain of size $[-10, 10] \times [-10, 10]$ with a maximum resolution of 6144^2 (which corresponds to $\lambda/5.6$). We maintain a CFL criterion of one, which gives a time step of about $\Delta t = 2 \times 10^{-3}$.

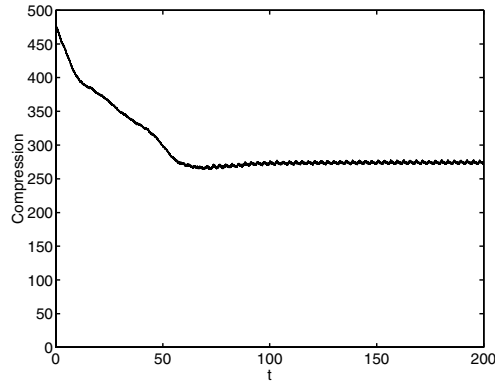We show a close-up of the vorticity in Figure 5.10. The vorticity field is similar

FIG. 5.6. *Compression for fixed cylinder at $Re = 100$ as a function of time. The average compression ratio is about 274 (corresponding to 23400 grid points) during the periodic shedding regime.*
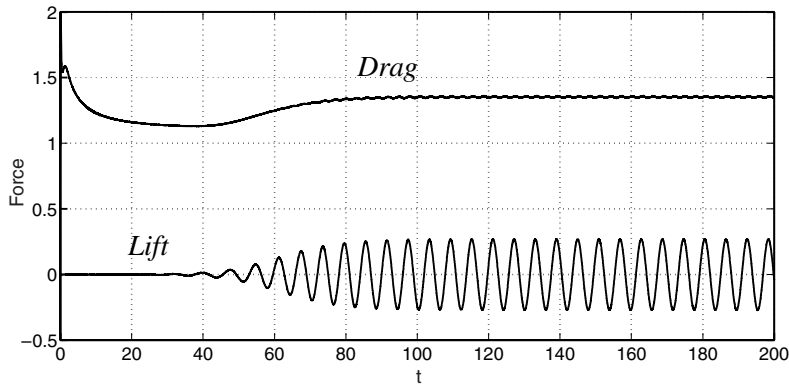

FIG. 5.7. *Lift and drag for a fixed cylinder at $Re = 100$. Note that the average drag during the shedding phase is $C_D = 1.35$, and the Strouhal number is $St = 0.168$.*

to Koumoutsakos and Leonard's [27] equivalent Figure 21 and shows fine vortical structure in the downstream boundary layer. The computational grid (not shown) exhibits the same fine adaptivity of the other cases, with few points in the interior of the cylinder.

In Figure 5.11 we compare our drag curve with the short-time asymptotic result of Bar-Lev and Yang [5] and the vortex method result of Koumoutsakos and Leonard [27]. The agreement is excellent, even during the early $t^{-1/2}$ drag singularity. The close match between the curves is especially interesting since the methods are entirely different: Koumoutsakos and Leonard use a vortex method to resolve the vorticity equation, while we use an adaptive high-order finite difference technique to resolve the Navier–Stokes equations in velocity-pressure form. In their paper, Koumoutsakos and Leonard show that previous efforts at simulating this flow were unable to capture the short-time singularity, or the plateau and peak shape between $t = 1$ and $t = 2$.

It is interesting to compare the number of computational elements used in the wavelet and vortex methods. In our case these elements are wavelets, and in Koumoutsakos and Leonard's case they are vortices. At $t = 3.0$ we use 4.3×10^4 wavelets
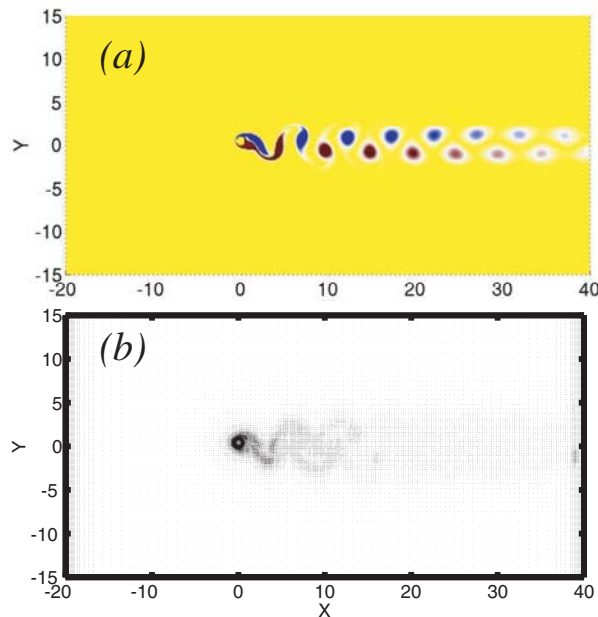
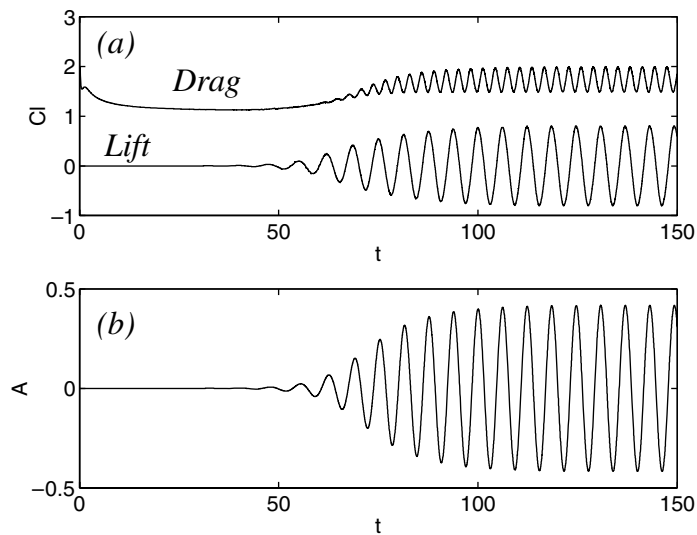FIG. 5.8. *Two-dimensional moving cylinder at* $Re = 100$, $t = 131$. (a) *Vorticity.* (b) *Adapted grid.*

FIG. 5.9. (a) *Lift and drag for a moving cylinder at* $Re = 100$. *Note that the average drag during the shedding phase is* $C_D = 1.74$, *lift amplitude is* $C_L = 0.81$, *and the Strouhal number is* $St = 0.189$. (b) *Cylinder displacement as a function of time (amplitude* $A = 0.42$).

(corresponding to a compression ratio of about 880 times), compared with 3.8×10^5 vortices used in the vortex method (i.e., almost nine times fewer computational elements). It therefore appears that the wavelet method is significantly more efficient (in terms of the number of computational elements) than the vortex method. We must remember, however, that the computational cost could be higher per element for wavelets than for vortices.
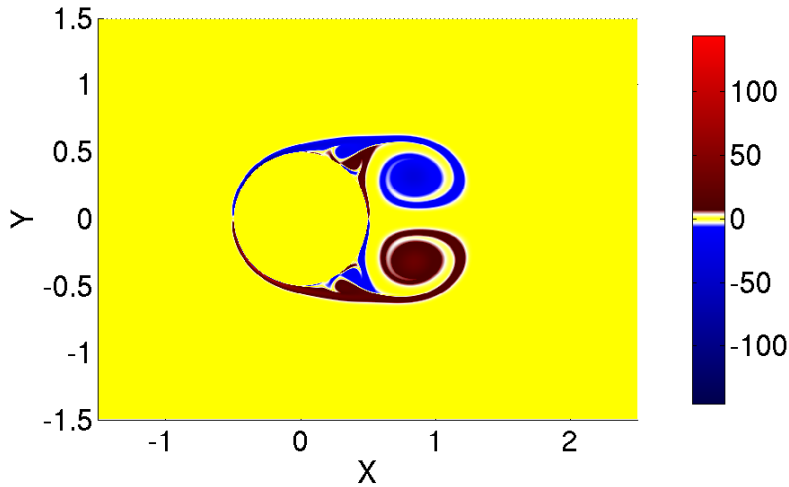
Fig. 5.10. *Vorticity field around a two-dimensional fixed cylinder at $Re = 3000$, $t = 3.0$.*
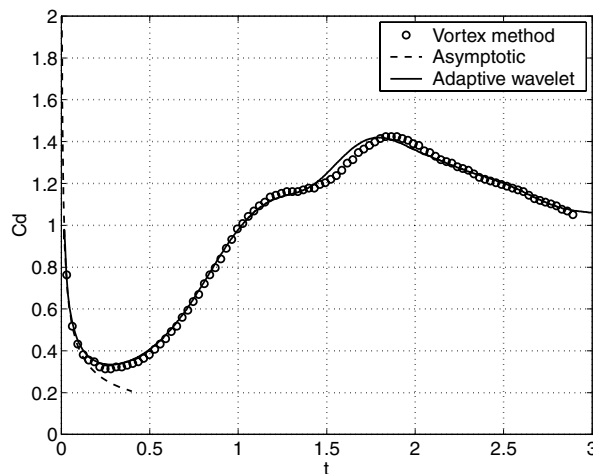

Fig. 5.11. *Drag curve for the impulsively started cylinder at $Re = 3000$ compared to the short-time asymptotic result of Bar-Lev and Yang [5] and the vortex method results of Koumoutsakos and Leonard [27].*

As mentioned in the introduction, we have found that the vorticity version of Brinkman penalization gives inaccurate results for high Reynolds numbers, especially at short times. This result has been confirmed for $Re = 3000$, where the error of the vorticity version at $t = 0.1$ is 50%, whereas the primitive variables version presented in Figure 5.11 is very accurate for short times. We have also found that the primitive variables formulation gives more accurate results for the moving cylinder (with no need to smooth the edge of the mask, as is necessary in the vorticity version). The only drawback of the primitive variables formulation is that more grid points are necessary: typically 20%–50% more. This increase is actually quite modest considering that vorticity is much more compactly distributed than velocity. These results confirm the importance of using the primitive variables formulation for high Reynolds number work.
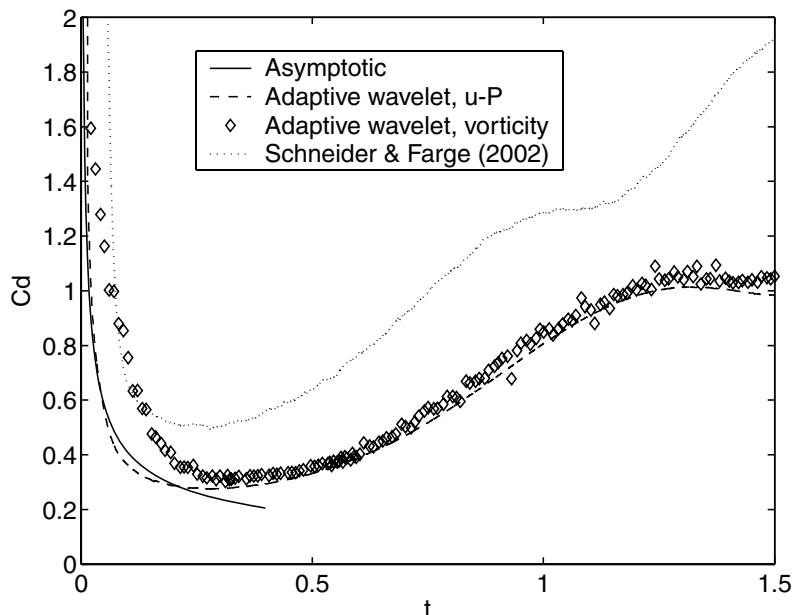
FIG. 5.12. *Drag curve for the impulsively started cylinder at Re = 3000 comparing adaptive wavelet methods using the vorticity and velocity formulations (on the smaller domain [32]) with the short-time asymptotic result [5].*

Finally, we compare our results with those of Schneider and Farge [32]. They use a Petrov–Galerkin scheme for spatial discretization along with vaguelette type wavelets [17] to solve the penalized vorticity equations (i.e., the curl of (3.4)). We are interested in two aspects: the relative efficiency and accuracy of the present primitive variables formulation compared with the vorticity formulation. We have done two additional simulations using Schneider and Farge's smaller periodic domain ($[-2, 2] \times [-2, 2]$) and coarser grid of 512^2 (i.e., $\lambda/2.4$). Both simulations use AWCM, but one uses the present primitive variables formulation, and the other uses a vorticity formulation [40]. These results are shown in Figure 5.12.

At $t = 3$ our primitive variables method uses 2.5×10^4 wavelets compared to 2.1×10^4 wavelets for the vorticity-based vaguelette code (i.e., about 20% more). Note that because we resolve a smoother field, we are able to use a larger $\epsilon = 10^{-4}$, compared with Schneider and Farge's value of $\epsilon = 10^{-5}$. This result suggests that the primitive variables version of the adaptive wavelet collocation code does not require many more wavelets than the vorticity version, even though velocity is a much smoother quantity than vorticity. The primitive variables version also produces more accurate results, as we see below.

Because the domain is rather small, we expect that the drag should be accurate for short times only. While the primitive variables formulation agrees well with the short-time asymptotic curve of Bar-Lev and Yang [5], both vorticity simulations give incorrect results (significantly overpredicting the drag). Note that differences between our results and those of Schneider and Farge [32] at later times may be attributed to the fact that they use a Petrov–Galerkin method with a fixed time step, and we use a collocation method with a variable time-step Krylov method in time. These results confirm our previous observations (referred to in section 1) that the vorticity form of the penalization method is inaccurate for $Re > 10^3$.

6. Conclusions. In this paper we have described a new method for calculating fluid-structure interaction at high Reynolds numbers. The method combines Brinkman penalization (to define solid structures) with an AWCM (to dynamically adapt the computational grid) to solve the Navier–Stokes equations. Note that although the adapted grid is nonuniform, it is a subset of a structured (Cartesian) grid. Our method also employs a new adaptive wavelet collocation multilevel elliptic solver [41] to solve the Poisson equation for pressure at each time step. After describing each aspect of the method (in sections 2, 3, and 4), we checked the accuracy and efficiency of the method in calculating the two-dimensional flow around cylinders in section 5. A summary of the main conclusions follows.

A series of calculations for $3 \times 10^1 \leq Re \leq 10^5$ demonstrated that the number of active grid points $\mathcal{N}$ scales like $Re^{1/2} \sim \lambda$ (where λ is the Taylor scale, or boundary layer thickness). The overall computational complexity ($\mathcal{N}/\Delta t$) scales like Re. This represents a significant improvement over the classical complexity estimate for two-dimensional turbulence, $\mathcal{N}/\Delta t = Re^{9/4}$. The code achieves typical compression ratios of between 270 and 1000 times compared with a uniform grid of the same resolution. Since our method is about four times slower per grid point than a similar pseudo-spectral method, this implies that our method is between 68 and 250 times faster than the equivalent pseudospectral simulation. Note that these scaling results are simply observations: we do not yet have a mathematical proof that the actual number of degrees of freedom scales like $Re^{1/2}$. It would be fascinating if this scaling could be proved mathematically.

The drag, lift, and Strouhal frequency for an isolated fixed cylinder at $Re = 10^2$ were found to be within a few percent of experimental and numerical vortex method results. We also did a calculation where the coupling between cylinder motion and fluid flow is modelled as a forced harmonic oscillator. This case demonstrated the flexibility of our method in allowing the obstacle to move relative to the grid, while the grid automatically adapts to follow the movement. Our results for oscillation amplitude, lift amplitude, and Strouhal frequency are close to those of the vortex method in Shiels, Leonard, and Roshko [34], but their drag coefficient is much higher. The moving cylinder was not significantly more expensive to calculate than the fixed cylinder (a slightly smaller time step was required due to the motion of the cylinder).

Finally, we compared drag curve calculated for impulsively started flow around a fixed cylinder at $Re = 3000$ with results from Koumoutsakos and Leonard [27]'s vortex method and Bar-Lev and Yang's [5] short-time asymptotic solution. Our method's drag curve follows the $t^{-1/2}$ asymptotic singularity at short times and agrees closely with the vortex method result at longer times. We also found that the adaptive wavelet method uses about nine times fewer computational elements than the vortex method. Comparison with the results of Schneider and Farge [32] confirms that the vorticity form of the Brinkman penalization gives inaccurate drag results for moderate Reynolds numbers ($Re > 1000$).

The present method is extremely flexible, and can be adapted to a wide variety of small and large Reynolds number flows with or without moving obstacles. Since the method is based on collocation and penalization, it is straightforward to extend from two to three dimensions. We have recently extended our method to the calculation of three-dimensional fluid-structure interaction. The method works well on a work-station for maximum resolutions of up to about 256^3, and we are currently working on parallelization and an improved data structure to allow much larger problems to be tackled efficiently. We will use the three-dimensional code to check the performance of our method for classical homogeneous isotropic turbulence, where compression relies

while $\|\mathbf{f}^J - \mathbf{L}\mathbf{u}_{\geq}^J\|_\infty > \delta_\epsilon$
    $\mathbf{r}^J = \mathbf{f}^J - \mathbf{L}\mathbf{u}_{\geq}^J$
    for all levels $j = J \; : \; -1 \; : \; j_{\min} + 1$
        do ν_1 steps of **approximate** solver for $\mathbf{L}\mathbf{v}^j = \mathbf{r}^j$
            $\mathbf{r}^{j-1} = I_w^{j-1} \left(\mathbf{r}^j - \mathbf{L}\mathbf{v}^j \right)$
        enddo
    end
    Solve for $j = j_{\min}$ level: $\mathbf{L}\mathbf{v}^j = \mathbf{r}^j$
    for all levels $j = j_{\min} + 1 \; : \; +1 \; : \; J$
        $\mathbf{v}^j = \mathbf{v}^j + I_w^j \mathbf{v}^{j-1}$
        do ν_2 steps of **approximate** solver for $\mathbf{L}\mathbf{v}^j = \mathbf{r}^j$ **enddo**
    end
    $\mathbf{u}_{\geq}^J = \mathbf{u}_{\geq}^J + \mathbf{v}^J$
    do ν_3 steps of **exact** solver for $\mathbf{L}\mathbf{u}^J = \mathbf{f}^J$ **enddo**
end

FIG. A.1. *Pseudocode for the multilevel wavelet collocation elliptic solver.*

entirely on turbulence's natural intermittency, rather than on the compact vorticity distributions associated with fluid-structure interaction. Compressible flows are another natural extension of the present method, and this is an area we are currently investigating.

The CVS (adaptive wavelet method) presented here permits efficient and accurate direct numerical simulation of flows in complex geometries for large Reynolds numbers.

Appendix. Multilevel wavelet collocation elliptic solver.

This appendix is a brief summary of the multilevel wavelet collocation algorithm for the solution of linear elliptic problems $\mathbf{L}\mathbf{u} = \mathbf{f}$. Details of the algorithm and convergence studies will appear in [41]. In the present paper the algorithm is used to solve the Poisson equation for pressure (4.6), which ensures that the velocity is incompressible.

The multilevel structure of the wavelet approximation provides a natural framework to establish multilevel V-cycle iterations on an adaptive computational grid $\mathcal{G}_{\geq}$. We recall that the adaptive computational grid $\mathcal{G}_{\geq} = \mathcal{G}_{\geq}^J$ is constructed as a set of nested adaptive computational grids $\mathcal{G}_{\geq}^j \subset \mathcal{G}_{\geq}$, such that $\mathcal{G}_{\geq}^j \subset \mathcal{G}_{\geq}^{j+1}$ for any $j < J-1$, where J is the finest level of resolution. This nested grid structure provides a framework that allows the use of the approximation from coarser levels of resolution to improve the approximation at the finest level.

The multilevel iterative algorithm is similar in spirit to multigrid methods [8], but is very different in details of implementation. First, the structure of the nested grids is different. In particular, in contrast to multigrid methods, the lower level grid is not necessarily coarser at every region of the domain. Second, a lower-order wavelet differentiation is used for the approximate solver (smoother). Third, wavelet interpolation is used for both prolongation and restriction operators. Finally, either BI-CGSTAB [38] or GMRES [31] is used as the exact solver.

The pseudocode for the full local elliptic solver (without grid adaptation) is given in Figure A.1, where δ_ϵ is the user-defined tolerance (which depends on the wavelet tolerance ϵ), $j_{\min}$ and J are, respectively, the lowest and the highest levels of resolution, ν_1 and ν_2 are, respectively, the number of pre- and postrelaxations, ν_3 is the number of iterations of the exact solver, and I_w^{j-1} and I_w^j are, respectively, the restriction and interpolation wavelet-based operators. In the numerical results presented in this paper the same damping parameters have been used after each cycle. Finally,

the weighted Jacobi second-order iterative solver was used as an approximate (i.e., smoothing) solver. The weight for the weighted Jacobi method was set to the optimal theoretical value of 2/3. Numerical experiments confirmed this weight to be optimal for adaptive calculations as well.

It is important to note that special care should be taken in situations where the elliptic operator is constructed as a product of two operators that are approximated discretely, such as factoring the Laplacian operator as $\Delta = \text{div}(\text{grad})$ in this paper. In this case the use of symmetric stencils for derivative operators would result in either complete uncoupling of odd-even points, or, at best, a weak coupling, which makes it impossible to construct efficient iterative solvers. To make the operator better conditioned and suitable for the elliptic solver we found that the simple strategy of making one operator upwind-biased and the other downwind-biased works well.

REFERENCES

[1] P. ANGOT, C.-H. BRUNEAU, AND P. FABRIE, *A penalization method to take into account obstacles in viscous flows*, Numer. Math., 81 (1999), pp. 497–520.

[2] P. ANGOT, *Analysis of singular perturbations on the Brinkman problem for fictitious domain models of viscous flows*, Math. Methods Appl. Sci., 22 (1999), pp. 1395–1412.

[3] E. ARQUIS AND J. P. CALTAGIRONE, *Sur les conditions hydrodynamiques au voisinage d'une interface milieu fluide-milieu poreux: application à la convection naturelle*, C. R. Acad. Sci. Paris Sér. II, 299 (1984), pp. 1–4.

[4] E. BACRY, S. MALLAT, AND G. PAPANICOLAOU, *Wavelet based space-time adaptive numerical method for partial differential equations*, M2AN Math. Model. Numer. Anal., 26 (1992), pp. 793–834.

[5] M. BAR-LEV AND H. YANG, *Initial flow field over an impulsively started circular cylinder*, J. Fluid Mech., 72 (1975), pp. 625–647.

[6] S. BERTOLUZZA, *An adaptive collocation method based on interpolating wavelets*, in Multiscale Wavelet Methods for Partial Differential Equations, W. Dahmen, A. Kurdila, and P. Oswald, eds., Academic Press, San Diego, CA, 1997.

[7] G. BEYLKIN AND J. KEISER, *On the adaptive numerical solution of nonlinear partial differential equations in wavelet bases*, J. Comput. Phys., 132 (1997), pp. 233–259.

[8] J. H. BRAMBLE, *Multigrid Methods*, Pitman Research Notes in Math. Ser. 294, Longman Scientific and Technical, Harlow, UK, 1993.

[9] A. BRANDT, *Guide to multigrid development*, in Multigrid Methods, Lecture Notes in Math. 960, W. Hackbusch and U. Trottenberg, eds., Springer-Verlag, New York, 1982, pp. 220–312.

[10] A. COHEN, W. DAHMEN, AND R. DEVORE, *Adaptive wavelet methods for elliptic operator equations—convergence rates*, Math. Comp., 70 (2001), pp. 27–75.

[11] A. COHEN, W. DAHMEN, AND R. DEVORE, *Sparse evaluation of compositions of functions using multiscale expansions*, SIAM J. Math. Anal., 35 (2003), pp. 279–303.

[12] A. COHEN, I. DAUBECHIES, AND J. FEAUVEAU, *Bi-orthogonal bases of compactly supported wavelets*, Comm. Pure Appl. Math., 45 (1992), pp. 485–560.

[13] W. DAHMEN, R. SCHNEIDER, AND Y. XU, *Nonlinear functionals of wavelet expansions—adaptive reconstruction and fast evaluation*, Numer. Math., 86 (2000), pp. 49–101.

[14] I. DAUBECHIES, *Orthonormal bases of compactly supported wavelets*, Comm. Pure Appl. Math., 41 (1988), pp. 909–996.

[15] D. L. DONOHO, *Unconditional bases are optimal bases for data compression and statistical estimation*, Appl. Comp. Harm. Anal., 1 (1993), pp. 100–115.

[16] M. FARGE, K. SCHNEIDER, AND N. K.-R. KEVLAHAN, *Non-Gaussianity and coherent vortex simulation for two-dimensional turbulence using an adaptive orthogonal wavelet basis*, Phys. Fluids, 11 (1999), pp. 2187–2201.

[17] J. FRÖHLICH AND K. SCHNEIDER, *Numerical simulation of decaying turbulence in an adaptive wavelet basis*, Appl. Comput. Harmon. Anal., 3 (1996), pp. 393–397.

[18] D. A. GOLDSTEIN AND O. V. VASILYEV, *Stochastic coherent adaptive large eddy simulation method*, Phys. Fluids, 16 (2003), pp. 2497–2513.

[19] M. GRIEBEL AND K. KOSTER, *Adaptive wavelet solvers for the unsteady incompressible Navier–Stokes equations*, in Advances in Mathematical Fluid Mechanics (Paseky, 1999), J. Malek, J. Necas, and M. Rokyta, eds., Springer-Verlag, Berlin, 2000, pp. 67–118.

[20] M. GRIEBEL AND K. KOSTER, *Multiscale methods for the simulation of turbulent flows*, in Numerical Flow Simulation III, Notes on Numerical Fluid Mechanics and Multidisciplinary Design 82, E. Hirschel, ed., Springer-Verlag, Berlin, 2003, pp. 203–214.

[21] J. L. GUERMOND AND J. SHEN, *Velocity-correction projection methods for incompressible flows*, SIAM J. Numer. Anal., 41 (2003), pp. 112–134.

[22] M. HOLMSTROM AND J. WALDEN, *Adaptive wavelet methods for hyperbolic PDEs*, J. Sci. Comput., 13 (1998), pp. 19–49.

[23] N. K.-R. KEVLAHAN, O. V. VASILYEV, AND A. CHERHABILI, *An adaptive wavelet method for turbulence in complex geometries*, in Proceedings of the 16th IMACS World Congress, Lausanne, Switzerland, 2000, Vol. 411-39, M. Deville and R. Owens, eds., IMACS, 2000.

[24] N. K.-R. KEVLAHAN AND O. V. VASILYEV, *An adaptive wavelet method for fluid-structure interaction*, in Direct and Large-Eddy Simulation Workshop 4, University of Twente, B. J. Geurts, R. Friedrich, and O. Métais, eds., Kluwer Academic Publishers, Norwell, MA, 2001, pp. 253–260.

[25] N. KEVLAHAN AND J.-M. GHIDAGLIA, *Computation of turbulent flow past an array of cylinders using a spectral method with Brinkman penalization*, Eur. J. Mech. B Fluids, 20 (2001), pp. 333–350.

[26] K. KHADRA, P. ANGOT, S. PARNEIX, AND J. P. CALTAGIRONE, *Fictitious domain approach for numerical modelling of Navier–Stokes equations*, Internat. J. Numer. Meth. Fluids, 34 (2000), pp. 651–684.

[27] P. KOUMOUTSAKOS AND A. LEONARD, *High-resolution simulations of the flow around an impulsively started cylinder using vortex methods*, J. Fluid Mech., 296 (1995), pp. 1–38.

[28] J. LIANDRAT AND P. TCHAMITCHIAN, *Resolution of the 1D Regularized Burgers Equation Using a Spatial Wavelet Approximation*, Tech. report, NASA Contractor Report 187480, ICASE Report 90-83, NASA Langley Research Center, Hampton, VA, 1990.

[29] H. PERSILLON AND M. BRAZA, *Physical analysis of the transition to turbulence in the wake of a circular cylinder by three-dimensional Navier–Stokes simulation*, J. Fluid Mech., 365 (1998), pp. 23–88.

[30] C. S. PESKIN, *Flow patterns around heart valves: A numerical method*, J. Comput. Phys., 10 (1972), pp. 252–271.

[31] Y. SAAD AND M. H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

[32] K. SCHNEIDER AND M. FARGE, *Adaptive wavelet simulation of a flow around an impulsively started cylinder using penalisation*, Appl. Comput. Harmon. Anal., 12 (2002), pp. 374–380.

[33] K. SCHNEIDER, N. K.-R. KEVLAHAN, AND M. FARGE, *Comparison of an adaptive wavelet method and nonlinearly filtered pseudo-spectral methods for two-dimensional turbulence*, Theoret. Comput. Fluid Dynamics, 9 (1997), pp. 191–206.

[34] D. SHIELS, A. LEONARD, AND A. ROSHKO, *Flow-induced vibration of a circular cylinder at limiting structural parameters*, J. Fluids Structures, 15 (2001), pp. 3–21.

[35] F. SPRENGEL, *Some Remarks on Multilevel Algorithms for Finite Difference Discretizations on Sparse Grids*, Tech. report MAS-R9924, Centrum voor Wiskunde en Informatica, Amsterdam, 1999.

[36] W. SWELDENS, *The lifting scheme: A custom-design construction of biorthogonal wavelets*, Appl. Comput. Harmon. Anal., 3 (1996), pp. 186–200.

[37] W. SWELDENS, *The lifting scheme: A construction of second generation wavelets*, SIAM J. Math. Anal., 29 (1998), pp. 511–546.

[38] H. A. VAN DER VORST, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 631–644.

[39] O. V. VASILYEV AND C. BOWMAN, *Second generation wavelet collocation method for the solution of partial differential equations*, J. Comput. Phys., 165 (2000), pp. 660–693.

[40] O. V. VASILYEV AND N. K.-R. KEVLAHAN, *Hybrid wavelet collocation—Brinkman penalization method for complex geometry flows*, Internat. J. Numer. Methods Fluids, 40 (2002), pp. 531–538.

[41] O. V. VASILYEV AND N. K.-R. KEVLAHAN, *An adaptive multilevel wavelet collocation method for elliptic problems*, J. Comp. Phys., to appear.

[42] O. V. VASILYEV AND S. PAOLUCCI, *A dynamically adaptive multilevel wavelet collocation method for solving partial differential equations in a finite domain*, J. Comput. Phys., 125 (1996), pp. 498–512.

[43] O. V. VASILYEV AND S. PAOLUCCI, *A fast adaptive wavelet collocation algorithm for multidimensional PDEs*, J. Comput. Phys., 138 (1997), pp. 16–56.

[44] O. V. VASILYEV, *Solving multi-dimensional evolution problems with localized structures using second generation wavelets*, Int. J. Comput. Fluid Dyn., 17 (2003), pp. 151–168.

[45] C. WILLIAMSON, *Defining a universal and continuous Strouhal-Reynolds number relationship for the laminar vortex shedding of a circular cylinder*, Phys. Fluids, 31 (1988), pp. 2742–2744.