An adaptive wavelet method for turbulence in complex geometries

N. K.-R. Kevlahan *   O. Vasilyev †   A. Cherhabili*

Abstract

This paper introduces the combination of an adaptive collocation wavelet method and Brinkman penalization as a way of solving the one- and two-dimensional Navier–Stokes equations in arbitrarily complex geometries. The main advantages of the wavelet and penalization techniques are described, and a brief summary of their implementation is given. The combined method is then applied to a simple one-dimensional test case (Stoke's flow) using the Krylov method in time, and the results are summarized. It is expected that results from a similar code for two-dimensional flow will be presented during the conference.

Key words: turbulence, wavelets, penalization, bluff-body.

AMS subject classifications: 65C20, 76G25.

1 Introduction

Adaptive wavelet methods have recently been developed to solve the Navier-Stokes equations at high Reynolds numbers *e.g.* [5], [7]. The adaptive wavelet method is appropriate for high Reynolds number 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 with little loss in accuracy. 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 collocation points actually used.

Parallel to the development of efficient wavelet codes for turbulence, we have been investigating the use of the Brinkman equation to simulate the presence of arbitrarily complex solid boundaries (which may be moving in time) [3]. This technique 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 estimated rigorously in terms of the penalization parameter. It can also be shown that the solution of the penalized equations converges to the exact solution in the limit as the penalization parameter tends to zero [1].

In this paper we describe the combination of the two approaches: a collocation adaptive wavelet method to solve the Navier–Stokes equations with a Brinkman penalization to impose solid boundaries. This technique is applied first to Stokes flow (to test convergence and accuracy) and will later be applied to two- and three-dimensional turbulent flows with solid obstacles. The combination of these two methods should allow for the first time direct simulations of turbulent flows in geometries of engineering interest. Fluid-structure interactions (where the obstacles move in response to the fluid forces) are a natural application of this technique, and they are currently under investigation.

---

*Department of Mathematics and Statistics, McMaster University

†Department of Mechanical Engineering, University of Missouri at Columbia

2 Penalization method

2.1 General theory

Let us consider a viscous incompressible fluid governed by the Navier–Stokes equations

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \nabla P = \nu \Delta \boldsymbol{u}, \tag{1}$$

$$\nabla \cdot \boldsymbol{u} = 0. \tag{2}$$

For simplicity we focus here on the case where the fluid occupies the complement in the plane R^2 of a periodic lattice of obstacles O_i. The boundary conditions associated with this problem are therefore:

$$\boldsymbol{u} \text{ is } Q\text{-periodic}, \ Q =]0, L_1[\times]0, L_2[, \tag{3}$$

$$\boldsymbol{u} = 0 \text{ on } \partial O_i, \ \forall i \tag{4}$$

To model the effect of the no-slip boundary conditions on the obstacles O_i without explicitly imposing (4) we replace (1) – (4) by the following set of *penalized* equations

$$\frac{\partial \boldsymbol{u}_\eta}{\partial t} + \boldsymbol{u}_\eta \cdot \nabla \boldsymbol{u}_\eta + \nabla P_\eta = \nu \Delta \boldsymbol{u}_\eta - \frac{1}{\eta} \chi_0 \boldsymbol{u}_\eta, \tag{5}$$

$$\nabla \cdot \boldsymbol{u}_\eta = 0, \tag{6}$$

$$\boldsymbol{u}_\eta \text{ is } Q\text{-periodic}, \tag{7}$$

Now (5) and (6) are to be satisfied in the *whole plane* R^2. Here $\eta > 0$ is a penalization coefficient and χ_0 denotes the characteristic (or mask) function

$$\chi_0(\boldsymbol{x}, t) = \begin{cases} 1 & \text{if } \boldsymbol{x} \in O_i, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

As $\eta \to 0$, it was proved theoretically [1] that the solutions of the penalized equations (5) to (7) converge to that of the Navier–Stokes equations with the correct boundary conditions (1) – (4). More precisely, the upper bound on the global error of the penalization was shown to be [1]

$$||\boldsymbol{u} - \boldsymbol{u}_\eta|| \leq C \eta^{1/4}. \tag{9}$$

This penalization has been implemented in a finite difference code [4] for flow around a cylinder and was found to give very good results. In fact, the actual error was 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.

Another advantage of the Brinkman penalization is that the force $\boldsymbol{F}_i$ acting on an obstacle O_i can be found by simply integrating the penalization term over the volume of the obstacle: $\boldsymbol{F}_i = 1/\eta \int_{O_i} \boldsymbol{u} \, d\boldsymbol{x}$. Thus, the calculation of lift and drag on an obstacle can be made simply, accurately and at very low cost.

2.2 Application to Stoke's flow

To help understand what the penalized approximation to the solution of the Navier–Stokes equations looks like near a solid boundary we consider the simple case of Stoke's flow. Stoke's solution describes a uni-directional flow accelerated impulsively from rest and bounded by an infinite flat plate parallel to the flow. The velocity of the flow must be zero at the surface of the plate, and thus the flow develops a typical 'boundary layer' shear profile that progressively diffuses vorticity into the interior of the flow. In this case the Navier–Stokes equations are

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2}, \tag{10}$$

where $u = u_2(x, t)$, $u(x, 0) = 1$, $u(0, t) = 0$ and $x \geq 0$ (note that we consider the lower half-plane to be solid). It is assumed that the flow remains uni-directional (*i.e.* that no instabilities develop). The above equations can be easily solved, giving

$$u(x, t) = \text{erf}\left(\frac{x}{\sqrt{4\nu t}}\right), \tag{11}$$

where erf is the usual error function: $\mathrm{erf}(z) \equiv \frac{2}{\sqrt{\pi}} \int_0^z e^{-s^2}\, \mathrm{d}s$.

Now, the penalized approximation to Stokes flow is given by

$$(12) \qquad \frac{\partial u_\eta}{\partial t} = -\frac{1}{\eta} H(-x) u_\eta + \nu \frac{\partial^2 u_\eta}{\partial x^2},$$

where $u_\eta(x,0) = 1$ and $u_\eta(\cdot, t)$ is defined on the entire plane R^2. Equation (12), with $\nu = 1/2$ for simplicity, can be easily solved using Laplace transforms [3] giving

$$(13) \qquad u_\eta(x,t) = e^{-t/\eta}\left[\mathrm{erf}\left(\frac{-x}{\sqrt{2t}}\right) + \frac{1}{\pi}\int_0^1 \frac{\exp\left(ty/\eta - \frac{1}{2t}\frac{x^2}{1-y}\right)}{\sqrt{y}\sqrt{1-y}}\, \mathrm{d}y\right], \quad x < 0,$$

$$(14) \qquad u_\eta(x,t) = \mathrm{erf}\left(\frac{x}{\sqrt{2t}}\right) + \frac{1}{\pi}\int_0^1 \frac{\exp(-ty/\eta - \frac{1}{2t}\frac{x^2}{1-y})}{\sqrt{y}\sqrt{1-y}}\, \mathrm{d}y, \qquad x > 0.$$

Note that the first term on the right hand side of (14) is the exact solution $u(x,t)$, and therefore the second term is the error. It can be shown [3] that the integrated error between the exact and the approximate solutions in the fluid is of the order of $\sqrt{\eta}$ (which is consistent with 9).

Although Stoke's flow is linear and uni-directional, it generates a solution to the three-dimensional Navier–Stokes equation. One might naturally ask whether the situation described above is exceptional or gives some indication of the general solution to the Navier–Stokes equations. It can be argued that in the evanescent layer of thickness $O(\sqrt{\eta})$ near a smooth boundary, the flow is approximately linear (since the convective terms become negligible with respect to the diffusive and penalized terms) and parallel to the boundary. Thus the results obtained above for the error and behaviour of the penalized solution should also hold for most three-dimensional flows.

3 Wavelet 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 much more simply than with wavelet-Galerkin methods. 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 where sharp gradients occur.

In this section we briefly review the adaptive collocation wavelet method used (for more details please see [8]). In particular, we will sketch efficient wavelet-based procedures for dynamic grid adaptation and calculation of spatial derivatives. Note that we use Swelden's [6] lifted wavelets, which can be calculated efficiently with the desired degree of smoothness.

3.1 Grid Adaptation

Grid adaptation occurs naturally in wavelet methods. To illustrate the algorithm, let us consider a function $f(x)$, defined on a closed interval Ω. We use interpolating wavelets constructed on a set of grids,

$$(15) \qquad \mathcal{G}^j = \left\{x_k^j \in \Omega : k \in \mathcal{K}^j\right\}, \quad j \in \mathcal{Z},$$

where grid points x_k^j can be uniformly or nonuniformly spaced. The only restriction is that $x_k^j = x_{2k}^{j+1}$, which guarantees the nestedness of the grids, *i.e.* $\mathcal{G}^j \subset \mathcal{G}^{j+1}$. We construct scaling functions $\phi_k^j(x)$ ($k \in \mathcal{K}^j$) and wavelets $\psi_l^j(x)$ ($l \in \mathcal{L}^j$) such that on each level of resolution J the function $f(x)$ can be approximated as

$$(16) \qquad f^J(x) = \sum_{k \in \mathcal{K}^0} c_k^0 \phi_k^0(x) + \sum_{j=0}^{J-1} \sum_{l \in \mathcal{L}^j} d_l^j \psi_l^j(x).$$

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_l^j will be small unless the function varies on the scale of j at the location x_l^k. In fact, the error incurred by ignoring coefficients with magnitude lower than ϵ is $O(\epsilon)$.

In order to realize the benefits of the wavelet compression, we need to be able to reconstruct the $f_{\geq}^J(x)$ from the subset of $\mathcal{N}$ grid points. Note that every scaling function $\phi_k^j(x)$ is uniquely associated with x_k^j, while each wavelet $\psi_l^j(x)$ is uniquely associated with the collocation point x_{2l+1}^j. So once the wavelet decomposition is performed, each grid point on the finest level of resolution J is uniquely associated either with the wavelet or the scaling function at the coarsest level of resolution. Consequently, the collocation point should be omitted from the computational grid if the associated wavelet is omitted from the approximation. Note that for the stability of reconstruction algorithm we need to keep all grid points associated with the scaling function at the coarsest level of resolution. 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 $j < J - 1$.

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 computation grid should include points associated with wavelets belonging to an *adjacent zone* of wavelets for which the magnitude of their coefficients is greater then an *a priori* prescribed threshold. We found that the optimal adjacent zone includes the nearest points at the same, one higher, and one lower level of resolution.

3.2 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, which takes advantage of the multiresolution wavelet decomposition, fast wavelet transform, and uses finite difference differentiation. In other words we make wavelets do what they do the best: compress and interpolate. Finite differences differentiate the resulting polynomials. We note that the differentiation procedure introduced in this section is similar in spirit to the procedure used in wavelet-Galerkin method by Walden [9].

The differentiation procedure is based on the interpolating properties of second generation wavelets. We recall that wavelet coefficients d_k^j measure the difference between the approximation of the function at $j + 1$ 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 x_k^j, *i.e.* $|d_{k+l}^j| < \epsilon$ ($l = -1, 0$) and points $x_{2k\pm1}^{j+1}$ are not present in $\mathcal{G}_{\geq}^{j+1}$, then there exist some neighbourhood of x_k^j, where the actual function is well approximated by local polynomial based on c_l^j ($l \in \mathcal{K}^j$).

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 each 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.

3.3 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 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 $d_k^j = 0$ for the new gridpoints.

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 error of approximation. 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. In typical

applications the value of ϵ varies between 10^{-2} and 10^{-5}, assuming that the unknown dependent variables have been properly normalized. As the value of ϵ increases, fewer grid points are used in the solution.

4 Results

We have solved the penalized equations for Stokes flow using the adaptive collocation method described in §3. This problem is the first step in applying the penalization method to more complicated geometries in two and three dimensions, and shows us that the wavelet method is able to resolve the strong discontinuity in the equations (a jump of 10^6) at the solid boundary. Note that this problem is a good test since the flow near the boundary in higher dimensions and more complicated geometries should look locally like the Stoke's flow solution. The small factor ϵ in the penalized equations means that the time evolution is *stiff* and thus we need to use a stiffly stable method to advance in time. We chose the Krylov method [2] which is stiffly stable and allows the order of accuracy of the method to be adjusted easily.

The following parameters were used in the simulation: $\nu = 0.5$, $\eta = 10^{-6}$, $\epsilon = 10^{-5}$, fourth-order lifted wavelets and a maximum of 13 levels in scale were allowed. The results of the simulation at $t = 1$ are shown in figure 1 along with the adapted grid (in space and scale) and the error compared with the exact penalized solution and with the exact physical solution (Stoke's flow). Figure 1(a) shows that the boundary layer profile has been cleanly resolved with no oscillations or overshoot. Figure 1(b) illustrates the automatic grid refinement which characterizes the wavelet method. The strength of wavelet coefficients is proportional to the gradient of the solution and weaker coefficients (below the ϵ threshold) are eliminated which produces only an $O(\epsilon)$ error. This procedure automatically generates the refined grid shown in the figure, with no need for re-meshing. There are only 125 active nodes out of a potential maximum (with 13 levels) of 32 768. The error compared with the exact penalized solution is shown in figure 1(c). Note that the error is consistent with the wavelet threshold parameter error=$O(\epsilon) = 10^{-5}$, except for a slight overshoot at the boundary. Finally, in figure 1(d) we show the error compared with the physical (non-penalized) solution, which is consistent with the estimated error $= O(\sqrt{\eta}) = 10^{-3}$.

To summarize, the adaptive collocation wavelet method was successfully applied to solve the penalized Stoke's flow problem. There was no sign of oscillation or overshoot, and the errors were consistent with the theoretical estimates. Furthermore, an accuracy of 10^{-5} was attained with only 125 active modes out of a total of 32 768 possible modes (only 0.38%). Since penalized Stoke's flow should be typical of boundary layer flow in more general situations, these results are encouraging for high Reynolds number calculations in two and three dimensions.

5 Future work

In this paper we have presented the results of combining an adaptive wavelet method with penalization for solid boundaries to solve the Stoke's flow problem for a one-dimensional boundary-layer. A stiffly stable Krylov subspace method was used for time advancement. The results are encouraging, and we are currently working on the two-dimensional Navier–Stokes equation version. We are confident that the two-dimensional problem will not pose any serious new problems and should allow us to evaluate the advantages/disadvantages of our approach for calculating turbulent flow in complex geometries.

The potential advantages of the combination of adaptive wavelet collocation in space with penalization for solid boundaries and the Krylov method in time are the following. First, the wavelet approach allows for automatic grid refinement with a well-controlled error. (The grid automatically follows strong gradients in the solution.) Secondly, the Brinkman penalization allows solid obstacles of arbitrary complexity to be placed in the flow without modifying the collocation grid (although, of course, the grid is automatically refined near the boundaries). Furthermore, the error in the penalization is known in terms of the penalization parameter η. The penalization approach also allows for boundaries which move (*e.g.* in response to fluid forces) or change shape (*e.g.* a beating heart). Finally, the Krylov method in time is stiffly stable and uses an adaptive step-size. The order of accuracy of the method can be optimized to suit the problem being calculated. Thus, we hope that the combination of an adaptive wavelet method with penalization will allow the calculation of turbulent flows in complex geometries at Reynolds numbers typical of engineering applications.
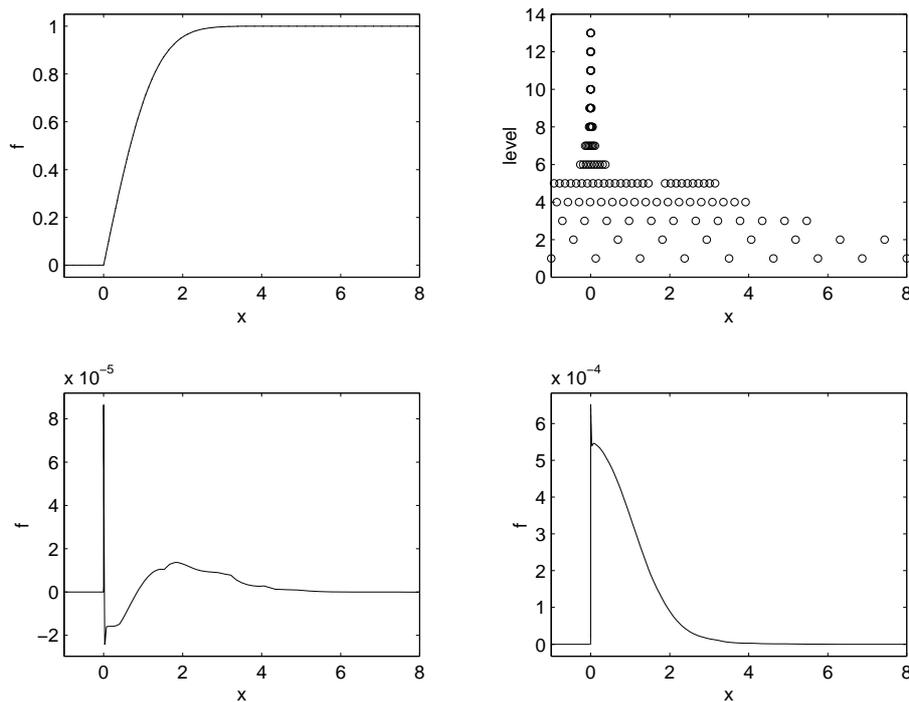
Figure 1: Results of penalized wavelet calculation of Stokes flow. (a) Velocity profile. (b) Active grid points as a function of scale and position. (c) Error compared with exact penalized solution. (d) Error compared with exact Stokes solution (unpenalized).

References

[1] P. Angot, C.-H. Bruneau, & P. Fabrie *A penalization method to take into account obstacles in viscous flows*, Numerische Mathematik 81 (1999), pp. 497–520.

[2] W. S. Edwards, L. S. Tuckerman, R. A. Friesner, & D. C. Sorensen, *Krylov methods for the incompressible Navier–Stokes equations*, J. Comp. Phys. 110, (1994) 82–102.

[3] N. K.-R. Kevlahan & J.-M. Ghidaglia *Computation of turbulent flow past an array of cylinders using a spectral method with Brinkman penalization*, preprint.

[4] K. Khadra, S. Parneix, P. Angot, & J.-P. Caltagirone *Fictitious domain approach for numerical modelling of Navier–Stokes equations* Int. J. Num. Methods in Fluids (accepted) (1999).

[5] K. Schneider, N. K.-R. Kevlahan & M. Farge, *Comparison of anadaptive wavelet method and nonlinearly filtered pseudo-spectral methods for two-dimensional turbulence*, Theoret. Comput. Fluid Dynamics, 9 (1997), pp. 191–206.

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

[7] O. Vasilyev & S. Paolucci, *A fast adaptive wavelet collocation algorithm for multidimensional PDEs*, J. Comput. Phys., 138 (1997), pp. 16–56.

[8] O. Vasilyev & C. Bowman, *Second Generation Wavelet Collocation Method for the Solution of Partial Differential Equations*, Submitted to J. Comput. Phys. (2000).

[9] J. Walden, *Filter Bank Methods for Hyperbolic PDEs*, SIAM J. Numer. Anal. 36, (1999) pp. 1183-1233 .