JOURNAL OF COMPUTATIONAL PHYSICS 122, 191–217 (1995)

Unstructured Spectral Element Methods for Simulation of Turbulent Flows

RONALD D. HENDERSON

Aeronautics and Applied Mathematics, California Institute of Technology, Pasadena, California 91125

AND

GEORGE EM KARNIADAKIS

Center for Fluid Mechanics, Division of Applied Mathematics, Brown University, Providence, Rhode Island 02912

Received December 9, 1994; revised May 12, 1995

In this paper we present a spectral element-Fourier algorithm for simulating incompressible turbulent flows in complex geometries using unstructured quadrilateral meshes. To this end, we compare two different interface formulations for extending the conforming spectral element method in order to allow for *surgical* mesh refinement and still retain spectral accuracy: the Zanolli iterative procedure and variational patching based on auxiliary "mortar" functions. We present an interpretation of the original mortar element method as a patching scheme and develop direct and iterative solution techniques that make the method efficient for simulations of turbulent flows. The properties of the new method are analyzed in detail by studying the eigenspectra of the advection and diffusion operators. We then present numerical results that illustrate the flexibility as well as the exponential convergence of the new algorithm for nonconforming discretizations. We conclude with simulation studies of the turbulent cylinder wake at Re = 1000 (external flow) and turbulent flow over riblets at Re = 3280 (internal flow). © 1995 Academic Press, Inc.

1. INTRODUCTION

High-order numerical methods, i.e., spectral and implicit finite difference schemes, have been used almost exclusively in the direct numerical simulation of turbulent flows in the last two decades [1]. They provide fast convergence, small diffusion and dispersion errors, easier implementation of the "inf-sup" condition for incompressible Navier–Stokes, better data volume over surface ratio for efficient parallel processing, and better input/output handling due to the smaller volume of data. Despite their higher computational cost (in number of operations per grid point) they are more efficient for long time integration of unsteady flow problems, as shown in the analysis of Kreiss [2]. They, perhaps, bear the greatest promise in extending the methods of flow simulation currently practiced in low Reynolds number regimes to higher and thus more realistic regimes. However, the geometric complexity in a number of important prototype turbulent flows substantially limits the Reynolds number regime that can be simulated using the resources currently available. Figure 1 shows this schematically and gives an approximate overview of simulations completed at the present time.

Multidomain, high-order methods that allow for local refinement and preserve the fast convergence of spectral discretizations could also handle complex geometries and offer many advantages over global spectral methods [3]. The spectral element method was developed with such an objective in mind [4, 5]. Since then several different versions have been formulated, including discretizations on triangular and nonconforming quadrilateral subdomains [6–8]. This later work is conceptually similar to the idea of "constrained approximation" developed by Demkowicz *et al.* as part of an *h-p* adaptive strategy for finite element methods [9]. They use continuity arguments to explicitly construct a constrained approximation space. In constrast, the methods we consider here generate these constraints implicitly from the underlying differential equation or from an adjoint variational equation.

Figure 2 illustrates the difference between a mesh of conforming elements and a mesh of nonconforming elements, which we will refer to as an "unstructured" mesh. Generally such a discretization will be the result of regular *h*-refinements of an initially conforming mesh, but the refinement is arbitrary and completely local. The only restriction is that elements of the mesh do not overlap.

Features such as selective local refinement and automatic mesh adaptation are important in order to increase the computational efficiency of direct (DNS) and large eddy simulations (LES) of turbulent flows. These features, which can only be implemented with great difficulty for conforming discretizations, provide the foundation for a new and potentially very effective approach for simulating wall-bounded turbulence. In particular, recognizing the difficulty in employing subgrid mod-

0021-9991/95 $12.00
Copyright © 1995 by Academic Press, Inc.
All rights of reproduction in any form reserved.

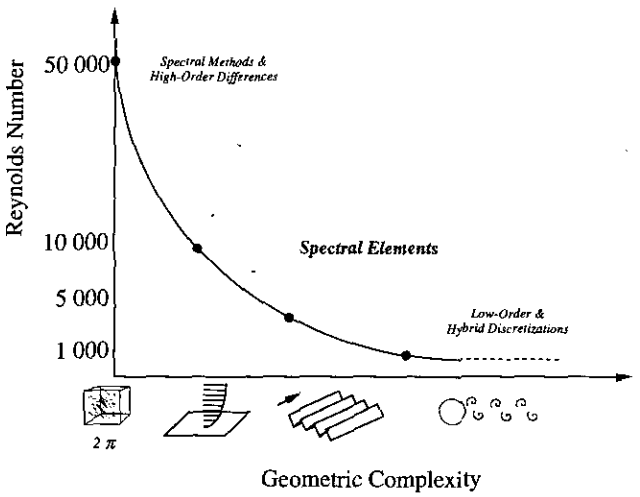

FIG. 1. Conceptual overview of turbulent flow simulations.

els in the near-wall region of a turbulent flow, all anisotropic scales near the wall are fully resolved (a DNS approach), while the more isotropic scales far from the wall are resolved using a coarser grid and subgrid models (an LES approach). Such a hybrid DNS–LES approach has already been implemented for a flow past a sphere using conforming spectral elements [10]. It is clear, however, that the success of such an approach in future and more demanding simulations will depend critically on composite meshes and nonconforming discretizations as shown in Fig. 2. In particular, the ability to easily construct a composite grid of very fine resolution with a more uniform background grid is essential for efficiently resolving wall-bounded turbulent flows.

In this work we develop nonconforming spectral element algorithms for the simulation of turbulent flows in nominally two-dimensional geometries of arbitrary geometric complexity; earlier work has addressed the conforming case [11]. The z-direction is homogeneous and thus Fourier expansions can be employed along that direction to enhance computational efficiency. We can then express the physical domain as $\Omega = \Omega_m \times [0, L_z]$, where L_z is the periodic length in the z-direction and $\Omega_m \subset \mathcal{R}^2$ is an arbitrary two-dimensional region. The full three-dimensional, time-dependent velocity field $\mathbf{u}(x, y, z, t)$ will be represented by Fourier interpolation through a set of evenly spaced planes in the periodic direction of the domain. The velocity field is thus determined in physical space by the M "samples" $\mathbf{u}_k \equiv \mathbf{u}(\cdot, k\Delta z)$, and in Fourier space by the $M + 1$ complex Fourier coefficients $\mathbf{u}_m \equiv \mathbf{u}(\cdot, i\beta_m)$, where $\beta_m = 2\pi m/L_z$ is a given spatial frequency and m a given wave number. These two sets of coefficients are related through the discrete Fourier transform

$$\mathbf{u}_k(x, y, t) = \frac{1}{M} \sum_{m=-M/2}^{M/2} \mathbf{u}_m(x, y, t) e^{-i\beta_m k\Delta z}. \tag{1}$$

This decomposition provides the ability to efficiently simulate the turbulent regime of several classical prototype flows, such as flow past a cylinder, flow over a backwards-facing step, and flow over riblets. The decomposition in modes is shown in Fig. 3; the corresponding two-dimensional, time-dependent Fourier coefficients are computed on the unstructured mesh shown in the figure, which remains the same for each mode m. An additional advantage of the unstructured spectral element-Fourier method for external flows and spatially developing flows, in general, is that large subdomains (elements) can be employed in the regions with relatively small field gradients, resulting in significant computational savings.

This paper is organized as follows. In Section 2 we present the integration scheme for the Navier–Stokes equations, the spatial discretization based on unstructured spectral elements, and an analysis of the discrete advection and diffusion operators. Section 3 covers the solution algorithms for the elliptic

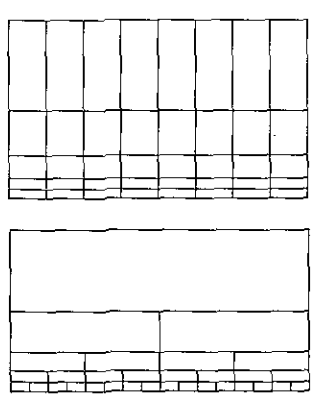

FIG. 2. Conforming (upper) and nonconforming (lower) spectral element meshes.

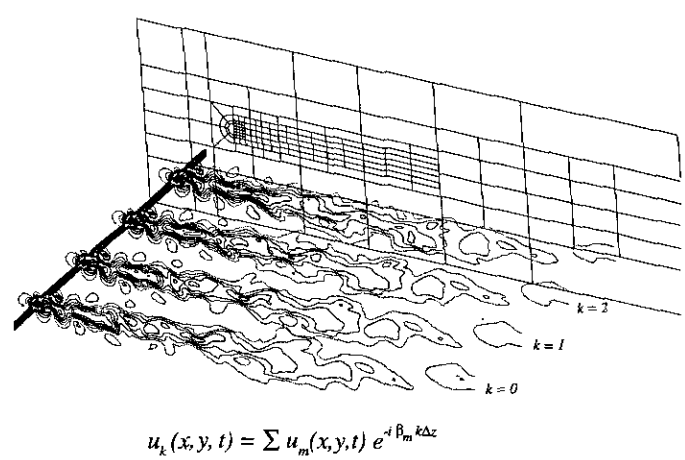

FIG. 3. Selected cross sections of a three-dimensional wake illustrating the computational approach used for flow past nominally two-dimensional geometries; k is the index of the physical plane defined in (1).

systems. Some examples that demonstrate accuracy and efficiency are given in Section 4, followed by two applications to turbulent flows and conclusions in Section 5.

2. FORMULATION

In this section we introduce the discrete form of the Navier–Stokes equations governing the flow, concentrating first on the discretization of time. Next we develop the spectral element approximations to the advection and diffusion operators on unstructured meshes and analyze the discrete eigenspectrum of these systems.

2.1. Navier–Stokes Equations

We consider here Newtonian, incompressible flows with constant properties governed by the Navier–Stokes equations in the form

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega, \tag{2a}$$

$$\frac{\partial \mathbf{u}}{\partial t} = -\nabla p + \frac{1}{\mathrm{Re}}\mathbf{L}(\mathbf{u}) + \mathbf{N}(\mathbf{u}) \quad \text{in } \Omega, \tag{2b}$$

subject to boundary conditions of the form

$$\mathbf{u} = \mathbf{u}_0 \quad \text{on } \Gamma; \tag{2c}$$

$\mathbf{u} = (u\hat{x} + v\hat{y} + w\hat{z})$ is the fluid velocity and p is the fluid pressure divided by density. The parameter $\mathrm{Re} \equiv Ud/\nu$ is the Reynolds number based on the kinematic viscosity ν of the fluid and appropriate velocity U and length scales d of the flow. The linear diffusion and nonlinear advection operators, represented above by $\mathbf{L}$ and $\mathbf{N}$, are defined as

$$\mathbf{L}(\mathbf{u}) \equiv \nabla^2 \mathbf{u}, \tag{3}$$

$$\mathbf{N}(\mathbf{u}) \equiv -\frac{1}{2}[\mathbf{u} \cdot \nabla \mathbf{u} + \nabla \cdot (\mathbf{u}\mathbf{u})], \tag{4}$$

where $\mathbf{N}$ is written in skew-symmetric form to minimize aliasing errors [12]. We wish to solve this set of equations for the velocity and pressure fields at each point $\mathbf{x} \in \Omega$, where Ω is an open set in $\mathcal{R}^3$ with piecewise smooth boundary Γ.

Let $\mathbf{u}^n \equiv \mathbf{u}(\cdot, n\Delta t)$ be the solution at time level n. Using the above representation for the velocity and pressure fields, we can write the semi-discrete system of equations satisfied by the mth Fourier mode as

$$\frac{1}{\Delta t}\left(\hat{\mathbf{u}}_m - \sum_{q=0}^{J-1} \alpha_q \mathbf{u}_m^{n-q}\right) = \sum_{q=0}^{J-1} \beta_q \mathbf{N}_m(\mathbf{u}^{n-q}), \tag{5a}$$

$$\frac{1}{\Delta t}(\hat{\hat{\mathbf{u}}}_m - \hat{\mathbf{u}}_m) = -\nabla \bar{p}_m^{n+1}, \tag{5b}$$

$$\frac{1}{\Delta t}(\gamma_0 \mathbf{u}_m^{n+1} - \hat{\hat{\mathbf{u}}}_m) = \mathbf{L}_m(\mathbf{u}_m^{n+1}). \tag{5c}$$

The constants α_q, β_q, and γ_0 are integration coefficients for a mixed explicit/implicit stiffly-stable scheme of order J and have been computed for schemes of up to third order [13]. $\hat{\mathbf{u}}_m$ and $\hat{\hat{\mathbf{u}}}_m$ are intermediate velocity fields used to split the time step into separate integration schemes for the nonlinear advection terms (explicit), incompressibility constraint (implicit), and linear diffusion (implicit). The operators $\mathbf{L}_m$ and $\mathbf{N}_m$ are defined in Fourier space as

$$\mathbf{L}_m(\mathbf{u}) \equiv \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} - \beta_m^2\right)\mathbf{u}_m, \tag{6}$$

$$\mathbf{N}_m(\mathbf{u}) \equiv \sum_{k=0}^{M-1} \mathbf{N}_k(\mathbf{u}) e^{i\beta_m k \Delta z}. \tag{7}$$

Note that the nonlinear terms are computed in *physical* space on each of the M data planes and then transformed back, unlike the linear terms in which the modes are decoupled.

In the second step (5b), $\bar{p}_m^{n+1}$ is the scalar field that ensures the final velocity field will be solenoidal and is computed by solving the Helmholtz equation

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} - \beta_m^2\right)\bar{p}_m^{n+1} = \frac{1}{\Delta t}\left(\frac{\partial \hat{u}_m}{\partial x} + \frac{\partial \hat{v}_m}{\partial y} + i\beta_m \hat{w}_m\right) \quad \text{in } \Omega_m, \tag{8}$$

subject to a Neumann-type boundary condition of the form

$$\frac{\partial \bar{p}_m^{n+1}}{\partial \mathbf{n}} = -\mathbf{n} \cdot \sum_{q=0}^{J-1} \beta_q \left[\mathbf{N}_m(\mathbf{u}^{n-q}) - \frac{1}{\mathrm{Re}} \nabla \times \boldsymbol{\xi}_m^{n-q}\right] \quad \text{on } \Gamma_m, \tag{9}$$

where $\boldsymbol{\xi} \equiv \nabla \times \mathbf{u}$ is the fluid vorticity and $\mathbf{n}$ is the unit outward normal to Γ_m. This high-order pressure boundary condition is the key to the time accuracy of the scheme, and is analyzed fully in Ref. [13].

To advance the solution to a new time level we must solve an advection equation and a sequence of elliptic Helmholtz equations for the pressure and velocity on a *two-dimensional* domain Ω_m. Our discrete approximation to these operators will be based on spectral element methods. In the next section we state the general form of the Helmholtz equation as a model problem for the elliptic kernel of the incompressible Navier–Stokes equations and analyze the numerical properties of the resulting discrete systems.

2.2. Conforming Variational Formulation

Let $\mathbf{x} \in \Omega$ be an open set in $\mathcal{R}^2$ with piecewise smooth boundary Γ. Assume that Γ can be constructed as $\Gamma = \Gamma_g \cup$

Γ_h and that Γ_h may be empty but Γ_g is not. The Helmholtz problem is: given $\lambda \in \mathcal{R}$ and smooth functions $f: \Omega \to \mathcal{R}$, $g: \Gamma_g \to \mathcal{R}$, and $h: \Gamma_h \to \mathcal{R}$, find u such that

$$\nabla^2 u - \lambda^2 u + f = 0 \quad \text{in } \Omega, \tag{10a}$$

subject to the boundary conditions

$$u = g \quad \text{on } \Gamma_g, \tag{10b}$$

$$\mathbf{n} \cdot \nabla u = h \quad \text{on } \Gamma_h. \tag{10c}$$

The starting point for our numerical solution is the variational form of (10a)–(10c). Let $\mathcal{S}$ and $\mathcal{V}$ be the spaces

$$\mathcal{S} = \{u : u \in H^1(\Omega), u = g \text{ on } \Gamma_g\},$$
$$\mathcal{V} = \{w : w \in H^1(\Omega), w = 0 \text{ on } \Gamma_g\},$$

where $H^1(\Omega)$ is the Hilbert space defined by

$$H^1(\Omega) = \{f : f \in L_2(\Omega); f_{,i} \in L_2(\Omega), i = 1, 2\}$$

and an inferior comma denotes differentiation with respect to the x_i Cartesian coordinate. $L_2(\Omega)$ is the space of square-integrable functions defined on Ω with inner product

$$(f, g) = \int_\Omega fg d\Omega \quad \forall f, g \in L_2(\Omega),$$

and norm

$$\|f\| = (f, f)^{1/2} \quad \forall f \in L_2(\Omega).$$

The variational form of our boundary-value problem is: Find $u \in \mathcal{S}$ such that

$$a(u, w) = (f, w) + (h, w)_{\Gamma_h} \quad \forall w \in \mathcal{V}, \tag{12}$$

where the symmetric, bilinear form $a(\cdot, \cdot)$ is defined as

$$a(u, w) = \int_\Omega (\nabla u \nabla w + \lambda^2 uw)\, d\Omega. \tag{13}$$

Assume that Ω is partitioned into a set of K quadrilateral elements such that

$$\Omega = \bigcup_{k=1}^{K} \bar{\Omega}^k, \quad \varnothing = \bigcap_{k=1}^{K} \Omega^k,$$

where Ω^k is the interior of the kth element and $\bar{\Omega}^k = \Omega^k \cup \Gamma^k$ is its closure. Any element boundary can be written as

$$\Gamma^k = \bigcup_{i=1}^{4} \Gamma_i^k,$$

where the Γ_i^k's are the edges of the elements.

Let $\mathcal{S}^h \subset \mathcal{S}$ be the space of C^0 piecewise polynomial interpolants of degree $N \otimes N$ which satisfy the essential boundary condition (10b) and $\mathcal{V}^h \subset \mathcal{V}$ a similar space of functions which have value zero on Γ_g. The mesh parameter $h = (K, N)$ depends on both the number of elements and the order of the polynomials used within each one. Here we consider only the case where N is constant throughout the mesh. For completeness, we begin by describing the method for *conforming* elements in which the edge of each subdomain exactly matches its neighboring edges, thus ensuring the solution will be C^0 continuous. This is accomplished by using a Lagrangian interpolant basis.

To complete the spectral element Galerkin approximation to (12), we separate the solution into $u^h = g^h + v^h$, where $g^h \in \mathcal{S}^h$ is a polynomial approximation to g, and $v^h \in \mathcal{V}^h$ is the unknown part of u^h. The solution will be represented locally as

$$v^k(r, s) = \sum_{m=0}^{N} \sum_{n=0}^{N} v_{mn}^k l_m(r) l_n(s) = \sum_{i=1}^{(N+1)^2} v_i^k \bar{l}_i(r, s), \tag{14}$$

where $\bar{l}_i(r, s) \equiv l_m(r) l_n(s)$ is the tensor product of two one-dimensional Gauss–Lobatto–Legendre (GLL) interpolants associated with local node i. Evaluation of the integral form (12) via GLL quadrature gives the elemental matrices

$$A_{ij}^k = a(\bar{l}_i, \bar{l}_j)_{\Omega^k}, \quad F_i^k = (f, \bar{l}_i)_{\Omega^k} + (h, \bar{l}_i)_{\Gamma^k} - a(g^h, \bar{l}_i)_{\Omega^k}, \tag{15}$$

and the discrete Galerkin equation for the kth element as

$$\mathbf{A}^k \mathbf{v}^k = \mathbf{F}^k. \tag{16}$$

To form the global system we define a map $\eta: (i, k) \to (p)$ from local node i to global equation number p so that

$$A_{pq} = \sum_{k=1}^{K}{}' A_{ij}^k, \quad F_p = \sum_{k=1}^{K}{}' F_i^k, \tag{17}$$

where Σ' represents "direct stiffness summation," the process of applying the map to assemble contributions from shared nodes along element interfaces. The final algebraic system

$$A_{pq} v_q = F_p, \quad p, q = 1, \ldots, n_{\text{dof}}, \tag{18}$$

requires the inversion of a symmetric, positive-definite matrix **A** whose bandwidth is determined by the map η.

2.3. Nonconforming Variational Formulation(s)

The conforming restriction is a definite drawback. Although it simplifies mesh generation and data management, a conforming mesh makes it practically impossible to implement features such as local refinement and automatic adaptation. The importance of these aspects of simulating complex physical systems has prompted the development of techniques that permit nonconforming meshes yet retain the convergence properties of global spectral methods. The resulting meshes are *unstructured*, placing no restriction on the size, location, or connectivity of the elemental subdomains.

These nonconforming methods fall under the general classification of patching. The central philosophy is to connect the solution on two subdomains separated by an interface called a *patch*, shown schematically in Fig. 4. Each subdomain may be composed of one or many conforming elements. The patch is constructed from a set of line segments γ^m associated with a particular set of elemental edges Γ_i^k. Its precise definition is

$$\mathcal{P} = \{\gamma : \gamma = \Gamma_i^k \cap \Gamma_j^l \neq \varnothing, |\gamma| = |\Gamma_i^k| \text{ for some } (i,k) \text{ and } (j,l)\},$$

where $|\cdot|$ measures the length of a curve in $\mathcal{R}^2$. This defines a set of nonconforming interfaces, where the conventional C^0 restriction cannot be guaranteed in the discrete solution. Treatment of the solution along these boundaries defines the patching scheme.

In this section we will present two methods for constructing nonconforming discretizations of the elliptic problem introduced in (10a)–(10c). First is the Zanolli iterative procedure, a method originally proposed for spectral collocation schemes to patch subdomains using a C^1 condition across the interface [14]. This algorithm has been modified and incorporated within the variational framework of spectral elements. The second patching scheme is based on a different interpretation of the mortar element method that treats the patching problem in a fully variational manner, allowing the discrete system to be solved directly [7].

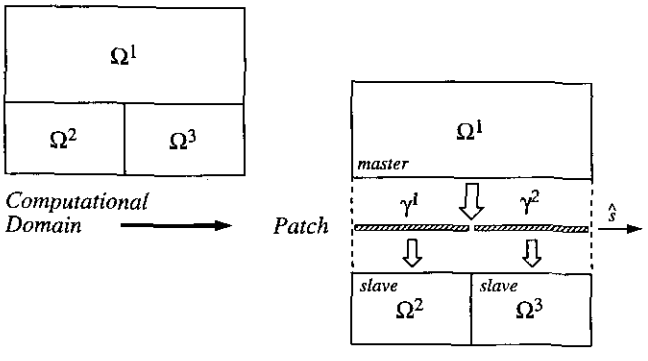

FIG. 4. Schematic of a patch between three subdomains.

2.3.1. *Interface Conditions*

Before describing the patching schemes for two-dimensional problems, we turn briefly to a one-dimensional problem to look at the effects of subdomain interface conditions on the convergence rate of the numerical approximations. The C^0 continuity used so far is optimal for Galerkin projections that adhere to the conforming restriction. The following modified interface conditions extend the continuity requirement to both a weak and strong C^1 condition that results in a tighter coupling between elements but maintains spectral convergence for two-dimensional problems. These modified interface conditions will then form the basis of the patching schemes to supplement or replace the C^0 continuity condition applied across conforming element boundaries. In the following we assume that u satisfies homogeneous boundary conditions.

We are interested in three forms of the discrete system, each of which will be developed for a simple mesh with two elements. First is the standard spectral element Galerkin equation, given by

Method A:

$$\sum_{k=1}^{K}{}' a(u^k, w^k) = \sum_{k=1}^{K}{}' (f, w^k),$$

where w^k is a test function for the kth element. C^0 continuity is enforced through the direct stiffness summation described earlier.

The next form will be referred to as a "mixed" formulation that combines Method A with a weak C^1 condition.

Method B:

$$a(u^1, w^1) + a(u^2, w^1) = (f, w^1) \quad \text{on } \Omega^1,$$

$$-\left(\frac{\partial u^1}{\partial x}, w^2\right) + a(u^2, w^2) = (f, w^2) \quad \text{on } \Omega^2.$$

Recall that each test function w^k vanishes outside of Ω^k, so that the two cross terms are effectively evaluated only at the interface node $x_N^1 = x_b = x_0^2$. Also, the set of weighting functions $\mathcal{V}^h$ is modified by the requirement $w^1(x_b) = 0$. This system enforces C^0 continuity and in addition it explicitly imposes a weak C^1 coupling between elements.

The final form incorporates a strong C^1 condition that gives exact continuity of the derivatives at the interface.

Method C:

$$a(w^1, u^1) + a(w^2, u^2) = (w^1, f) + (w^2, f),$$

$$\left.\frac{\partial u^1}{\partial x}\right|_{x_b} = \left.\frac{\partial u^2}{\partial x}\right|_{x_b}.$$

For this method, members of $\mathcal{V}^h$ must satisfy $w^1(x_b) = 0 = w^2(x_b)$, i.e., the variational form is not tested at the interface.

These three schemes were applied to solve a one-dimensional Helmholtz equation with $\lambda = 1$, with the exact solution given by $u = \exp - x^2 \cos 7\pi x/2$ and appropriate forcing over the domain $-1 \leq x \leq 1$. The mesh is composed of two equal-sized elements. Figure 5 shows convergence of the discrete solution with respect to the total degrees of freedom in the mesh, which is refined by increasing the polynomial order. The "spectral" solution shown for reference is a single domain spectral element formulation and it achieves the best convergence rate. Method A is the best of the three multidomain solutions because the Galerkin formulation is optimal in H^1. Methods B and C destroy this optimality by choosing a progressively "coarser" test space $\mathcal{V}^h$, yet they still achieve the same asymptotic convergence rate. The advantage of these latter two is that they can be generalized to two-dimensional problems, where adjacent elements may not be geometrically conforming.

2.3.2. *Iterative Patching*

To extend the standard spectral element method to allow for a nonconforming interface between elements we need to define a new condition to replace the C^0 continuity applied across element boundaries. Zanolli iteration achieves weak C^1 continuity through an iterative procedure where Dirichlet and Neumann boundary conditions are imposed on opposing sides of the patch [14]. This is the generalization of Method B, and the following version was modified to allow solution updates to be computed in parallel.

Assume that the solution u satisfies homogeneous boundary conditions on Γ and let $g = u^1|_\gamma$ and $h = \mathbf{n} \cdot \nabla u^2|_\gamma$ be the *unknown* boundary conditions along the patch $\gamma = \overline{\Omega}^1 \cap \overline{\Omega}^2$, with u^1 and u^2 each being the solution of an appropriate variational problem in the domains Ω^1 and Ω^2, respectively. For each of the unknowns, let f_n be a sequence of approximations generated from the following algorithm. At the beginning of subdomain iteration n, approximate patch boundary conditions are computed using

$$g_n = \theta g_{n-1} + (1 - \theta)\mathbf{I}^{12} u^2_{n-1}|_\gamma, \quad h_n = \mathbf{I}^{21}\mathbf{n} \cdot \nabla u^1_{n-1}|_\gamma, \quad (19)$$

where θ is a relaxation parameter, and $\mathbf{I}^{kl}$ is a general interpolation operator from Ω^l to Ω^k along γ. Next, a new approximation is computed independently on each subdomain by solving a standard Galerkin problem with appropriate Dirichlet or Neu-

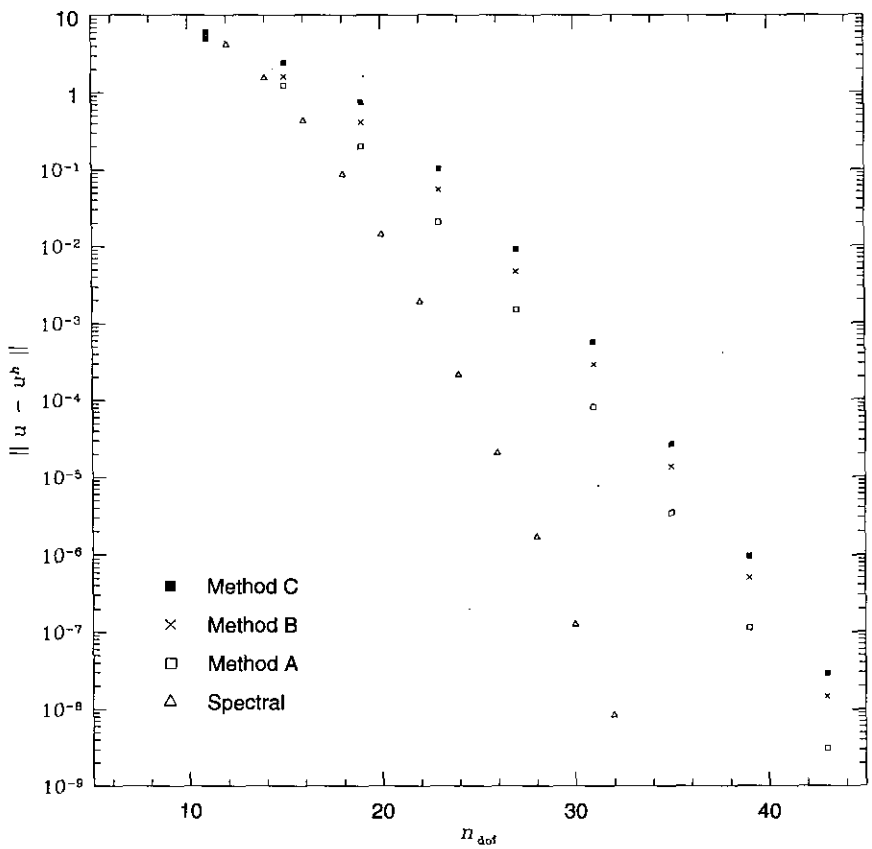

FIG. 5. Convergence in the H^1 norm for various interface conditions from a pure GLL spectral method (no interface) to a strong C^1 condition.

mann boundary conditions. The resulting interface errors on each domain boundary are defined as

$$e_n^1 = g_n - g_{n-1}, \quad e_n^2 = (u_n^2 - u_{n-1}^2)|_\gamma. \tag{20}$$

If either error exceeds a specified tolerance, then a new relaxation parameter is computed using

$$\theta = \frac{(e_n^1, e_n^1 - \mathbf{I}^{12} e_n^2)}{\|e_n^1 - \mathbf{I}^{12} e_n^2\|^2} \tag{21}$$

and a new iteration begins with $n \leftarrow n + 1$. The relaxation parameter guarantees not only that the procedure will converge independently of the relative subdomain size ($g = \lim_{n\to\infty} g_n$) but also that it will converge *rapidly*, typically within 5–10 iterations [15]. Since each subdomain problem is necessarily well-posed, convergence of (19) guarantees that $u_n^k \to u^k$ on Ω^k. The final solution is "weakly" C^1 continuous (as defined by the integral form of the Neumann boundary condition) and satisfies a coupled set of equations analogous to Method B discussed above. Consequently, the solution space $\mathcal{S}^h$ is the same as in the conforming case—the patch simply introduces an internal cut where the boundary conditions are generated iteratively using (19). The test space $\mathcal{V}^h$ is coarser because the variational form is never tested along γ in Ω^1.

Since the algorithm requires the solution to a variational problem on each subdomain but not a particular discretization, it can be used in a variety of ways. It is ideal for both unstructured and *hybrid* computations; we have used this approach successfully to couple spectral elements with finite element and finite difference schemes [15]. Each subdomain may be recursively partitioned as a way of performing local refinement, the details of which are "hidden" from other patched subdomains. Generalization to an arbitrary number of subdomains is straightforward, although perhaps not efficient for a large number of subdomains.

A final note on implementation. This is an inherently parallel algorithm, and the approach we have taken is based on the parallel virtual machine (PVM) communications model [16]. A host program monitors the errors along the patch and generates successive boundary conditions. It sends these boundary conditions to a set of concurrently executing processes that update the solution and send back new boundary data. Each concurrent process is a standard spectral element solver with a simple interface to the host. Efficiency is achieved by using a large number of concurrent solvers.

2.3.3. *Variational Patching*

The next method does not impose a continuity requirement between adjacent elements but rather selects the "most continuous" solution consistent with a set of patching conditions. It is the generalization of Method A, where the C^0 requirement between elements is found as a particular solution to a coupled minimization problem. This approach was first introduced by Bernardi *et al.* who coined the term "mortar element method" because the discretization introduces a set of functions that mortar the brick-like spectral elements together [7, 17, 18].

Since the trial space $\mathcal{S}^h$ now admits discontinuous functions, the spaces $\mathcal{S}^h$ and $\mathcal{S}$ are nonconforming. Members of $\mathcal{S}^h$ cannot be measured in $H^1(\Omega)$, so we introduce the alternative

$$H^*(\Omega) = \{f : f \in H^1(\Omega^k), \quad \forall k = 1, \ldots, K\}$$

with inner product

$$(f, g)_* = \sum_{k=1}^{K} \int_{\Omega^k} (fg + f_{,i} g_{,i})\, d\Omega \quad \forall f, g \in H^*(\Omega).$$

and norm

$$\|f\|_* = (f, f)_*^{1/2} \quad \forall f \in H^*(\Omega).$$

All error analysis will be made in terms of $H^*(\Omega)$, sometimes called a "broken norm" since integral forms over Ω are broken into a sum of integrals over each Ω^k.

This formulation generalizes the treatment of the solution along element boundaries in the mesh, defined as

$$\tilde{\Gamma} = \bigcup_{k=1}^{K} \Gamma^k = \bigcup_{m=1}^{M} \gamma^m,$$

where each γ^m is called a "mortar" and corresponds to an arbitrary subset of the Γ^k's. We now introduce a new space of continuous functions $\mathcal{M}$ defined over $\tilde{\Gamma}$,

$$\mathcal{M} = \{\phi : \phi \in C^0(\tilde{\Gamma}),\ \phi = g \text{ on } \Gamma_g\},$$

which represent additional "unknowns" to be determined by the variational statement of the problem.

The discrete trial, test, and mortar spaces are now:

$$\mathcal{S}^h = \{u^h : u^h \in H^*(\Omega)\},$$
$$\mathcal{V}^h = \{w^h : w^h (H)^*(\Omega),\ w^h = 0 \text{ on } \Gamma_g\},$$
$$\mathcal{M}^h = \{\phi^h : \phi^h \in C^0(\tilde{\Gamma}),\ \phi^h = g \text{ on } \Gamma_q\}.$$

Members of $\mathcal{S}^h$ and $\mathcal{V}^h$ are represented as before, while members of $\mathcal{M}^h$ are represented as Nth-order polynomials over each mortar γ^m. In terms of these spaces, the variational prob-

lem to solve is: Find $u^h \in \mathcal{S}^h$, $\phi^h \in \mathcal{M}^h$, such that for every $w^h \in \mathcal{V}^h$

$$a(u^h, w^h) = (f, w^h) + (h, w^h) \tag{23a}$$

$$\int_{\gamma^m} (u^h - \phi^h)\psi \, ds = 0 \quad \forall \psi \in P_{N-2}(\gamma^m), \quad \forall \gamma^m \tag{23b}$$

$$(u^h - \phi^h)\big|_{s=0,\bar{\gamma}^m} = 0 \quad \forall \gamma^m, \tag{23c}$$

where $s = 0$, $\bar{\gamma}^m$ are the endpoints of a mortar, and $P_N(\gamma)$ is the space of the Nth-order polynomials defined over γ. The test function is only a polynomial of order $N - 2$ because the endpoint condition removes two degrees of freedom from the constraint.

To solve the discrete equations, first we use (23b)–(23c) to express the unknown values of u^h around the boundary of each element in terms of the mortar function ϕ. In "segment coordinates," let (z_i^k, u_i^k, ϕ_i^m) be nodal values of the mesh, solution and mortar interpolants along the edge of element k, and segment m such that $\Lambda \equiv \Gamma^k \cap \gamma^m \neq \emptyset$; this defines the path of integration for Eq. (23b). Using an Nth-order quadrature rule, we can define the mass and projection matrices for the segment

$$B_{ij} \leftarrow \int_\Lambda u^k \psi_i \, ds, \quad P_{ij} \leftarrow \int_\Lambda \phi^m \psi_i \, ds. \tag{24}$$

In terms of these matrices, Eqs. (23b)–(23c) can be expressed as

$$u_0^k = \sum_{j=0}^{N} \phi_j^m l_j(z_0^k) \tag{25a}$$

$$\sum_{i=1}^{N-1} B_{ij} u_j^k = \sum_{i=1}^{N-1} P_{ij} \phi_j^m \tag{25b}$$

$$u_N^k = \sum_{j=0}^{N} \phi_j^m l_j(z_N^k) \tag{25c}$$

or, by defining $\mathbf{Z} = \mathbf{B}^{-1}\mathbf{P}$, as the single matrix–vector relation:

$$u_i^k = \sum_{j=0}^{N} Z_{ij}^m \phi_j^m. \tag{26}$$

This algebraic equation expresses the solution along the edge of an element in terms of a mortar function, thus eliminating the boundary nodes of u^k as "unknowns" in the mesh.

The projection matrix and mortar values for each edge are assembled as $\tilde{\mathbf{Z}} = \Sigma' \mathbf{Z}^m$, $\tilde{\boldsymbol{\phi}} = \Sigma' \boldsymbol{\phi}^m$. If $\mathbf{u}_b$ and $\mathbf{u}_i$ are the nodal values on the boundary and interior of u^k, we can write the elemental unknowns in vector form as

$$\mathbf{u}^k = \begin{bmatrix} \mathbf{u}_b \\ \mathbf{u}_i \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{Z}} & \\ & \mathbf{I} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\phi}} \\ \mathbf{u}_i \end{bmatrix} = [\mathbf{Z}^k]\, \tilde{\mathbf{u}}^k, \tag{27}$$

where $\tilde{\mathbf{u}}^k$ represents the true "unknowns" for an element: projected mortar values and internal nodal values. Finally, we use (27) to write the Galerkin equations for a single element in the least squares form:

$$[\mathbf{Z}^k]^T(\mathbf{F}^k - \mathbf{A}^k[\mathbf{Z}^k]\tilde{\mathbf{u}}^k) = 0. \tag{28}$$

The global system is assembled by summing the elemental matrices

$$\mathbf{A} = \sum_{k=1}^{K}{}' [\mathbf{Z}^k]^T \mathbf{A}^k [\mathbf{Z}^k], \quad \mathbf{F} = \sum_{k=1}^{K}{}' [\mathbf{Z}^k]^T \mathbf{F}^k, \tag{29}$$

resulting in a global system $\mathbf{Au} = \mathbf{F}$ identical to that for a conforming mesh, but whose solution determines values at the mortar nodes (boundary solution) and internal element nodes (interior solution).

Belgacem and Maday recently proposed a new version of the mortar element method that relaxes the condition of strict continuity at vertices expressed in (23c), citing as their motivation a "difficulty in the implementation" because of the multiplicity of such nodes [19]. They show that errors are still bounded asymptotically in the same way as both the method considered here and the conforming method, but the estimate is not sharp enough to predict the increase in consistency error. Our experience has been that with a proper choice of data structures this condition does not cause significant problems and that enforcing it increases the accuracy of the discrete solution.

2.3.4. *Eigenspectra of Advection and Diffusion Operators*

Next we look at spectra of the discrete advection and diffusion operators on unstructured meshes to verify that the desirable approximation properties of conforming discretizations have not been sacrificed. First consider the linear advection equation,

$$\frac{\partial u}{\partial t} = \mathbf{a} \cdot \nabla u \quad \text{on } \Omega, \tag{30}$$

where u is a scalar and $\mathbf{a}$ is a given velocity vector field defined on Ω. For simplicity we assume $|\mathbf{a}| = 1$ pointwise. The weak form of (30) is: Find $u^h \in \mathcal{S}^h$ such that for all $w^h \in \mathcal{V}^h$

$$\int_\Omega w^h(\dot{u} - \mathbf{a} \cdot \nabla u^h) \, d\Omega = 0, \tag{31}$$

where $\dot{u} = \partial u^h / \partial t$. The least squares form of the elemental system is

$$[\mathbf{Z}^k]^T(\mathbf{B}^k[\mathbf{Z}^k]\dot{\mathbf{u}}^k - \mathbf{D}^k[\mathbf{Z}^k]\tilde{\mathbf{u}}_k) = 0, \tag{32}$$

where the elemental mass and advection matrices are

$$D_{ij}^k = (\mathbf{a} \cdot \nabla \bar{l}_i, \bar{l}_j)_{\Omega^k}, \quad B_{ij}^k = (\bar{l}_i, \bar{l}_j)_{\Omega^k}. \tag{33}$$

The solution to the global system is $\dot{\mathbf{u}} = \mathbf{B}^{-1}\mathbf{D}\mathbf{u}$. The global matrices are assembled in the same way as indicated in (29):

$$\mathbf{B} = \sum_{k=1}^{K}{}' [\mathbf{Z}^k]^T \mathbf{B}^k [\mathbf{Z}^k], \quad \mathbf{D} = \sum_{k=1}^{K}{}' [\mathbf{Z}^k]^T \mathbf{D}^k [\mathbf{Z}^k]. \tag{34}$$

To avoid inversion of the full mass matrix $\mathbf{B}$, which is sparse but not diagonal, we can approximate it with the lumped matrix

$$\tilde{B}_{pp} = \sum_{q=1}^{n_{\text{dof}}} B_{pq}, \quad p = 1, \ldots, n_{\text{dof}}. \tag{35}$$

This is a good approximation since the consistent mass matrix is already diagonal *except* for the portion associated with the mortar nodes, but we will verify this in the following analysis.

For skew-symmetric boundary conditions, the advection operator becomes skew-symmetric, i.e., $(\mathbf{a} \cdot \nabla u, w) = -(u, \mathbf{a} \cdot \nabla w)$. Such systems are nondissipative and possess purely imaginary eigenvalues, in this case determined by the nontrivial solutions (λ, u) of

$$(\mathbf{a} \cdot \nabla - \lambda)u = 0. \tag{36}$$

Skew-symmetric boundary conditions are any boundary conditions that guarantee the surface integral $\int_\Gamma (\mathbf{n} \cdot \mathbf{a}u)w \, d\Gamma = 0$, so there is zero net flux of u across Γ. For all other boundary conditions the eigenvalues are complex and depend on the boundary data and $\mathbf{a}$. Eigenvalues of the discrete problem are determined by the system

$$(\mathbf{G} - \lambda \mathbf{I})\mathbf{u} = 0, \tag{37}$$

where $\mathbf{G} = \mathbf{B}^{-1}\mathbf{D}$.

Figures 6 and 7 compare eigenvalues of the discrete advection

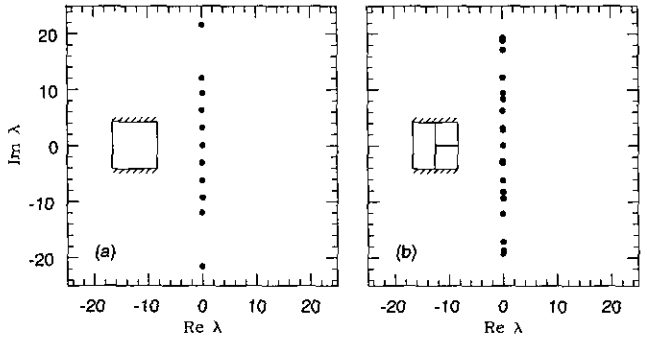

FIG. 6. Computed eigenvalues of the discrete advection operator on a domain with skew-symmetric boundary conditions: (a) conforming mesh with $K = 1$, $N = 11$, $n_{\text{dof}} = 110$; (b) nonconforming mesh with $K = 3$, $N = 7$, $n_{\text{dof}} = 133$.

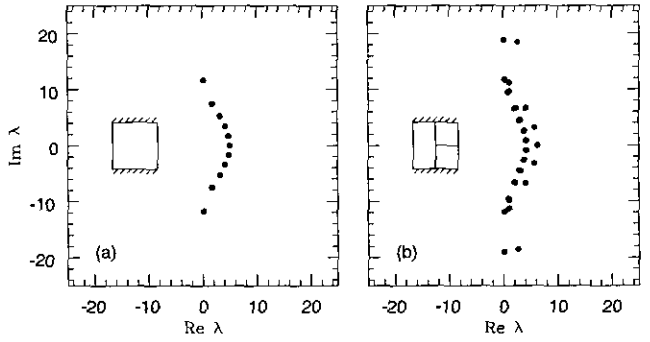

FIG. 7. Computed eigenvalues of the discrete advection operator on a domain with inflow–outflow boundary conditions; the mesh is the same as in the previous figure.

operator in a simple channel domain discretized using conforming and nonconforming meshes with approximately the same number of unknowns. The geometry of the domain is simply $[-1, 1] \times [-1, 1]$. Each eigenspectrum was computed using the double-precision version of LAPACK's eigenvalue package for nonsymmetric matrices [20]. This verifies that the unstructured advection operator has purely imaginary eigenvalues for the periodic problem and complex eigenvalues for the inflow–outflow problem of magnitude comparable to those of the structured mesh operator. Figure 8 further verifies that the maximum eigenvalue for both methods scales in the same way as in global spectral methods, i.e.,

$$|\lambda|_{\max} \sim \mathcal{O}(N^2). \tag{38}$$

Since the nonlinear advection equation (5a) will be treated explicitly, the eigenvalues of the related matrix $(\mathbf{I} + \Delta t \mathbf{G})$ must lie within the stability region of the discrete time-stepping scheme; this last result confirms that the time step restriction on an unstructured mesh scales the same way as that for a structured mesh. Our final result, in Fig. 9, shows that the lumped mass matrix causes some dispersion at the highest wave numbers, but it does not affect the eigenvalues near the origin.

Next we look at the diffusion operator developed in the previous section. In this case we are interested only in the condition number of $\mathbf{A}$ since this determines the convergence rate for iterative solution techniques. Heuristically, we can show that the condition number of the conforming spectral element Laplacian scales as

$$\kappa_A \sim \mathcal{O}(KN^3). \tag{39}$$

This estimate assumes that the size of the smallest element in the mesh is $\sim 1/K$; if the mesh contains smaller elements then κ_A will be larger. Using a forward and inverse power method, we computed condition numbers for a series of meshes that combine elements of various sizes. Figure 10 shows a family of curves for these domains, each one exhibiting a growth

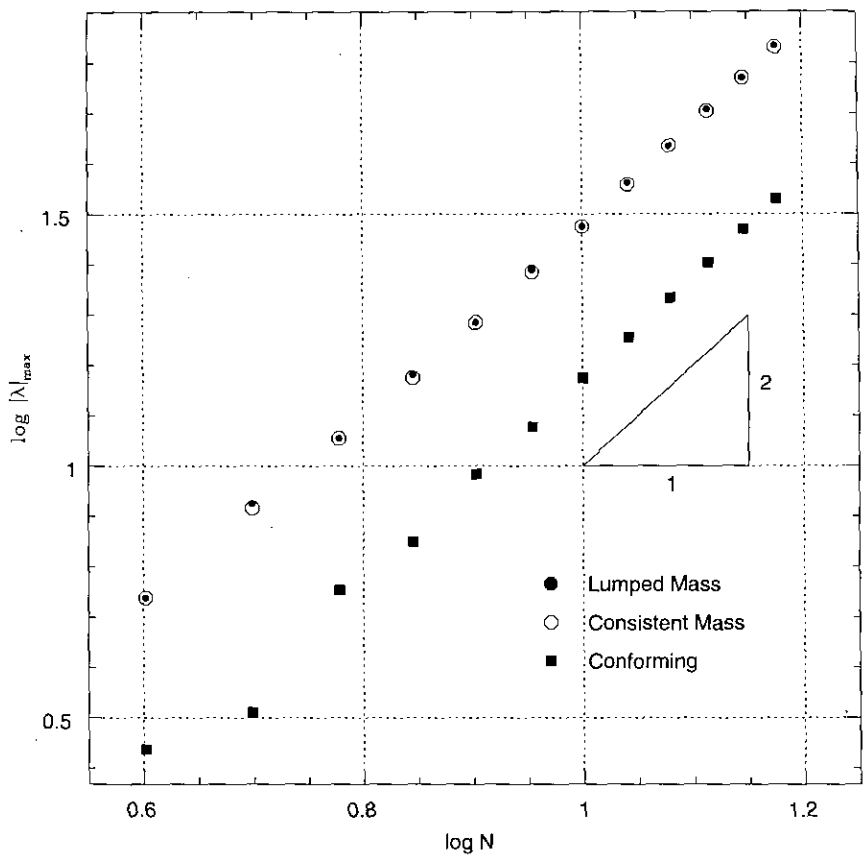

FIG. 8. Growth rate of the maximum eigenvalue of the discrete advection operator **G** on conforming and nonconforming meshes versus polynomial order N.

rate which is $\mathcal{O}(N^3)$. Comparing domains with similarly sized elements, we see that **A** contains a blend of the eigenvalues from the largest and smallest, but overall adheres to the estimate given in Eq. (39). This figure also shows that *local* refinement of a mesh produces a matrix with a lower condition number than that obtained by global refinement with the same element size.

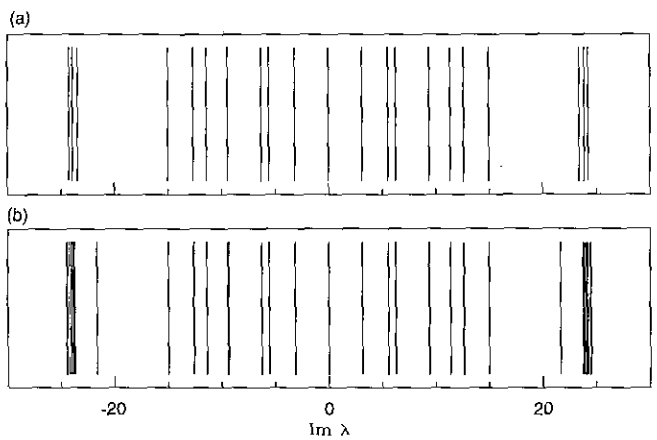

FIG. 9. Computed eigenvalues of the discrete advection operator on an unstructured mesh using $N = 8$: (a) consistent mass matrix; (b) lumped mass matrix.

3. COMPUTATIONAL COMPLEXITY AND SOLUTION ALGORITHMS

In this section we will describe efficient iterative and direct methods for inverting the large algebraic systems that result from nonconforming spectral element discretizations. Iterative methods are more appropriate for steady-state calculations or calculations involving variable properties, such as a changing time step or a Helmholtz equation with a variable coefficient. For direct methods the issue is one of memory management—storing **A** as efficiently as possible without sacrificing the performance needed for fast back-substitution. The development of fast direct and well-preconditioned iterative solvers represents a major advance towards the application of nonconforming spectral element methods to the simulation of turbulent flows on unstructured meshes.

3.1. Conjugate Gradient Iteration

Conjugate gradient methods [21] have been particularly successful with spectral elements because the tensor–product form

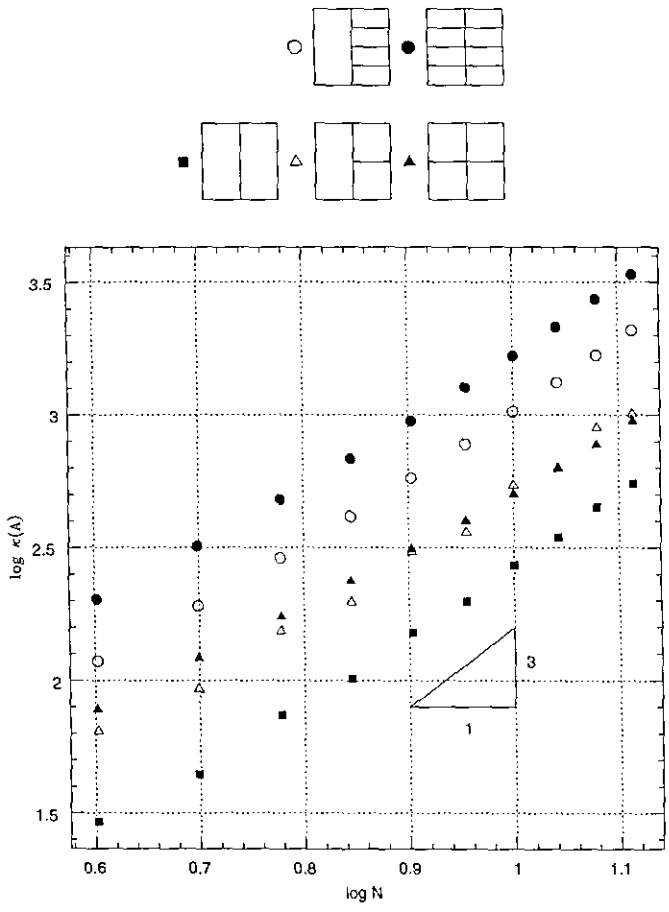

FIG. 10. Condition numbers of the discrete Laplace matrix for a family of conforming and nonconforming spectral element meshes.

and local structure allows the global Helmholtz inner product to be evaluated using only elemental matrices. To solve the system $\mathbf{Au} = \mathbf{F}$ by the method of conjugate gradients we use the algorithm

$$
\begin{aligned}
&k = 0;\ u_0 = 0;\ r_0 = \mathbf{F};\\
&\textbf{while } r_k \neq 0\\
&\qquad \text{Solve } \mathbf{M}q_k = r_k;\ k = k + 1\\
&\qquad \textbf{if } k = 1\\
&\qquad\qquad p_1 = q_0\\
&\qquad \textbf{else}\\
&\qquad\qquad \beta_k = r_{k-1}^{\mathrm{T}} q_{k-1} / r_{k-2}^{\mathrm{T}} q_{k-2}\\
&\qquad\qquad p_k = q_{k-1} + \beta_k p_{k-1}\\
&\qquad \textbf{end}\\
&\qquad \alpha_k = r_{k-1}^{\mathrm{T}} q_{k-1} / p_k^{\mathrm{T}} \mathbf{A} p_k\\
&\qquad r_k = r_{k-1} - \alpha_k \mathbf{A} p_k\\
&\qquad u_k = u_{k-1} + \alpha_k p_k\\
&\textbf{end}\\
&\mathbf{u} = u_k
\end{aligned}
$$

where k is the iteration number, r_k is the residual, and p_k is the current search direction. The matrix $\mathbf{M}$ is a preconditioner used to improve the convergence rate of the method and is discussed in detail next.

Selection of a good preconditioner is critical for rapid convergence; the preconditioner must be spectrally close to the full stiffness matrix yet easy to invert. Popular preconditioners for spectral methods include incomplete Cholesky factorization and low-order (finite element, finite difference) approximations [22, 23]. Unfortunately, these preconditioners can be as complicated to construct for an unstructured mesh as the full stiffness matrix $\mathbf{A}$. Next we present three preconditioners which are simple to build and apply even when the mesh is unstructured.

In conjugate gradient methods the number of iterations required to reach a given error level scales as $\sqrt{\kappa_A}$. This is only an estimate, since the actual convergence rate is determined by the *distribution* of eigenvalues—if all of $\mathbf{A}$'s eigenvalues are clustered together, convergence is much faster. To assess the effectiveness of a given preconditioner we begin by looking at the condition number of $\mathbf{M}^{-1}\mathbf{A}$.

Each of the following methods is based on selecting a subset of entries from the full stiffness matrix. The first two preconditioners are diagonal matrices given by

$$M_{ii} = A_{ii} \qquad \text{"diagonal,"} \tag{40}$$

$$M_{ii} = \sum_{j=0}^{n_{\mathrm{dof}}} |A_{ij}| \qquad \text{"row-sum,"} \tag{41}$$

where n_{dof} = rank ($\mathbf{A}$); the diagonal (40) is sometimes called a point Jacobi preconditioner. Both are direct estimates of the spectrum of $\mathbf{A}$ and have the advantage of minimal storage and work. They can be quite effective for diagonally dominant systems such as the viscous correction step of the splitting scheme described in Section 2.1. The third preconditioner is a block-diagonal matrix

$$M_{ij} = \begin{cases} |A_{ij}|, & \text{if } i \leq n_{\mathrm{bof}}, j = i \\ 0, & \text{if } i \leq n_{\mathrm{bof}}, j \neq i, \text{ "block-diagonal,"} \\ A_{\mathrm{ij}}, & \text{otherwise,} \end{cases} \tag{42}$$

where n_{bof} is the number of mortar nodes in the mesh. The structure of this matrix assumes that $\mathbf{A}$ is arranged in the static condensation format described in Section 3.2. Applying this preconditioner amounts to storing and inverting the isolated blocks of $\mathbf{A}$ associated with the degrees of freedom on the interior of each element, while applying a simple diagonal matrix to the mortar nodes.

The following test examines the iterative solution to a Helmholtz equation for the two extreme cases $\lambda^2 = 1$ and $\lambda^2 = 10{,}000$. Convergence is measured with respect to the solution $u(x, y) = \sin \pi x \sin \pi y$. The mesh has $K = 10$ elements generated by recursively subdividing a square domain, with $N = 15$ in each element. Figures 11 and 12 show the convergence history

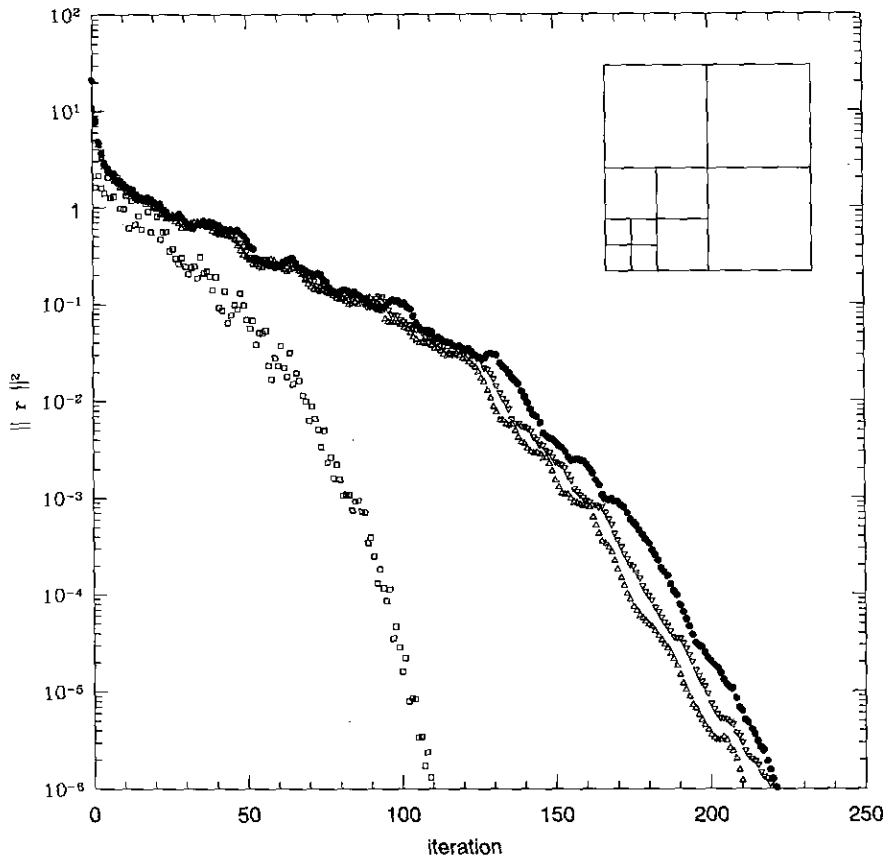

FIG. 11. Conjugate gradient iteration convergence history for a Helmholtz equation with $\lambda^2 = 1$: ● = none, △ = diagonal, ▽ = row-sum, and □ = block-diagonal preconditioner.

for the weakly and strongly diagonally dominant systems. The difference in convergence rates is explained in part by the condition numbers of $\mathbf{M}^{-1}\mathbf{A}$, given in Table I and Table II. In spite of yielding a lower κ_A, the row-sum preconditioner converges slower and therefore offers no particular advantage over the simpler diagonal preconditioner. The block-diagonal matrix performs significantly better than the other two, effectively doubling the convergence rate in both cases. This preconditioner is fully parallelizable and offers the most promise in distributed computing environments, where the cost per iteration can include significant time performing interprocessor communication; its main drawbacks are the higher operation count and storage requirement. The methods described in the next section for implementing fully direct solvers can also be used to reduce the storage requirement for the block-diagonal preconditioner.

We conclude this section by giving the memory requirements and computational complexity for a preconditioned conjugate gradient (PCG) solver. Since the elemental Helmholtz operator can be evaluated using only the one-dimensional Lagrangian derivative matrix, the required memory is simply storage for the nodal values and geometric factors:

$$S_I = s_1 KN^2. \tag{43}$$

As mentioned above, the dominant numerical operations are vector–vector and matrix–vector products, although the derivative calculations are folded into a more efficient matrix–matrix multiplication. The operation count for the entire solver is

$$C_I = J^\varepsilon [c_1 KN^3 + c_2 KN^2 + c_3 KN], \tag{44}$$

where $J^\varepsilon \propto \sqrt{KN^3}$ is the number of iterations required to reach a given error level ε. Our numerical results (Tables I and II) show that with these preconditioners J^ε is still proportional to KN^3, but the constant is reduced. The block matrix operations required to compute the elemental inner products provide good data locality and can be coded efficiently on both vector processors and RISC microprocessors.

3.2. Static Condensation

The static condensation algorithm is a method for reducing the complexity of the stiffness matrices arising in finite element and spectral element methods. Static condensation is particularly attractive for unstructured spectral element methods because of the natural division of equations into those for boundaries (mortars) and element interiors. To apply this method to the discrete Helmholtz equation, we begin by writing Eq. (28) in the form

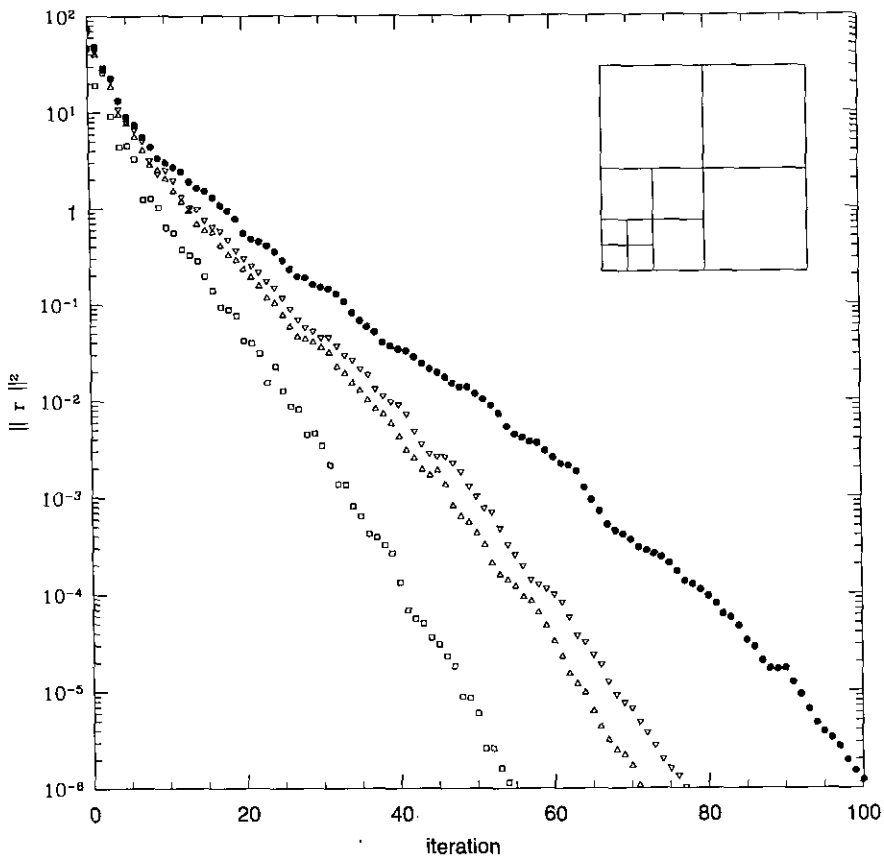

FIG. 12. Conjugate gradient iteration convergence history for a Helmholtz equation with $\lambda^2 = 10{,}000$: ● = none, △ = diagonal, ▽ = row-sum, and □ = block-diagonal preconditioner.

$$\begin{bmatrix} \tilde{Z}^T A_{11} \tilde{Z} & \tilde{Z}^T A_{12} \\ A_{21} \tilde{Z} & A_{22} \end{bmatrix}^k \begin{bmatrix} \tilde{\phi} \\ \mathbf{u}_i \end{bmatrix}^k = \begin{bmatrix} \tilde{Z}^T \mathbf{F}_b \\ \mathbf{F}_i \end{bmatrix}^k, \tag{45}$$

where A_{11} is the boundary matrix, $A_{12} = [A_{21}]^T$ is the coupling matrix, and A_{22} is the interior matrix. This system can be factored into one for the mortar nodes and one for the interior nodes, so that on Ω^k:

$$([\tilde{Z}^T A_{11} \tilde{Z}] - [\tilde{Z}^T A_{21}][A_{22}]^{-1}[A_{12} \tilde{Z}])\tilde{\phi} = \tilde{Z}^T \mathbf{F}_b - [\tilde{Z}^T A_{21}][A_{22}]^{-1} \mathbf{F}_i \tag{46}$$

$$\mathbf{u}_i = [A_{22}]^{-1} \mathbf{F}_i - [A_{22}]^{-1}[A_{21} \tilde{Z}]\tilde{\phi}. \tag{47}$$

TABLE I

Condition Numbers of $M^{-1} A$ for a Helmholtz Equation with $\lambda^2 = 1$

N	None	Diagonal	Row-sum	Block-diagonal
5	177	70	46	34
6	278	108	70	52
7	404	155	99	75
8	558	211	135	104
9	743	277	177	139
10	963	354	226	180
15	3042	961	677	517

TABLE II

Condition Numbers of $M^{-1} A$ for a Helmholtz Equation with $\lambda^2 = 10{,}000$

N	None	Diagonal	Row-sum	Block-diagonal
5	325	18.1	17.9	7.37
6	283	20.1	19.6	8.20
7	273	22.1	21.4	8.71
8	247	23.4	22.4	9.44
9	237	25.1	23.7	10.43
10	229	27.1	25.2	11.82
15	243	44.3	36.1	24.40

During a preprocessing phase, the global boundary matrix is assembled by summing the elemental matrices,

$$\mathbf{A}_{11} = \sum_{k=1}^{K}{}' ([\tilde{Z}^T A_{11} \tilde{Z}] - [\tilde{Z}^T A_{21}][A_{22}]^{-1}[A_{12}\tilde{Z}])^k, \qquad (48)$$

and prepared for the solution phase via LU factorization. Equation (48) may also be recognized as the Schur complement of A_{22} in A. As part of this phase we also compute and store for each element the inverse of the interior matrix $[\mathbf{A}_{22}]^{-1}$ and its product with the coupling matrix $[\tilde{Z}^T A_{21}][A_{22}]^{-1}$. The system is solved by setting up the modified right-hand side of the global boundary equations, solving the boundary equations using back-substitution, and then computing the solution on the interior as the matrix–vector product indicated in (28). Because the coupling between elements is only C^0, the element interiors are independent of each other and on a multiprocessor system this final stage can be solved concurrently.

Figure 13 illustrates the structure of a typical spectral element stiffness matrix factored using this approach. To reduce computational time and memory requirements for the boundary phase of the direct solver, we wish to find an optimal form of the discrete system corresponding to a minimum bandwidth for the matrix $\mathbf{A}_{11}$. This is complicated by the irregular connectivity generated by the patching. Our approach to bandwidth optimization reduces the problem to that of finding an optimal path through the mesh that visits "nearest neighbors." During each of the K stages of the optimization, an estimate is made of the new bandwidth that results from adding one of the unnumbered elements to the current path. The element corresponding to the largest increase is chosen for numbering, resulting in what is essentially a Greedy algorithm. This basic concept is illustrated in Fig. 14. The reduction in bandwidth translates to direct savings in memory and quadratic savings in computational cost.

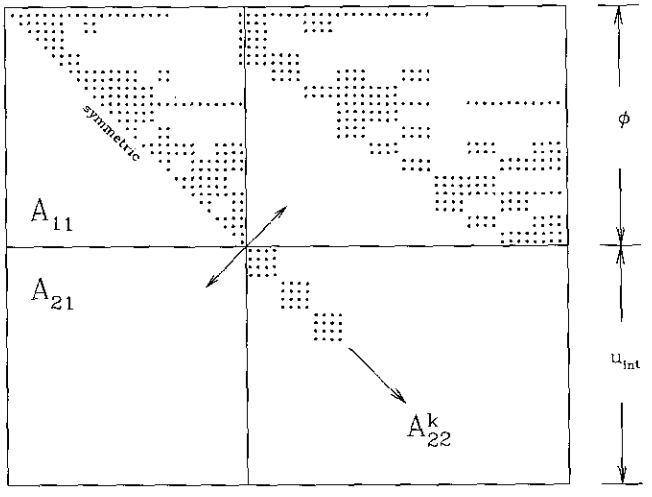

FIG. 13. Static condensation form of the spectral element stiffness matrix.

The search for an optimal numbering system can be accomplished during preprocessing, so the extra work has no impact on the simulation cost and can result in significant savings. Table III shows the results of bandwidth optimization for each of the computational domains pictured in Fig. 15. For computers where memory is a limitation, this procedure can determine whether an in-core solution is even possible. Other simple memory optimizations include storage of only a single copy of the interior and coupling matrices for each element with the same geometry and evaluation of the force vector $\mathbf{F}$ using tensor product summation instead of matrix operations. By carefully organizing memory usage, the overall memory requirement scales as

$$S_D = \tfrac{1}{2}s_1 K^2 N^2 + s_2 K N^3 + s_3 K N^4. \qquad (49)$$

As mentioned in the introduction to this section, the direct solver is advantageous only when the cost of factoring this stiffness matrix can be spread over a large number of solutions. Therefore, we consider only the cost of a back-substitution using the factored stiffness matrix, for which the operation count scales as

$$C_D = c_1 K^{3/2} N^2 + c_2 K N^4 + c_3 K N. \qquad (50)$$

For a well-conditioned, diagonally dominant system this method usually results in at least a factor of two savings versus an iterative solver. For a system that is not diagonally dominant, like the Navier–Stokes pressure equation, it can be faster by a full order of magnitude. In the next section we compare specific timings for an unsteady simulation.

4. NUMERICAL SIMULATIONS

4.1. Illustrations

Each of the following examples addresses a different issue: accuracy, resolution of singular behavior, and computational efficiency for large-scale fluid flow problems. Here we center our attention on the two-dimensional nonconforming discretizations that provide the foundation for other (more complicated) simulations.

A. *Kovasznay Flow*

In 1948, Kovasznay solved the problem of steady, laminar flow behind a two-dimensional grid [24]. This exact solution to the Navier–Stokes equations is given by

$$u = 1 - e^{\lambda x}\cos 2\pi y, \quad v = \frac{\lambda}{2\pi} e^{\lambda x} \sin 2\pi y,$$
$$p = \frac{1}{2}(1 - e^{\lambda x}), \qquad (51)$$

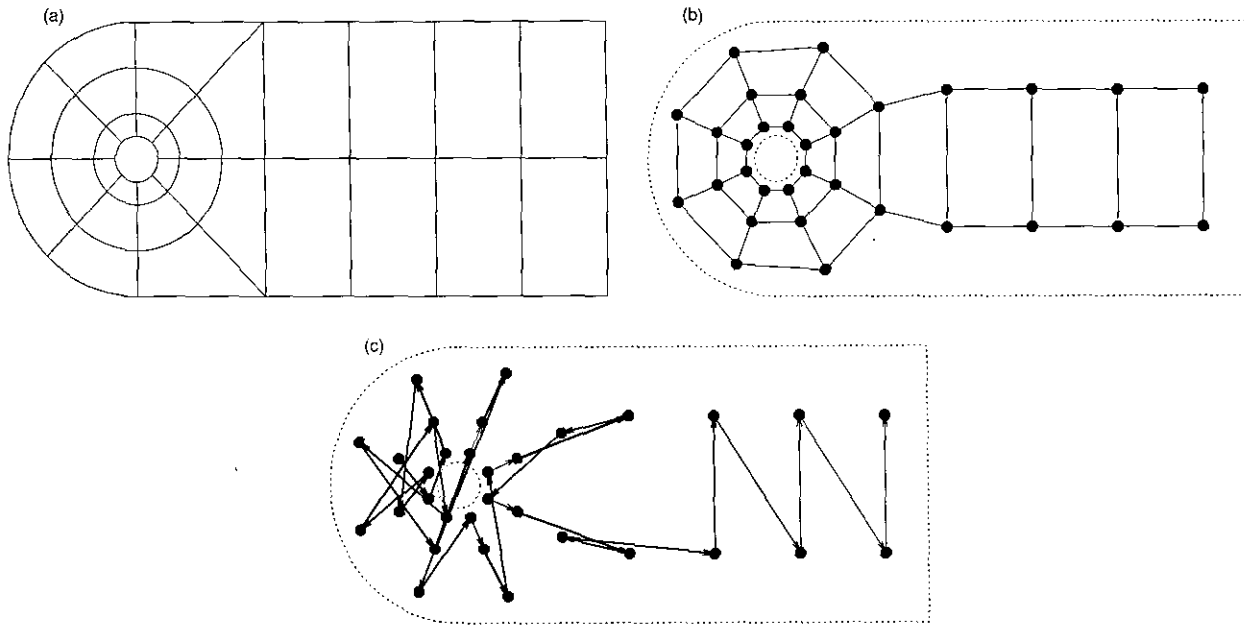

FIG. 14. Bandwidth optimization for a spectral element mesh: (a) computational domain; (b) connectivity graph; and (c) an optimal path for numbering the boundary nodes in the mesh. Line thickness demonstrates the change in global bandwidth with each step.

where $\lambda = \mathrm{Re}/2 - (\mathrm{Re}^2/4 + 4\pi^2)^{1/2}$. Streamlines of the flow field are similar to the wake of a cylinder array at low Reynolds numbers.

Using the family of meshes shown in Fig. 16, the two-dimensional Navier–Stokes equations were solved using the velocity boundary conditions indicated in Eq. (51) on the perimeter of the domain. The computed velocity field converges exponentially in the H^* norm for all discretizations. Since the nonconforming mesh contains a mixture of the largest and smallest elements of the conforming domains, we expect the error to be bounded by the dashed lines (a least squares fit to $\|u - u^h\|_*$) in Fig. 16. This "numerical proof" shows that the additional consistency error associated with the nonconforming space $\mathcal{S}^{ph}$ is small compared to the other discretization errors.

Figure 16 demonstrates another important point: the two patching methods presented here give nearly indistinguishable results for smooth problems. In this example the variation is less than 0.1% over the range of N considered. While Bernardi *et al.* have proven that the mortar approximation is optimal in the H^* norm defined above [7], the results presented here do not contradict this fact because the Zanolli iteration is applied to a different approximation space using standard Galerkin projections. While no mathematical connection has yet been made between the two techniques, numerical tests in a variety

TABLE III

Matrix Rank and Optimized Bandwidth of Three Complex-Geometry Domains Representative of Internal and External Flow Problems

Mesh	K	N	Rank	Original	Optimized	Savings
riblets	91	9	1484	1483	250	83%
cylinder	114	11	2416	2406	402	83%
half-cylinder	176	7	2177	2156	399	81%

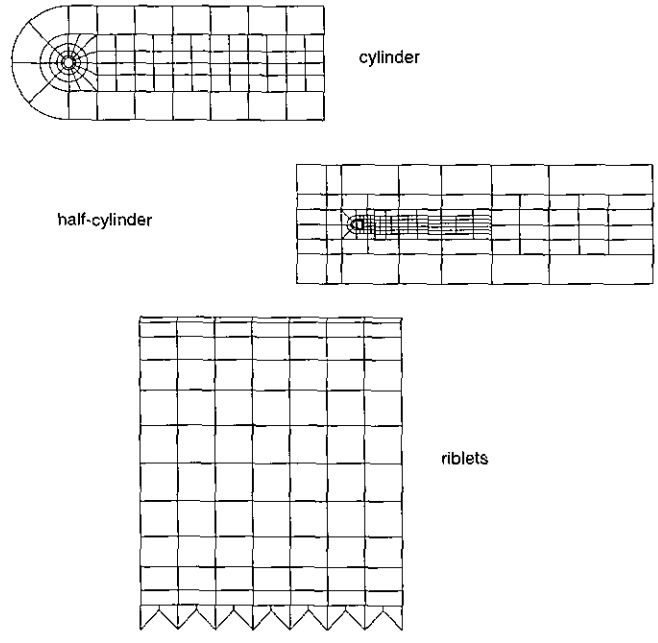

FIG. 15. Nonconforming meshes used to test the bandwidth optimization.

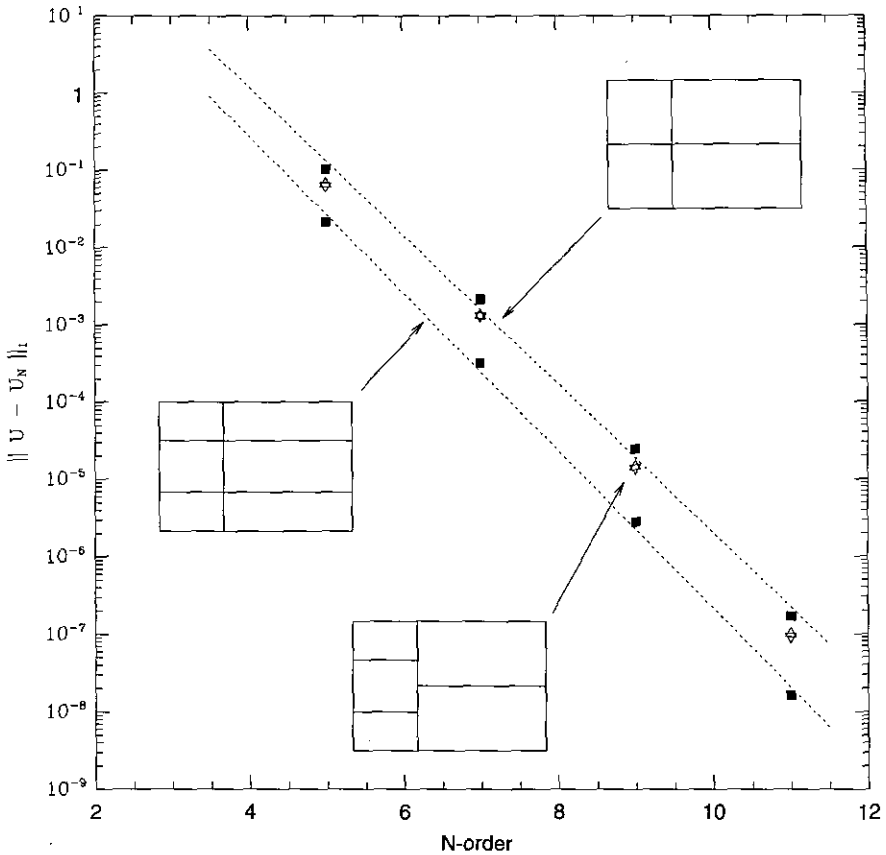

FIG. 16. Convergence of spectral element approximations to the Kovasznay problem in the H^* norm: ■ = conforming, △ = mortar patching, and ▽ = Zanolli iteration.

of problems have shown this trend is consistent and that in some cases the C^1 patching is slightly more accurate.

B. *Creeping Flow in a Wedge*

The next example is the solution to Stokes flow in a wedge, where the "lid" moves with a constant velocity ($u = 1$, $v = 0$). The solution is characterized by a series of Moffat's eddies [25], each one decaying geometrically in an infinite cascade toward the tip. For the wedge shown here ($2\alpha = 2 \arctan \frac{1}{4}$) the asymptotic ratios of successive eddy location and strength are $\ln r_n/r_{n+1} \doteq 0.6975$ and $\ln \omega_n/\omega_{n+1} \doteq 6.008$; Fig. 18 shows that away from the moving wall the simulations reproduce the cascade down to machine precision. Figure 17 shows the difficulty of representing a discontinuous function, in this case the u-component of velocity, with high-order polynomials. Oscillations induced by the discontinuity in the boundary data at the corner propagate into the domain but are essentially limited to the adjacent elements. This "limited influence" is a general feature of the C^0 interface condition and a good indication of why proper h-refinement is still an important ingredient of the spectral element method. In the mesh shown in the right of this figure, a selective refinement of the corner elements isolates the problem and effectively "resolves" the singularity. In this case the refinement is achieved by adding four elements and two patches to the mesh, and the patching is performed using mortar projection.

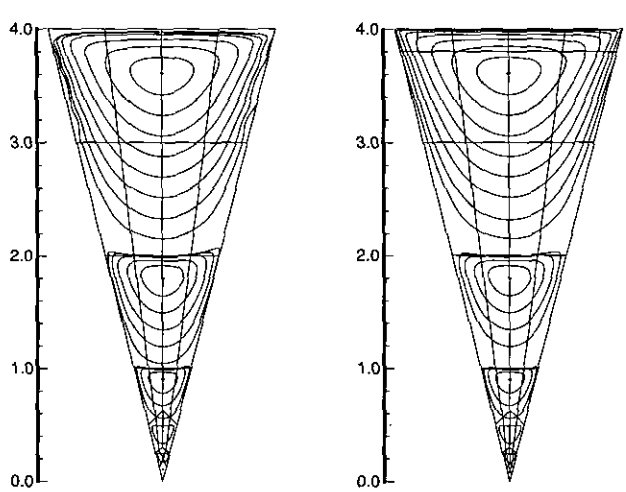

FIG. 17. Streamlines for creeping flow in a wedge: (left) conforming mesh with $K = 24$; (right) nonconforming mesh with $K = 34$. Both discretizations use a constant polynomial order of $N = 8$ in each element.

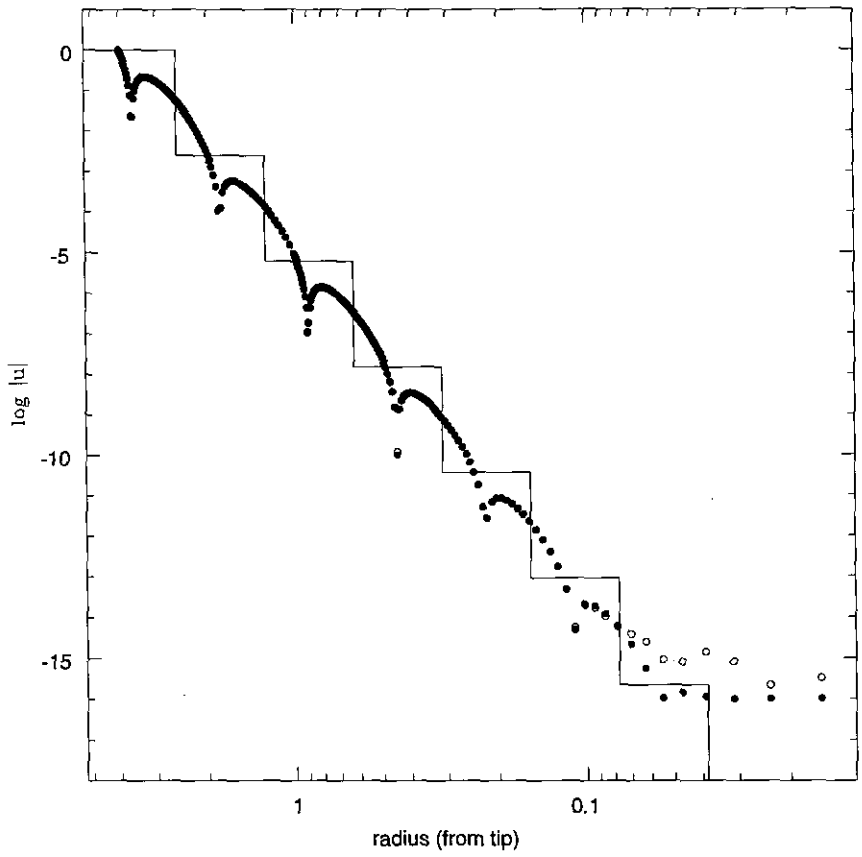

FIG. 18. Geometric decay of eddy strength for creeping flow in a wedge, indicated by the magnitude of u-velocity along the centerline: (——) asymptotic decay rate, ○ = initial conforming calculation, ● = solution after corner and tip refinement (radius decreases to the right).

C. *Wakes*

In the next example we include elements of these two problem areas by simulating the flow past a bluff body in the shape of a half-cylinder at Re = 250. The half-cylinder geometry is a canonical bluff body where the separation points are fixed. Boundary conditions along the edges of the computational domain are uniform flow ($u = 1$, $v = 0$) across the left, top, and bottom boundaries and outflow ($(\mathbf{n} \cdot \nabla)\,\mathbf{u} = 0$) across the right boundary. The conforming mesh uses $K = 276$ elements while the nonconforming mesh uses $K = 176$ elements, both shown in Fig. 19. Each simulation uses 49 collocation points per element ($N = 6$). We consider only the mortar patching for this example because of the complexity of the mesh.

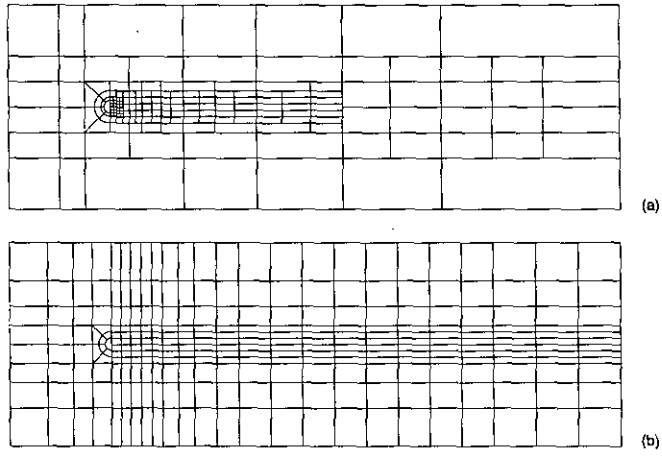

FIG. 19. (a) Nonconforming and (b) conforming meshes for the half-cylinder simulation.

The goal is to study the vortex formation region in the near wake of the body, shown in Fig. 20 as vorticity contours. The vorticity is obtained from the curl of the computed velocity field, and the noisy solution on the conforming mesh can be interpreted as a type of aliasing in physical space that arises because of small-scale features unresolved by the mesh. The singular corner presents an additional difficulty in modeling this flow. The nonconforming discretization of the domain isolates these features within a few elements close to the surface and resolves the fine scales present in the forming vortex. At the same time, it eliminates unnecessary elements away from the body where the solution is smooth but maintains far-field boundaries at the same distance.

This example is more typical of the problem size involved in "production runs" of unsteady flows. Table IV compares the CPU time and memory requirements for both discretizations, where (a) all the velocity and pressure systems are solved directly, (b) the velocity system is solved using the PCG solver, and (c) the velocity system is solved without the preconditioner. This table identifies the bandwidth of the stiffness matrix as a key element in the performance of the direct solver. With an arbitrary numbering scheme (*random*) the nonconforming discretization outperforms the conforming one because of the reduction in the number of elements. The slight improvement in the advection phase for the optimized conforming mesh reflects better memory performance during the direct stiffness summation procedure. Bandwidth reduction is more difficult for the nonconforming mesh because the element coupling extends beyond nearest neighbors, so the size of each optimized system is approximately the same. The increase in time required for the iterative solver (b) is consistent with the prediction of the condition number scaling given earlier and reflects a reduction in the smallest element size of roughly 4:1. This data supports the conclusion that a nonconforming discretization coupled with efficient solvers substantially improves the results of the simulation by giving better qualitative results for approximately the same computational cost and lower memory requirements.

4.2. Turbulent Flow Simulations

In this section we present simulations of two prototype turbulent flows that take advantage of the nonconforming discretizations we have developed. They represent both external (spatially developing) and internal (fully developed) turbulent flows. Each of these simulations was performed on an Intel Paragon XP/S, using a number of processors (8 to 64) equal to the number of Fourier modes in the discretization. Each processor solves the independent equations for the velocity and pressure on one Fourier mode, communicating with other processors in the mesh only when nonlinear interactions are computed (in physical space). Our largest simulations require 4–6 s per time step. Integration of the flow over a dynamically significant time interval requires at least 10^5 time steps. Next we describe how nonconforming meshes make these simulations practical.

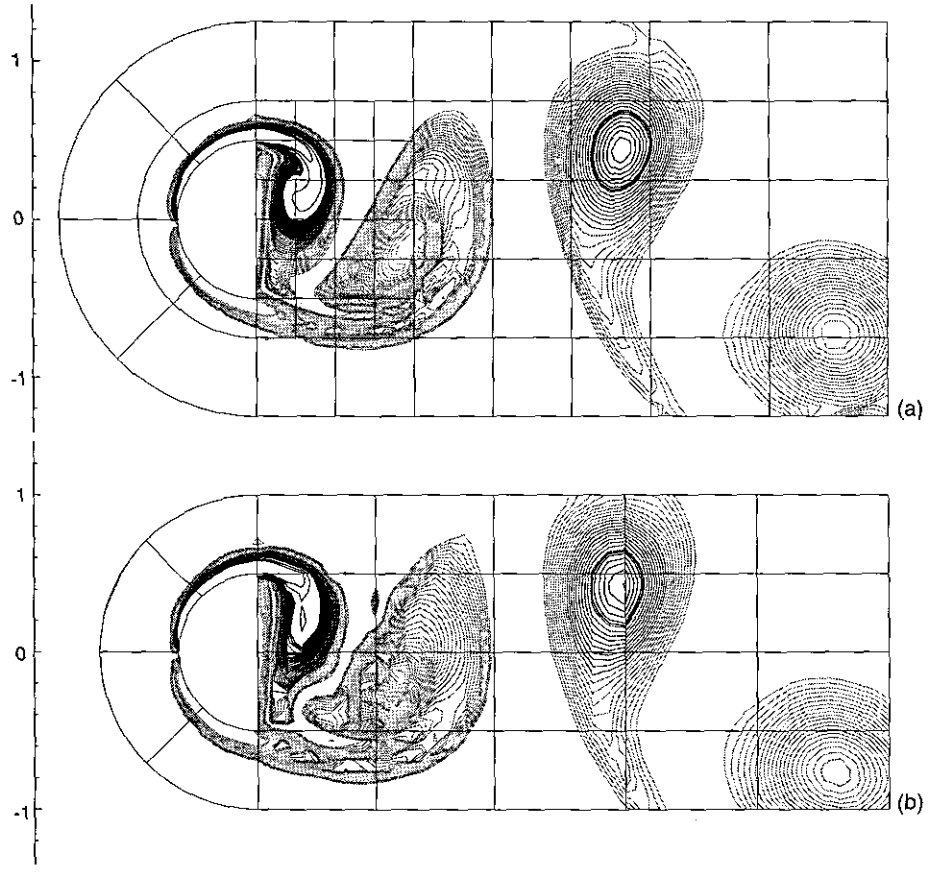

FIG. 20. Expanded view of the vortex formation region behind the half-cylinder.

TABLE IV

Comparison of CPU Time/Time Step (Seconds) and Memory Requirements (M-Words) for the Half-Cylinder Simulations: (a) All Direct Solvers; (b) Iterative Velocity/Direct Pressure; (c) All Iterative Solvers

	Solver	Advection	Pressure	Diffusion	Total	Memory
Nonconforming	a	0.15	1.31	2.44	3.90	6.15
Random	b			5.03	6.49	3.33
	c			298.84	300.30	3.33
Optimized	a	0.15	0.65	1.22	2.02	3.64
	b			5.03	5.83	1.97
	c			298.84	299.64	1.97
Conforming	a	0.25	4.25	5.89	10.39	11.09
Random	b			1.90	6.40	5.81
	c			118.49	122.99	5.81
Optimized	a	0.23	0.59	1.22	2.04	3.44
	b			1.88	2.70	1.73
	c			118.45	119.27	1.73

Note. Simulations were performed on a Silicon Graphics Indigo workstation with 12 M-words of memory and a 50 MHz IP20 processor.

A. *Wakes*

We present here simulation results for flow past a circular cylinder at Reynolds number $\mathrm{Re} \equiv u_\infty d/\nu$ of 1000. Laboratory experiments have shown that the wake of the circular cylinder becomes turbulent for $\mathrm{Re} \gtrsim 400$ [26]. Consider the resolution requirements for this problem. The flow near the cylinder is composed of a thin boundary layer and separating shear layers. In the near wake, these shear layers roll up into coherent structures that separate and move downstream as distorted tubes of vorticity. The flow outside the wake is described by a potential in which disturbances caused by the presence of the wake decay at a rate of $\sim 1/r^2$. Therefore, a large computational domain is needed to account for the interaction of the wake with the far field. An accurate flow simulation must resolve each of these regimes.

Our refinement strategy is as follows. Starting from a small initial domain (mesh M_1) with a span of $L_z = 2\pi$ and $M = 16$, we get an approximation to the cylinder wake. Next, we simultaneously refine the boundary layer and near-wake region while expanding the global domain size to remove blockage effects (meshes M_2 and M_3). Figure 21 shows a sequence of three meshes that illustrate this approach, and Table V lists the corresponding resolution and domain size parameters; all lengths are nondimensionalized by cylinder diameter d. The refinement criteria we employ are based on Strouhal number, base pressure coefficient, drag coefficient, and rms values of lift and drag fluctuations. The results of this refinement sequence are shown in Table VI, along with accepted experimental results [27, 28]. However, this essentially *two-dimensional* refinement does not produce a final result that agrees with experimental measurements, and further refinement seems to produce negligible effects. In each case, the refinement is an approximate bisection of the elements containing the boundary layer, although in the final mesh the refinement is much more extensive and includes the vortex formation region in the very near wake.

In the final stage of our refinement study, we consider only the effects of the *spanwise* length and corresponding resolution on our largest domain M_3. We have performed simulations with

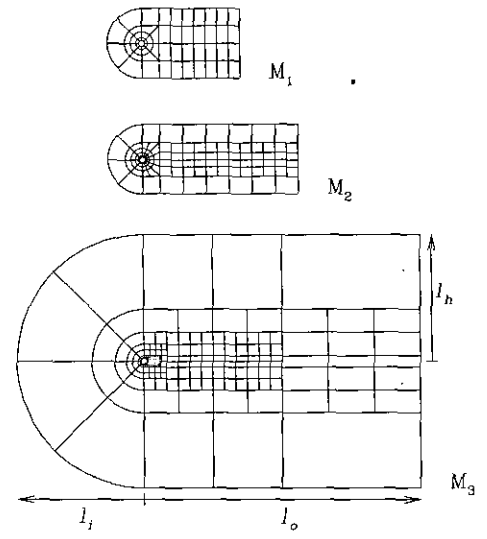

FIG. 21. Cross-sectional view of the meshes used in the turbulent wake simulations; corresponding size parameters are listed in Table V.

TABLE V

Domain Size and Cross-Sectional Resolution Parameters

	l_i	l_o	l_h	K	N	h_{min}	h_{avg}
Mesh 1	6	17	6	58	10	0.295	0.736
2	6	27	6	114	10	0.061	0.327
3	22	48	22	201	8	0.011	0.201

Note. h_{min} and h_{avg} are the minimum and average element sizes in the region $|r| \leq 2$.

TABLE VII

Spanwise Resolution Study for the Circular Cylinder Wake, Including the Corresponding Values from a 2D Simulation

	St	$-C_{pb}$	C_D	C_D'	C_L'
2D	0.237	1.6944	1.5144	0.1594	1.0494
$L_z = 2\pi$	0.213	1.2344	1.2814	0.0562	0.5394
4π	0.211	1.1225	1.1999	0.0491	0.2052
8π	0.21	1.16	1.23	0.04	0.2
16π	0.21	1.14	1.22	0.02	0.2
Experiment	0.21	0.8	1.0	?	?

$L_z = 2\pi$ to 16π, and for the circular cylinder wake we find the surprising result that a span of $L_z \geq 8\pi$ is required for even reasonable qualitative agreement with experimental measurements. Table VII shows that as we increase the spanwise length all global quantities converge, yet only the Strouhal number is in perfect agreement with the available experimental measurements. Global force measurements are all some 20% higher, implying that the wake of an infinitely long cylinder is different in some fundamental way from the wake observed in the experimental facilities. At the present time it is not clear if this is a numerical artifact or due to the inaccuracy of measurements given that quantities such as base suction and fluctuating lift are based on pressure measurements at very low levels.

The spanwise resolution is examined in Fig. 22, where we plot modal energy versus wavenumber for the larger spanwise lengths and different numbers of Fourier modes; we will refer to these results by the pair (L_z, M). The $(2\pi, 32)$ spectrum was omitted because it obviously gives incorrect behavior for the intermediate wavenumbers. Results for the domains $(4\pi, 32)$ and $(8\pi, 64)$ agree perfectly, except for the absence of the longest wavelength in the narrower domain. Also, computations with the higher spanwise resolution $(8\pi, 128)$ and the longer span $(16\pi, 128)$ agree for the intermediate wavenumbers. Because of the mapping used in the parallel implementation, the nonlinear terms are not dealiased, resulting in the cusp observed at the highest wavenumbers in each spectrum. Our refined calculations show that aliasing effects are confined to these high wavenumbers, and based on this we believe the small-scale behavior in the wake is fully resolved.

The results we present next correspond to mesh M_3 with $L_z = 8\pi$ and $M = 64$. In Fig. 23 we plot the fluctuating lift and drag on the cylinder as a function of time. This time series data exhibits a low-frequency variation on a time scale of ~10 shedding cycles. This same type of behavior has been observed in experimental measurements for other bluff bodies, but it was not clear if the force modulation was related to end effects or the intrinsic three-dimensionality of the wake [29].

In Fig. 24 we show a flow visualization of the normal velocity on the wake mid-plane ($y = 0$). This sequence of images covers approximately 16 shedding cycles and captures the full cycle of events responsible for the spanwise modulation of the flow. In the figure, flow is from left to right and the size of the visualized area is approximately 30 diameters downstream of the cylinder and the full 8π diameters across the span. In (a) the wake has developed what Williamson refers to as a "vortex dislocation," a region where the vortex shedding is out of phase along the span [30]. Later, in (d), a second dislocation occurs and restores the phase of vortex shedding along the span. Note that each of these solutions is still consistent with the periodic boundary conditions. This cycle repeats intermittently and occurs over a cell size of $\lambda \approx 4\pi$. This wavelength (which must be a function of Reynolds number) is too long to appear in our domains with $L_z < 8\pi$, and, as expected, the large variations in force amplitude are absent on the narrower domains. We [illegible]

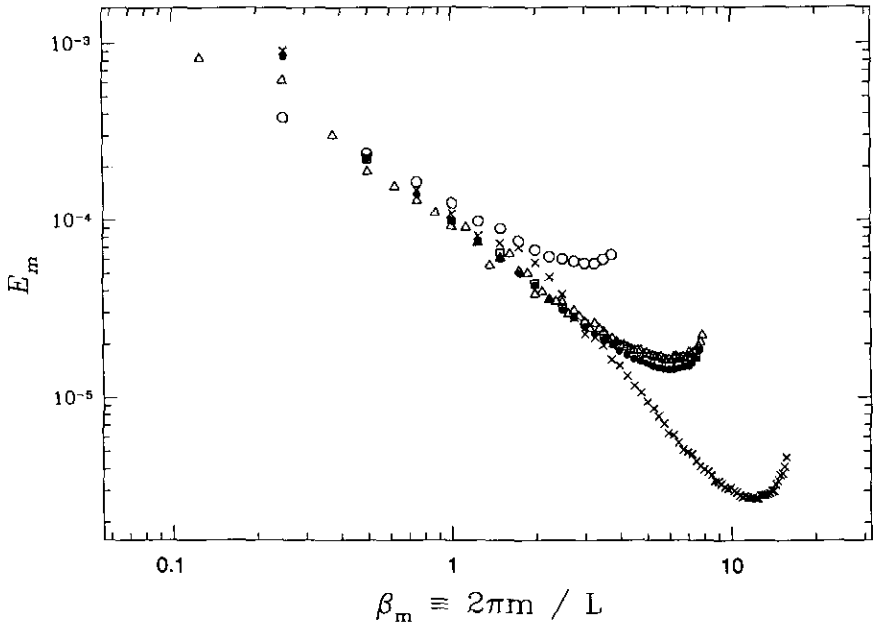

FIG. 22. Normalized spanwise energy spectrum (L_z, M): △ = (16π, 128), × = (8π, 128), ● = (8π, 64), ○ = (8π, 32), □ = (4π, 32).

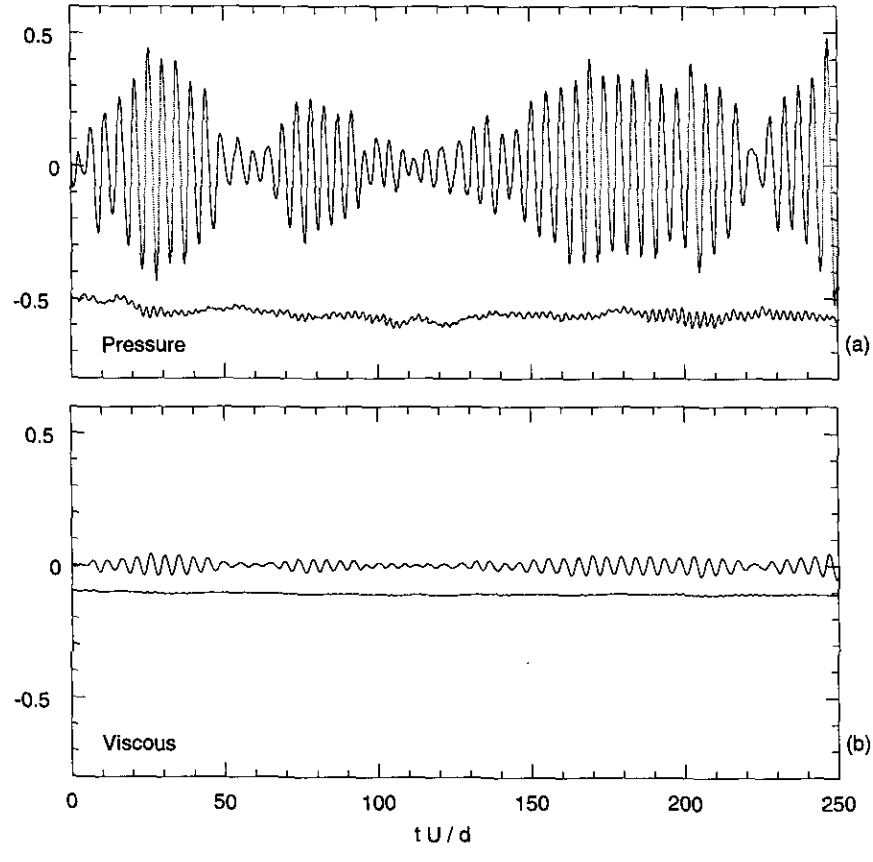

FIG. 23. Fluctuating lift and drag coefficients for flow past a circular cylinder at Re = 1000: (a) pressure forces and (b) viscous forces.

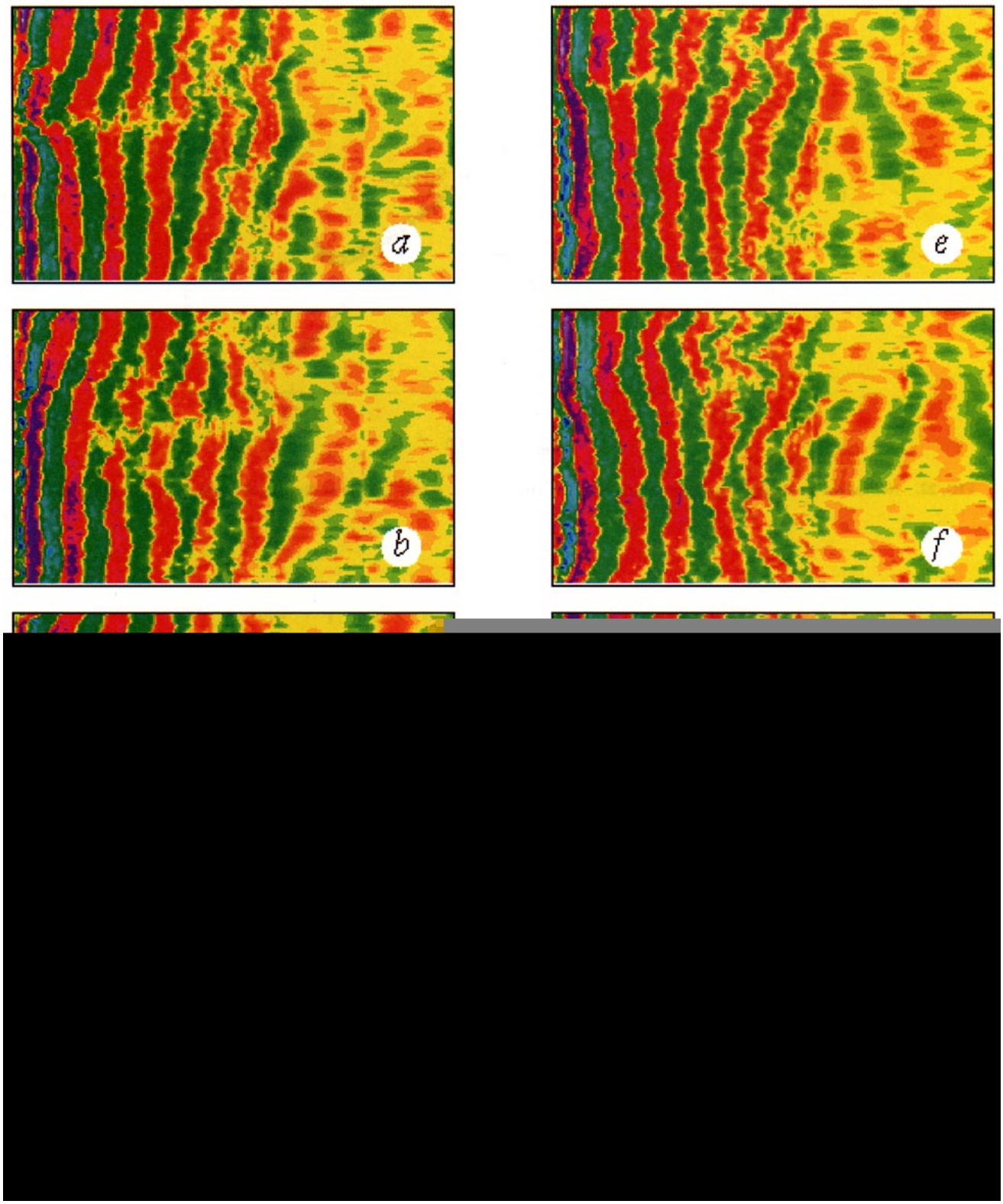

FIG. 24. Turbulent cylinder wake at Re = 1000: contours of normal velocity on the wake mid-plane mark the boundaries between successive vortices. The sequence of images (a)–(h) are spaced approximately two shedding cycles apart (10 time units).

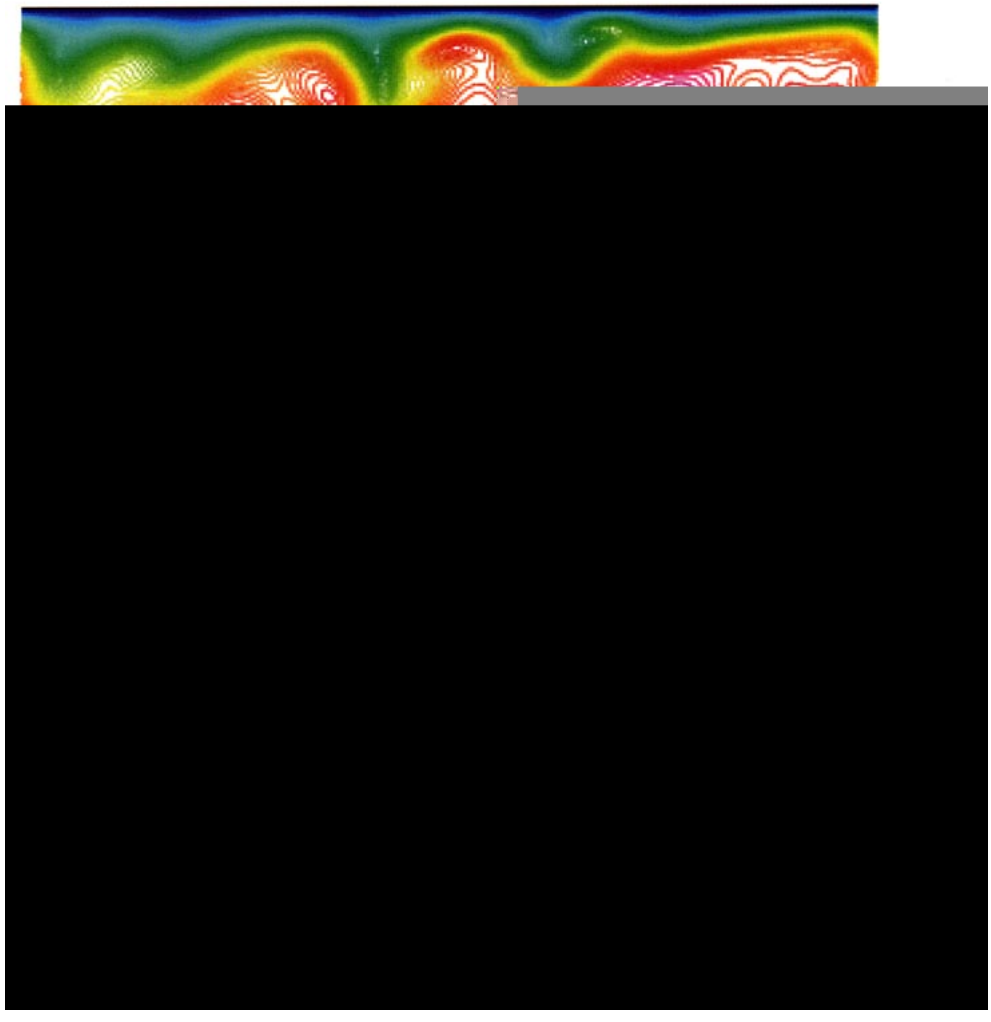

FIG. 27. Contours of streamwise velocity on a cross-flow plane.

sional and bears little resemblance to an idealized model of two-dimensional vortex shedding.

Our last result for the circular cylinder, in Fig. 25, shows the mean and corresponding rms streamwise and normal velocity fluctuations at three downstream locations, averaged over approximately 20 shedding cycles. The latter provides information on the spatial distribution of the turbulence intensity, and the location and variation of maximum turbulent stresses in the near wake. To show the relative mesh-independence of the velocity statistics, we have included results from mesh M_2. To our knowledge there are no similar experimental data to compare against at the present time, although current efforts point to that effect (M. Gharib and D. Williams, personal communications).

B. *Riblets*

In previous work we have investigated numerically the drag reduction obtained in turbulent flow over surfaces modified with riblets along the streamwise direction [32, 33]. In particular, the flow in a channel with its lower wall mounted with streamwise riblets and a smooth upper wall was simulated using a conforming spectral element–Fourier method. It was found that in the laminar regime there is no drag reduction, but in the transitional and turbulent regimes drag reduction of up to *six percent* can be achieved for the riblet-mounted wall in comparison with the smooth wall. These results also show a reduction of the peak rms fluctuation and peak Reynolds stress at the riblet valley wall.

In work under way, similar simulations of higher resolution are being carried out for different riblet configurations [34]. Accurate representation of the riblet geometry requires very high resolution around the riblet tip. This could be accomplished using conforming spectral elements; however, in that case the entire computational domain needs to be remeshed in order to satisfy the conforming discretization constraint. We can avoid rebuilding the mesh by using nonconforming discretizations as shown in Fig. 26.

Simulations were carried out both with conforming and nonconforming discretizations for the domain shown in Fig. 26 with dimensions $5.61H$ in the streamwise, $1.82H$ in the spanwise and $2H$ in the vertical direction (measured from the riblet midpoint to the smooth wall). Here, H is the channel half-height. There are seven riblets in the domain of size $0.13H$ height and $0.26H$ width. The simulations were performed at Reynolds number $\mathrm{Re} \equiv U_c H/\nu$ of 3280, where U_c is the mean centerline velocity. This corresponds to a Reynolds number based on wall units of

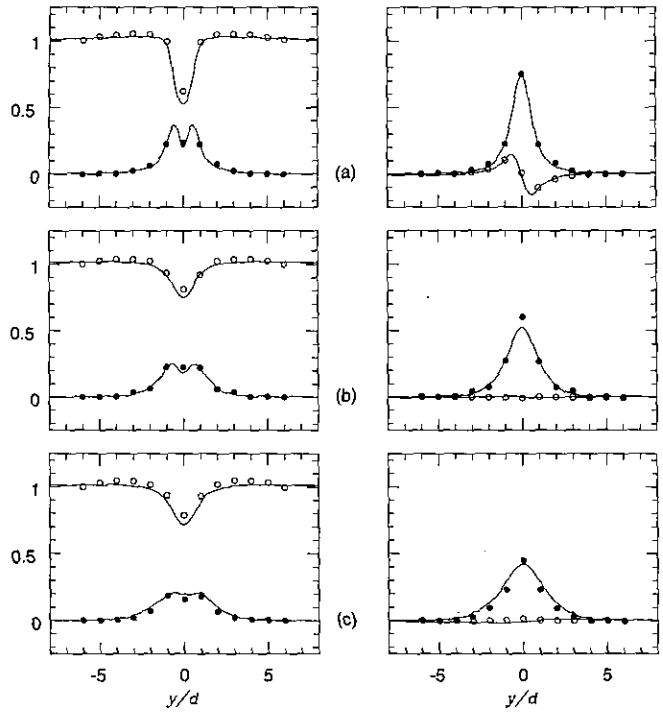

FIG. 25. Re = 1000 variation of (left) streamwise and (right) normal velocity fluctuations and mean profiles computed on mesh M_3: (a) $x/d = 2$, (b) $x/d = 5$, (c) $x/d = 10$; (●, ○) = corresponding values computed on M_2.

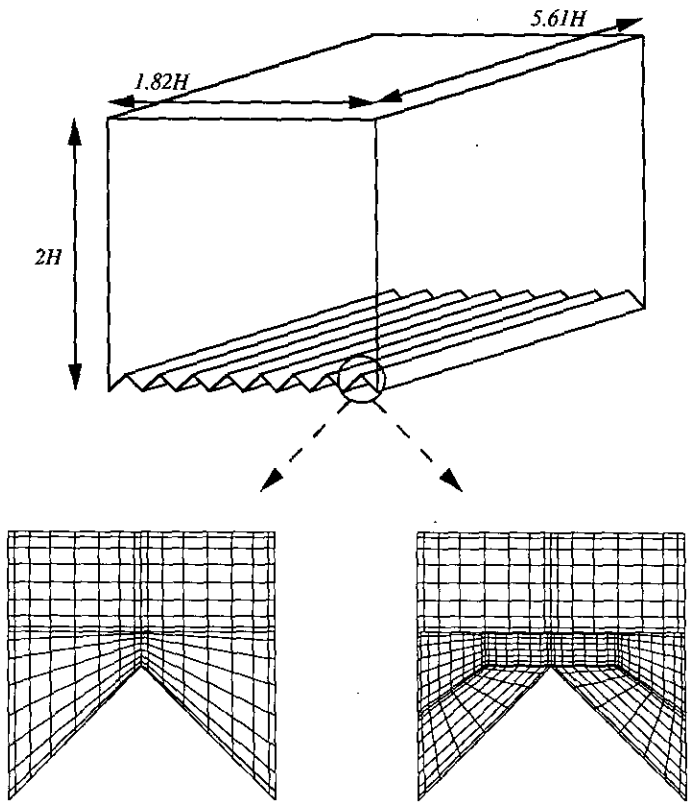

FIG. 26. Channel geometry for the riblet simulations and detail of the conforming and nonconforming meshes near the riblet tips.

$Re_* \approx 140$; therefore, the riblets are approximately 20 wall units (height) by 40 wall units (width). The refined (nonconforming) simulation started from the conforming one and was carried out for 12,000 time steps of $\Delta t = 0.005$. In order to have a direct comparison we have also continued the conforming discretization simulation for the same time interval.

Figure 27 is a flow visualization of the turbulent velocity field. It shows contours of the instantaneous streamwise velocity (normal to the page), indicating that a smooth and continuous velocity field is obtained in the near-wall region where the nonconforming discretization is placed. Typical statistical results of the two simulations are shown in Figs. 28 and 29 which show the mean velocity profile averaged over 12,000 time steps and the corresponding streamwise turbulence intensity normalized by the average velocity. The two simulations produce identical results. The riblet side corresponds to lower

for simulating incompressible turbulent flows with only one homogeneous direction. Using these methods, the computational domain can be discretized by employing globally unstructured meshes where we use local h-refinement to capture important features of the flow.

We concentrated on formulations for the elliptic kernel of the Navier–Stokes equations. The first of the two methods we presented is more general and can accomodate heterogeneous discretizations; interface continuity conditions are obtained via an iterative procedure (Zanolli patching). In the second method an additional variational equation establishes appropriate continuity conditions along nonconforming element boundaries. We have demonstrated that both patching schemes give essentially

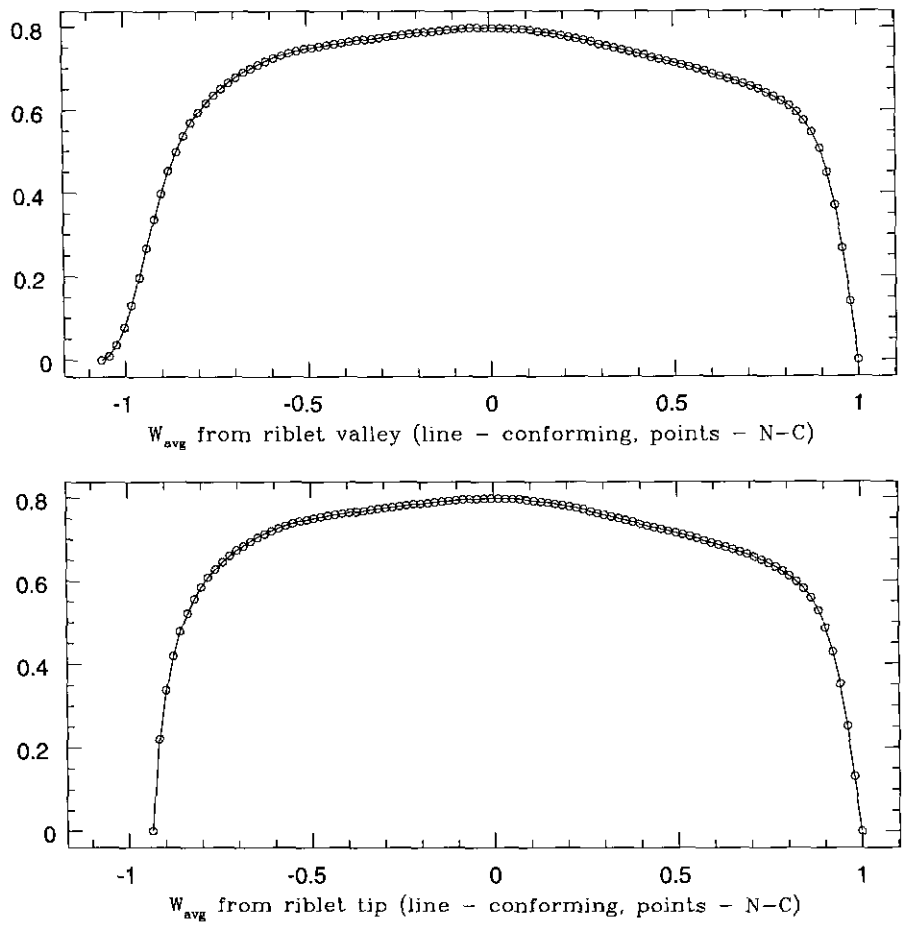

FIG. 28. A comparison of mean velocity profiles.

To pursue large-scale turbulent flow simulations, we have implemented this nonconforming spectral element–Fourier method for multiprocessor systems. Our hardware model is based on a parallel, distributed memory computer where a simple Fourier-mode to computer-node mapping results in good efficiency and scalability. A major advantage of this mapping is that it allows the use of direct solvers for the independent elliptic systems that must be solved on each processor. We also indicated how iterative solvers can be used to deal with very large problems that require more memory than is available on a single node. Because of the nature of these simulations, which require tracking the statistical behavior of the flow over long time periods, exclusive use of these iterative methods is not practical. We presented preliminary work on preconditioners that significantly improve the conditioning of the stiffness matrix, but none of which changed the scaling of computational work required for high resolution. A major focus of future work must be to improve the performance of iterative solvers, for example, by using preconditioners based on the Schur complement of the full stiffness matrix [35, 36].

In unstructured spectral discretizations, exponential convergence is preserved for smooth solutions of the Navier–Stokes equations. This was demonstrated numerically for the Kovasznay flow, an exact two-dimensional Navier–Stokes solution. The flexibility that nonconforming discretizations provide was demonstrated for flow in a wedge, where spurious oscillations due to discontinuous boundary conditions were accurately removed from the global solution. Finally, the new spectral element–Fourier method was applied to simulations of turbulent wakes and wall-bounded flows. We performed calculations on successively larger and more refined computational domains to examine the effect on the turbulent wake of the circular cylinder at Re = 1000. Numerical convergence was established and detailed mean flow data and turbulence statistics were presented for first time.

ACKNOWLEDGMENTS

We thank Catherine Crawford for preparing the riblet data and figures. Our turbulent flow simulations were performed in part using the Intel Paragon System operated by Caltech on behalf of the Concurrent Supercomputing Consortium. Access to this facility was provided by the California Institute of

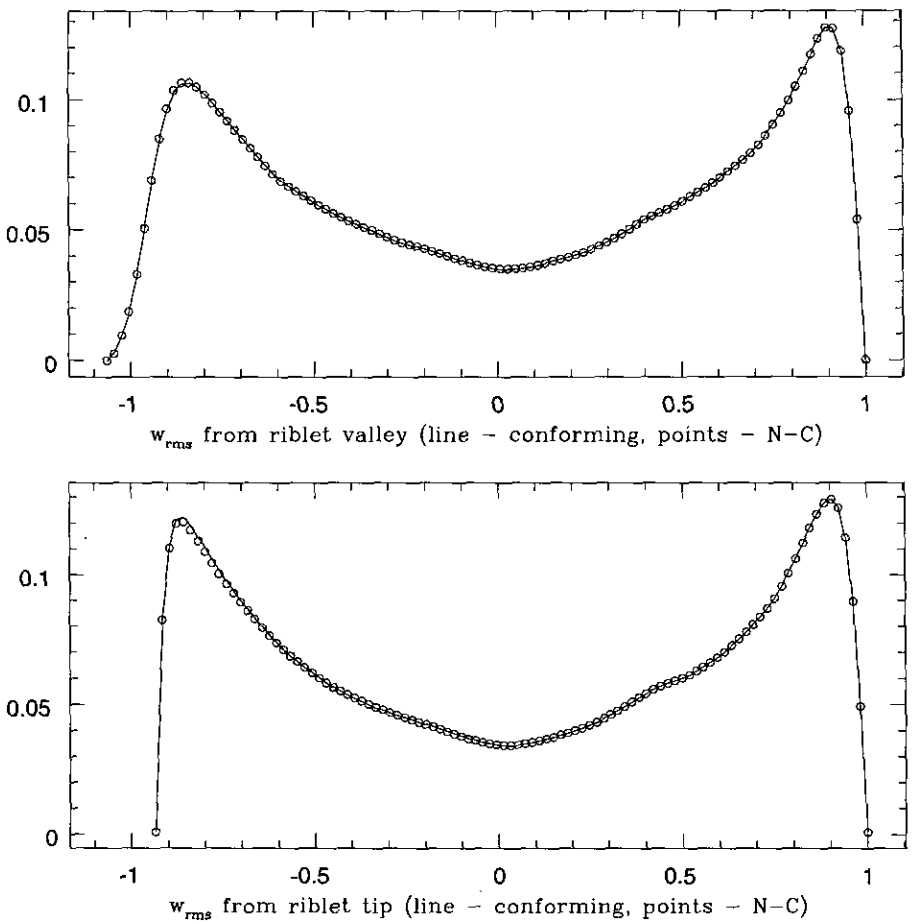

FIG. 29. A comparison of streamwise turbulence intensity.

Technology. Primary financial support for this work was provided by the Office of Naval Research through Grant N00014-90-J-1315. Supplementary support was provided by the AFOSR under Contract F49620-94-1-0313. The first author gratefully acknowledges further support from an AASERT fellowship paid for by Grant N00014-93-0791. Computing time and resources for the turbulent wake calculations reported here were made possible by the National Science Foundation under Grant CDA-9318145.

REFERENCES

1. G. E. Karniadakis and S. A. Orszag, *Phys. Today* **34** (1993).
2. H. O. Kreiss, technical report, University of Uppsala, Sweden, 1978.
3. D. Gottlieb and S. A. Orszag, *Numerical Analysis of Spectral Methods: Theory and Applications* (SIAM, Philadelphia, 1977).
4. A. T. Patera, *J. Comput. Phys.* **54,** 468 (1984).
5. G. E. Karniadakis, E. T. Bullister, and A. T. Patera, "A Spectral Element Method for Solution of Two- and Three-Dimensional Time Dependent Navier–Stokes Equations," in *Finite Element Methods for Nonlinear Problems* (Springer-Verlag, New York/Berlin, 1985).
6. G. E. Karniadakis, S. A. Orszag, E. M. Rønquist, and A. T. Patera, "Spectral Element and Lattice Gas Methods for Incompressible Fluid Dynamics," in *Incompressible Computational Fluid Dynamics: Trends and Advances*, edited by M. D. Gunzburger and R. A. Nicolaides, Chap. 8, p. 203 (Cambridge Univ. Press, Cambridge, UK, 1993).
7. C. Bernardi, Y. Maday, and A. T. Patera, "A New Nonconforming Approach to Domain Decomposition: The Mortar Element Method," in *Nonlinear Partial Differential Equations and Their Applications*, edited by H. Brezis and J. L. Lions (Pitman and Wiley, New York, 1992).
8. S. J. Sherwin and G. E. Karniadakis, *Comput. Methods Appl. Mech. Eng.* **123,** 189 (1995).
9. L. Demkowicz, J. T. Oden, W. Rachowicz, and O. Hardy, *Comput. Methods Appl. Mech. Eng.* **77,** 79 (1989).
10. A. G. Tomboulidas, S. A. Orszag, and G. E. Karniadakis, "Direct and Large Eddy Simulations of Axisymmetric Wakes," in *31st Aerospace Sciences Meeting & Exhibit, Reno, NV, 1993*; AIAA-93-0546.
11. G. E. Karniadakis, *Comput. Methods Appl. Mech. Eng.* **80,** 367 (1990).
12. T. A. Zang, *Appl. Numer. Math.* **7,** 27 (1991).
13. G. E. Karniadakis, M. Israeli, and S. A. Orszag, *J. Comput. Phys.* **97,** 414 (1991).
14. D. Funaro, A. Quarteroni, and P. Zanolli, *SIAM J. Numer. Anal.* **25,** 1213 (1988).
15. R. Henderson and G. E. Karniadakis, *J. Sci. Comput.* **6,** 79 (1991).
16. Oak Ridge National Laboratory, Oak Ridge, TN, *PVM 3 User's Guide and Reference Manual* (1993).
17. C. Mavriplis, Ph.D. thesis, Massachusetts Institute of Technology, 1989 (unpublished).

18. G. Anagnostou, Ph.D. thesis, Massachusetts Institute of Technology, 1991 (unpublished).
19. F. Ben Belgacem and Y. Maday, *Comput. Methods Appl. Mech. Eng.* **116,** 59 (1994).
20. E. Anderson *et al.*, *LAPACK Users' Guide* (SIAM, Philadelphia, 1992); software available via `ftp` from `netlib.att.com`.
21. R. Barrett *et al.*, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*; text available via `ftp` from `netlib.att.com`, 1994.
22. S. A. Orszag, *J. Comput. Phys.* **37** (1980).
23. M. O. Deville and E. H. Mund, *J. Comput. Phys.* **60,** 517 (1985).
24. L. I. G. Kovasznay, in *Proc. Cambridge Philos. Soc.* (1948).
25. H. K. Moffat, *J. Fluid Mech.* **18,** 1 (1964).
26. M. S. Bloor, *J. Fluid Mech.* **19,** 290 (1964).
27. M. M. Zdravkovich, *J. Wind Eng.* **33,** 53 (1990).
28. C. Norberg, *J. Fluid Mech.* **258,** 287 (1994).
29. A. Roshko, *J. Wind Eng.* **49,** 79 (1993).
30. C. H. K. Williamson, *J. Fluid Mech.* **243,** 393 (1992).
31. T. Dallard and F. K. Browand, *J. Fluid Mech.* **247,** 339 (1993).
32. D. Chu, R. D. Henderson, and G. E. Karniadakis, *Theor. Comput. Fluid Dyn.* **3,** 219 (1992).
33. D. C. Chu and G. E. Karniadakis, *J. Fluid Mech.* **250,** 1 (1993).
34. C. H. Crawford, Ph.D. thesis, Princeton University, 1995, in preparation.
35. J. Mandel, *Comput. Methods Appl. Mech. Eng.* **116,** 175 (1994).
36. W. Couzy and M. O. Deville, *J. Comput. Phys.* **116,** 135 (1995).