# A stabilized finite element method for computing turbulence

Kenneth E. Jansen

*Department of Mechanical Engineering, Aeronautical Engineering & Mechanics, Rensselaer Polytechnic Institute, Troy, NY 12180 USA*

**Abstract**

For many flows, Reynolds-averaged Navier-Stokes simulation (RANSS) techniques do not give an acceptable description of the flow. In these cases a more detailed simulation of the turbulence is required. One such detailed simulation technique, large-eddy simulation (LES) has matured to the point of application to complex flows. Historically, these simulations have been carried out with structured grids which suffer from two major difficulties: the extension to higher Reynolds numbers leads to an impractical number of grid points, and most real world flows are rather difficult to represent geometrically with structured grids. Unstructured-grid methods offer a release from both of these constraints. Herein, we describe the development and validation of a massively parallel, stabilized finite element method for LES.

## 1 Introduction

Computation of turbulence can be carried out in three distinct ways: direct numerical simulation (DNS), where the computational method resolves all of the turbulent motions (from the largest scale down to the scale where motion is converted to heat via viscous dissipation), LES, where the large eddies carrying most of the energy are resolved by the computational method (leaving the small or subgrid-scale (SGS) eddies to be modeled), and RANSS, where no attempt is made to resolve any of the turbulent motions (rather, the net effect of these motions upon the mean flow is modeled). DNS can be expected to remain out of reach of most of the engineering problems in the next couple of decades. RANSS, though computationally efficient for some steady flows, has been disappointing for unsteady and/or complex flows. Therefore, LES has been identified as the most promising simulation tool for the next two decades of turbulence simulation.

Over the past three decades, a significant amount of research has been devoted to LES. The overwhelming majority of this research has been carried out with spectral or structured grid finite difference methods. While much has been learned about the basic physics of turbulence using these approaches, they suffer from two significant problems: difficulty in application to arbitrarily complex geometries, and a rapid increase in the number of degrees of freedom necessary to describe flows at typical engineering Reynolds numbers. Recently, LES has been extended to finite element methods on unstructured grids [1] with near perfect parallel scaling [2]. This extension not only facilitates simulation of flows within or around complex geometries found in engineering applications, but also allows great reductions in computational effort through the ability of unstructured grids to adapt locally to fine scale structures in one flow region while remaining coarse in regions where the structures are large. Already, these methods have realized a 27 fold point reduction over an equivalent-resolution structured grid ([3],[4]).

Within this paper we will describe the LES equations, the finite element formulation used to solve them, and the salient features leading to efficient parallel implementation. We also give a description of the extension of the dynamic sub-grid model to unstructured-grid simulations. After a cursory description of mesh generation techniques, results of past and current simulations will be provided to illustrate the efficiency and the accuracy of the approach. Results will be presented both from simple validation problems and from the very challenging simulation of flow over an airfoil.

## 2 The finite element formulation

There are many finite element and finite volume formulations available for application with unstructured grids. However, few have had their accuracy and stability as carefully scrutinized as what have become known as stabilized finite element methods. Two important members of this family, Galerkin/Least-Squares (GLS) and Streamline Upwind Petrov Galerkin (SUPG), have been proven stable and higher order accurate (converging at the optimal rate for a given function space) for flows ranging from inviscid to viscous dominated. The early analyses were done in the context of multi-dimensional linear advective-diffusive systems (see Hughes et al.[5] and Hughes et al.[6]) and later followed by an analysis of the frozen coefficient Navier-Stokes equations (see Franca and Hughes [7]). The interested reader is encouraged to look to these references for detail as the space of this paper allows only a summary of the formulation below.

Consider the compressible Navier-Stokes equations (complete with continuity and total energy equations) written in filtered form after the application of a subgrid-scale model (see Moin et al.[8], Moin & Jimenéz [9], Germano et al.[10] or section 4 for details ).

$$\boldsymbol{U}_{,t} + \boldsymbol{F}^{\mathrm{adv}}_{i,i} - \boldsymbol{F}^{\mathrm{diff}}_{i,i} = \boldsymbol{\mathcal{S}} \tag{1}$$

where

$$\boldsymbol{U} = \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \end{Bmatrix} = \overline{\rho} \begin{Bmatrix} 1 \\ \tilde{u}_1 \\ \tilde{u}_2 \\ \tilde{u}_3 \\ \tilde{e}_{\text{tot}} \end{Bmatrix}, \qquad \boldsymbol{F}_i^{\text{adv}} = \tilde{u}_i \boldsymbol{U} + \overline{p} \begin{Bmatrix} 0 \\ \delta_{1i} \\ \delta_{2i} \\ \delta_{3i} \\ \tilde{u}_i \end{Bmatrix}, \qquad \boldsymbol{F}_i^{\text{diff}} = \begin{Bmatrix} 0 \\ \tau_{1i} \\ \tau_{2i} \\ \tau_{3i} \\ \tau_{ij}\tilde{u}_j - q_i \end{Bmatrix} \tag{2}$$

and

$$\tau_{ij} = 2(\mu + \mu_T)(S_{ij}(\tilde{\boldsymbol{u}}) - \frac{1}{3} S_{kk}(\tilde{\boldsymbol{u}})\delta_{ij}) \quad , \quad S_{ij}(\tilde{\boldsymbol{u}}) = \frac{\tilde{u}_{i,j} + \tilde{u}_{j,i}}{2} \tag{3}$$

$$q_i = -(\kappa + \kappa_T)\tilde{T}_{,i} \quad , \quad \kappa_T = c_p \frac{\mu_T}{Pr_T} \quad , \quad \tilde{e}_{\text{tot}} = \tilde{e} + \frac{\tilde{u}_i \tilde{u}_i}{2} \quad , \quad \tilde{e} = c_v \tilde{T} \tag{4}$$

Where we use the overbar to denote an unweighted filter and a tilde to denote a density weighted filter. The filtered variables are: the velocity $\tilde{u}_i$, the pressure $\overline{p}$, the density $\overline{\rho}$, the temperature $\tilde{T}$ and the total energy $\tilde{e}_{\text{tot}}$. The constitutive laws relate the stress, $\tau_{ij}$, to the deviatoric portion of the strain, $S^d_{ij} = S_{ij} - \frac{1}{3} S_{kk}\delta_{ij}$, through a molecular, $\mu$, plus turbulent viscosity, $\mu_T$. Similarly, the heat flux, $q_i$, is proportional to the gradient of temperature with the proportionality constant given by the addition of a molecular conductivity, $\kappa$, and a turbulent conductivity, $\kappa_T$ which is assumed proportional to the turbulent viscosity as described above. While the formulation is not limited to an ideal gas, $\overline{p} = \overline{\rho} R \tilde{T}$, and constant specific heats at constant pressure, $c_p$, and at constant volume, $c_v$, that is the model used in the calculations shown in this paper. Furthermore, since all the calculations shown in this paper are at low Mach number where temperature variation is low we have also assumed a constant molecular viscosity and constant conductivity through a constant Prandtl number, though, this again is not a necessary simplification. Finally $\boldsymbol{\mathcal{S}}$ is a body force (or source) term.

For the specification of the methods that follow, it is helpful to define the quasi-linear operator (with respect to some yet to be determined variable vector $\boldsymbol{Y}$) related to (1) as

$$\boldsymbol{\mathcal{L}} \equiv \boldsymbol{A}_0 \frac{\partial}{\partial t} + \boldsymbol{A}_i \frac{\partial}{\partial x_i} - \frac{\partial}{\partial x_i}(\boldsymbol{K}_{ij} \frac{\partial}{\partial x_j}) \tag{5}$$

from which $\boldsymbol{\mathcal{L}}$ can be naturally decomposed into time, advective, and diffusive portions

$$\boldsymbol{\mathcal{L}} = \boldsymbol{\mathcal{L}}_{\text{t}} + \boldsymbol{\mathcal{L}}_{\text{adv}} + \boldsymbol{\mathcal{L}}_{\text{diff}}. \tag{6}$$

Here $\boldsymbol{A}_i = \boldsymbol{F}^{\text{adv}}_{i,\boldsymbol{Y}}$ is the $i^{th}$ Euler Jacobian matrix, $\boldsymbol{K}_{ij}$ is the diffusivity matrix, defined such that $\boldsymbol{K}_{ij}\boldsymbol{Y}_{,j} = \boldsymbol{F}^{\text{diff}}_i$, and $\boldsymbol{A}_0 = \boldsymbol{U}_{,\boldsymbol{Y}}$ is the change of variables metric. For a complete description of

$\boldsymbol{A}_0$,$\boldsymbol{A}_i$ and $\boldsymbol{K}_{ij}$, the reader is referred to [11]. Using this, we can write (1) as simply $\boldsymbol{\mathcal{L}}\boldsymbol{Y} = \boldsymbol{\mathcal{S}}$.

To proceed with the finite element discretization of the Navier-Stokes equations (1), we must make precise the approximation spaces with which we are dealing. First, let $\bar{\Omega} \subset \boldsymbol{R}^N$ represent the closure of the physical spatial domain (i.e. $\Omega \cup \Gamma$ where $\Gamma$ is the boundary) in $N$ dimensions, only $N = 3$ is considered here. In addition, $H^1(\Omega)$ represents the usual Sobolev space of functions with square-integrable values and derivatives on $\Omega$.

Next, $\Omega$ is discretized into $n_{el}$ finite elements, $\Omega^e$. With this, we can define the trial solution space for the semi-discrete formulations as

$$\boldsymbol{\mathcal{V}}_h = \{\boldsymbol{v} | \boldsymbol{v}(\cdot,t) \in H^1(\Omega)^m, t \in [0,T], \boldsymbol{v}|_{x\in\Omega^e} \in P_k(\Omega^e)^m, \boldsymbol{v}(\cdot,t) = \boldsymbol{g} \text{ on } \Gamma_g\}, \tag{7}$$

and the weighting function space

$$\boldsymbol{\mathcal{W}}_h = \{\boldsymbol{w} | \boldsymbol{w}(\cdot,t) \in H^1(\Omega)^m, t \in [0,T], \boldsymbol{w}|_{x\in\Omega^e} \in P_k(\Omega^e)^m, \boldsymbol{w}(\cdot,t) = \boldsymbol{0} \text{ on } \Gamma_g\}, \tag{8}$$

where $P_k(\Omega^e)$, is the space of all polynomials defined on $\Omega^e$, complete to order $k \geq 1$, and $m$ is the number of degrees of freedom ($m = 5$).

To derive the weak form of (1), the entire equation is dotted from the left by a vector of weight functions, $\boldsymbol{W} \in \boldsymbol{\mathcal{W}}_h$, and integrated over the spatial domain. Integration by parts is then performed to move the spatial derivatives onto the weight functions (reducing the continuity requirements). This process leads to the integral equation (often referred to as the weak form): find $\boldsymbol{Y} \in \boldsymbol{\mathcal{V}}_h$ such that

$$\begin{aligned} 0 = &\int_{\Omega} \left( \boldsymbol{W} \cdot \boldsymbol{U}_{,t} - \boldsymbol{W}_{,i} \cdot \boldsymbol{F}_i^{\text{adv}} + \boldsymbol{W}_{,i} \cdot \boldsymbol{F}_i^{\text{diff}} \right) d\Omega - \int_{\Gamma} \boldsymbol{W} \cdot \left( -\boldsymbol{F}_i^{\text{adv}} + \boldsymbol{F}_i^{\text{diff}} \right) \, n_i \; d\Gamma \\ &+ \sum_{e=1}^{n_{el}} \int_{\Omega^e} \boldsymbol{\mathcal{L}}^T \boldsymbol{W} \cdot \boldsymbol{\tau} \left( \boldsymbol{\mathcal{L}}\boldsymbol{Y} - \boldsymbol{\mathcal{S}} \right) d\Omega \end{aligned} \tag{9}$$

The first line of (9) contains the Galerkin approximation (interior and boundary) and the second line contains the least-squares stabilization. SUPG stabilization is obtained by replacing $\boldsymbol{\mathcal{L}}^T$ by $\boldsymbol{\mathcal{L}}^T_{\text{adv}}$. The stabilization matrix $\boldsymbol{\tau}$ is an important ingredient in these methods and is well documented in Shakib [12] and in Franca and Frey [13]. Note that we have chosen to find $\boldsymbol{Y}$ instead of $\boldsymbol{U}$. As discussed in Hauke and Hughes [14], $\boldsymbol{U}$ is often not the best choice of solution variables, particularly when the flow is nearly incompressible. For the calculations performed herein, the SUPG stabilized method was applied with linearly interpolated pressure-primitive variables viz.

$$\boldsymbol{Y} = \left\{ \begin{array}{c} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \end{array} \right\} = \left\{ \begin{array}{c} \overline{p} \\ \tilde{u}_1 \\ \tilde{u}_2 \\ \tilde{u}_3 \\ \tilde{T} \end{array} \right\} \tag{10}$$

By inspecting (2)-(4) it is clear that all quantities appearing in (9) may be easily calculated from (10).

To develop a numerical method, the weight functions ($\boldsymbol{W}$), the solution variable ($\boldsymbol{Y}$), and it's time derivative ($\boldsymbol{Y}_{,t}$) are expanded in terms of basis functions (typically piecewise polynomials; all calculations described herein were performed with linear basis). The integrals (9) are then evaluated using Gauss quadrature resulting in a system of non-linear ordinary differential equations which can be written as

$$\boldsymbol{M}\check{\boldsymbol{Y}}_{,t} = \boldsymbol{R}(\check{\boldsymbol{Y}}) \tag{11}$$

where the check is added to make clear that $\check{\boldsymbol{Y}}$ is the vector of solution values at discrete points (interpolated through the space by the basis functions) and $\check{\boldsymbol{Y}}_{,t}$ are the time derivative values at the same points. An implicit, second order accurate family of time integrators has been developed for application to this problem. This new time integrator is an extension of Chung and Hulbert's generalized alpha method [15] to our first order system. The family is characterized by one free parameter, the amplification factor as the time step over the period of the frequency of interest tends to infinity, $\rho_\infty$, which may be chosen to have a value between 0 (Gears Method) to 1 (Midpoint Rule). Full documentation of Chung and Hulbert's method extended to the Navier-Stokes equations is not possible in the confines of this paper but should appear in the literature shortly. The method results in a non-linear matrix problem that is solved in a predictor-corrector format yielding successive linear problems. Each linear problem is solved using the Matrix-Free Generalized Minimal RESidual (MF-GMRES) solution technique with a block diagonal preconditioner developed by Johan et al.[16]. Convergence of the non-linear problem is confirmed before moving to the next time step. We have observed that there is little difference in the turbulent statistics between solving the equations with two iterations (2.5-3.0 order of magnitude reduction) and three iterations (3.0-4.0 orders of magnitude reduction). There is of course a difference in the instantaneous signal after a long time integration as will always be the case when dealing with turbulence. It is interesting to note that even roundoff errors will eventually lead to a divergence of an instantaneous signal regardless of the level of solution. This is initially alarming until we remember that it is the turbulence statistics that we are most interested in predicting.

Finite element methods are relatively new to computational fluid dynamics. Most applications thus far have been to steady, laminar or Reynolds-averaged flows. Until recently their application to

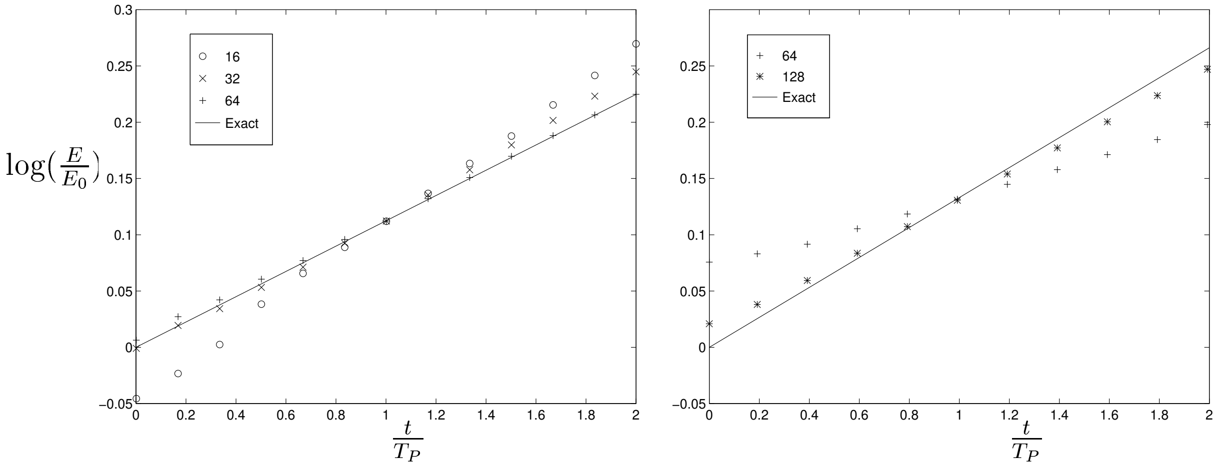


Fig. 1. Shifted growth of energy of small disturbance energy (to match theory after one period of the eigenmode disturbance) of the finite element method (left) and the central-difference finite difference method (right). Legend shows the number of elements used in the $y$ direction.

unsteady turbulent flows was deemed too expensive and not sufficiently accurate. This assessment is puzzling given that the methods can be proven higher order accurate and that the unstructured grid can lead to point reductions which more than offset the additional work associated with the method. In fact the inclusion of a consistent mass matrix in (11) leads to a super-convergent phase accuracy in regions where the mesh is uniform or slowly varying. This super-convergence is the result of the rational approximation (Padé like) and can be illustrated by studying the growth of a small disturbance with the shape of the most unstable eigenmode of parallel channel flow. Here we repeat the flow conditions studied by Malik et al.[17] ($Re = 7500, \alpha_r = 1$). The growth of the disturbance energy within the Navier-Stokes code (no subgrid-scale model) can be compared to linear stability theory to test the accuracy of the numerical method.

In Figure 1 we compare the growth rates of our formulation to that obtained by central differences on a staggered grid (the workhorse of the LES community). The central difference results come from a well tested code but are at a slightly higher Reynolds number ($Re = 8000$) and are thus compared to the exact growth rate at that Reynolds number. In both methods 32 cells were used in $x$ direction while the $y$ resolution was varied using a fixed stretching function and a variable number of points. In both methods there is an initial transient which has been removed by shifting the curves such that they are correct at one period ($T_P$) of the flow disturbance. Finally, the convergence rates can be seen in Figure 2. From these figures one can clearly see the superior accuracy of the finite element formulation both in terms of rate (slope) and in terms of the constant. It is worth pointing out here that the author has used the best central-difference staggered grid results available ([17] publish much worse results with decay on grids of $N_y = 64!$). These figures clearly challenge the notion that kinetic energy conservation (a property of central difference schemes not enjoyed by stabilized finite element methods) is paramount since, at least in this problem, accurate interaction of the disturbance with the mean flow is more critical than the very small additional dissipation generated by the stabilization operator.

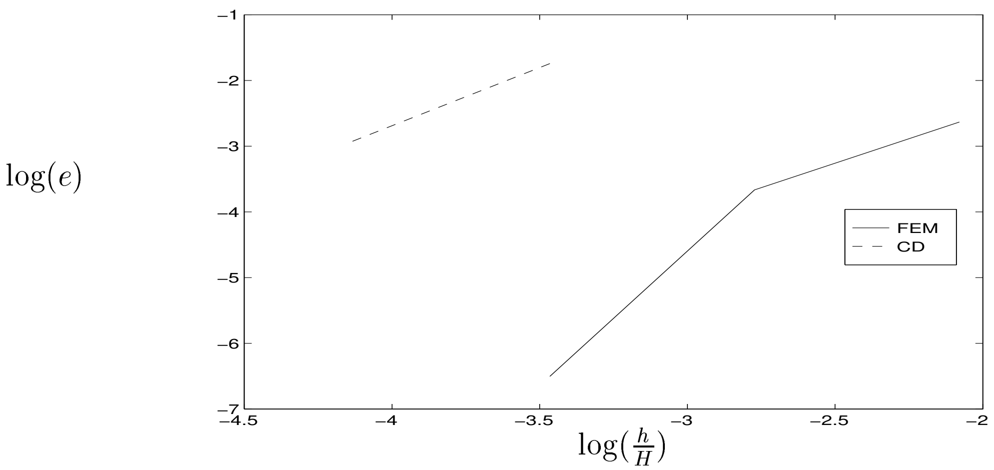


Fig. 2. Error in the second period (found by shifting the calculated energy to match solution from linear theory at one period of the disturbance and then comparing the difference after two periods) as a function of the average $y$ spacing, $h$, over the channel height, $H$.

## 3 Parallel implementations

The original implementation of this approach was on a Thinking Machines CM5 following the so-called data parallel approach. Finite element methods are amenable to this approach since the bulk of the computational effort lies in evaluating local integrals over each element domain. Parallelization of these operations is trivial after a gather of the data from the global nodes to an element based data structure. After the element integrations are performed in parallel, the results (residuals of (11)) are then scattered (assembled) back to the nodes where the equations are solved.

On the CM5 these gather/scatter operations are performed using CMSSL library procedures requiring no preprocessing of the data. This made application to this platform quite easy. Unfortunately, other parallel platforms do not have libraries of this type. For this reason a second version of the code was developed, employing the Message Passing Interface (MPI) library to do the gather and scatter operations. This version of the code requires that the problem be pre-processed to break the computational domain into $n$ pieces where $n$ is the number of processors, on which, the problem will be run. This preprocessing step is non-trivial for unstructured grids since load balance (equal distribution of the elements onto each processor) and minimal communication are necessary. Recently we have accomplished this preprocessing procedure and have begun testing our MPI version of the code on a variety of platforms (IBM-SP2, SGI-Origin, SUN-Sparc and Cray J90). The algorithm has shown perfect scalability (see Bastin [18]) on moderate sized problems (typically the bigger the better) due to the low ratio of communication to calculation that is inherent in finite element methods. The MPI version of the code also performs the communication much more efficiently than the CMSSL version of the code. On the CM5 communication required 20-25% of the total CPU time. With the MPI code this percentage is reduced to 3-5%. While this improvement is dramatic, both methods still use the majority of the CPU time computing, making communications improvement have little impact. Greater efficiency gains require a reduction in the computational effort, even if it comes at some loss in communication efficiency.

## 4 Dynamic model development

In section 2, the modeled form of the equations was given. These equations were closed by an expression for the subgrid-scale viscosity, $\mu_T$, which still must be given. To give a proper development, we retreat back to the compressible Navier-Stokes equations. After application of a filter, denoted by the overbar symbol, which is assumed to remove all the scales smaller than $\overline{\Delta}$ (usually defined to be the grid spacing) we obtain the the un-modeled large-eddy simulation equations. The filter applied to the convection term leads to a closure problem or what is known as the subgrid-scale stress, $t_{ij} = \overline{\rho u_i u_j} - \overline{\rho}\tilde{u}_i\tilde{u}_j = \overline{\rho}\widetilde{u_i u_j} - \overline{\rho}\tilde{u}_i\tilde{u}_j$, whose deviatoric portion, $t^d_{ij}$, is usually modeled to be proportional to the deviatoric portion of the mean strain-rate tensor $S^d_{ij}(\tilde{\boldsymbol{u}})$ (defined in (3)) viz.

$$t^d_{ij} = -2\mu_T S^d_{ij}(\tilde{\boldsymbol{u}}) \tag{12}$$

where the eddy viscosity, $\mu_T$, is usually given by the Smagorinsky model [19],

$$\mu_T = C\overline{\Delta}^2\overline{\rho}|S|, \qquad |S| = \sqrt{2S_{ij}S_{ij}} \tag{13}$$

where $C$ is the so-called Smagorinsky "constant".

This "constant" proved very elusive to the researchers of the seventies and the eighties. After several studies it seemed clear that this "constant" in fact varied from flow to flow and, worse yet, seemed to be a function of position within even the simplest flows (especially in near wall regions). In the early nineties a dynamic procedure for determining the subgrid-scale viscosity was developed by Germano et al.[10]. The key observation here was that if one considered the solution on a second grid which was coarser than the first (usually by a factor of 2 and denoted by the symbol $\widehat{\overline{\Delta}}$), insight might be gained into this spatially varying Smagorinsky "constant". On this coarser grid one could again define a subgrid-scale stress

$$T_{ij} = \widehat{\overline{\rho u_i u_j}} - \widehat{\overline{\rho}}\widetilde{\widehat{u}}_i\widetilde{\widehat{u}}_j \tag{14}$$

When this coarser grid, subgrid-scale stress was again modeled by Smagorinsky's model we obtain

$$T^d_{ij} = -2C\widehat{\overline{\rho}}\widehat{\overline{\Delta}}^2|\widehat{S}|\widehat{S}^d_{ij} \tag{15}$$

In reality the procedure avoids the calculation on the coarser grid by filtering the solution obtained on the computational grid. The filtering process is carried out through the convolution of any function, $f$ with a filtering kernel $G$ viz.

$$\hat{f}(\boldsymbol{x}) = \frac{\int_\Omega G(\boldsymbol{x}, \boldsymbol{x}')f(\boldsymbol{x}')d\boldsymbol{x}'}{\int_\Omega G(\boldsymbol{x}, \boldsymbol{x}')d\boldsymbol{x}'} \tag{16}$$

The fruit of this labor was first noted by Germano et al.[10]

$$\begin{aligned} L_{ij} =& T_{ij} - \widehat{t_{ij}} && (17)\\ =& \widehat{\overline{\rho u_i u_j}} - \widehat{\overline{\rho}}\widehat{\tilde{u}_i}\widehat{\tilde{u}_j} - \widehat{\overline{\rho u_i u_j}} + \widehat{\overline{\rho}\tilde{u}_i\tilde{u}_j} && (18)\\ =& \widehat{\overline{\rho}\tilde{u}_i\tilde{u}_j} - \widehat{\overline{\rho}}\widehat{\tilde{u}_i}\widehat{\tilde{u}_j} && (19) \end{aligned}$$

This is an exact expression for which all the terms in the last equality can be determined by filtering various components and products of the solution. If we take the deviatoric part of (19) and make use of (12), (13), and (15) to eliminate $\widehat{t^d_{ij}}$ and $T_{ij}$ we obtain a system of equations where the only variable is $C\overline{\Delta}^2$.

$$L^d_{ij} = 2C\overline{\Delta}^2 M_{ij} \tag{20}$$

where

$$M_{ij} = \widehat{\overline{\rho}|S|S^d_{ij}} - \left(\frac{\widehat{\overline{\Delta}}}{\overline{\Delta}}\right)^2 \widehat{\overline{\rho}}|\widehat{S}|\widehat{S^d_{ij}} \tag{21}$$

Again, all terms in $M_{ij}$ can be calculated by filtering quantities obtained from the solution. As there are a number of approaches to solving (20), none of which is particularly novel to unstructured grids, the interested reader is referred to Ghosal et al.[20]. The calculations performed herein used a simple least-squares minimization of the error of (20), first proposed by Lilly [21]

$$C\overline{\Delta}^2 = \frac{1}{2}\frac{L_{ij}M_{ij}}{M_{ij}M_{ij}} \tag{22}$$

where we note that we no longer need to take the deviatoric portion of $L_{ij}$ since it multiplies the trace free $M_{ij}$ in the numerator. Following this procedure the Smagorinsky "constant" and the length-scale squared ($C\overline{\Delta}^2$) are dynamically determined to be a function of space and time. This procedure has been shown to have the proper behavior approaching walls and in transition (both from laminar to turbulent and from turbulent to laminar). To maintain a physical realizability of $C\overline{\Delta}^2$ (and therefore viscosity), the numerator and the denominator of (22) are sometimes averaged in the homogeneous directions of the problem. When this is not done and when this process is not sufficient to leave the total viscosity positive at all points in the domain, clipping of the total viscosity to a strictly non-negative result is usually employed.

The above constitutes a model for the deviatoric subgrid-scale stress. Models have been proposed for the trace of the subgrid-scale stress as well [8]. When this is not done, the subgrid scale stress is usually assumed to be added to the pressure thereby redefining it. For the low Mach number flows studied within this paper this assumption is acceptable and the additional model was deemed not necessary though the additional effort would not be significant.

It is worth mentioning that the dynamic model procedure is not limited to the use of Smagorinsky's model. Any subgrid-scale model may be applied at both scales, followed by the use of the procedure to determine the local constant. A family of dynamic models was proposed in Carati, Jansen and Lund [22] and current research is underway to assess the merits of some of the alternatives.

The dynamic model procedure obviates the choice of $\overline{\Delta}$ since it is built into the calculation. This is especially comforting for highly anisotropic grids where one might be at a loss for which cell length to chose for $\overline{\Delta}$.

Applications of the Smagorinsky-based dynamic model flourished within the spectral methods community and the finite difference community where it gained wide acceptance. Use of this model on unstructured grids required an extension of the filtering operator. A variety of filtering operators were developed in [1]. If $G$ is a 1-D "top-hat" filter and (16) is integrated with Simpson's rule the filtering operation on a structured grid leads to the following formula for each internal node $A$ ($A = 1, \cdots, n_p$) in the mesh with $n_p$ such internal nodes

$$\widehat{f}^A = \frac{1}{6}(4f^A + f_E^A + f_W^A)$$

where the subscripts $E$ and $W$ denote the point due East and West of node $A$. In multi-dimensions on a structured grid the process is simply repeated in each direction. On an unstructured grid there often is not a point due East or West. Furthermore, the location of the points that might approximate the due East or West neighbor is not simply determined through a recursive formula as is the case in a structured grid. One could pre-process these approximate neighbors and store a pointer list for each point in the grid. This would require additional memory which is unattractive. Also, for parallel machines this approach is very inefficient. The inefficiency stems from the fact that the nodal data is scattered among the processors and retrieval of that data requires substantial communication. The time required to perform these communication operations is often much larger than the time required to perform the actual calculations on parallel machines. For this reason it is attractive to explore element based filtering procedures which minimize the amount of communication required. Four alternative filtering operators have been developed and are described next.

### *4.1 Face-based filtering*

Start by obtaining the function at the element centroids, $f^e$, where the superscript $e$ corresponds to the $e^{th}$ element. Then define the following filter operation,

$$\widehat{f}^e = w_0 f^e + \sum_{i=1}^{n_f} w_i f_i^e$$

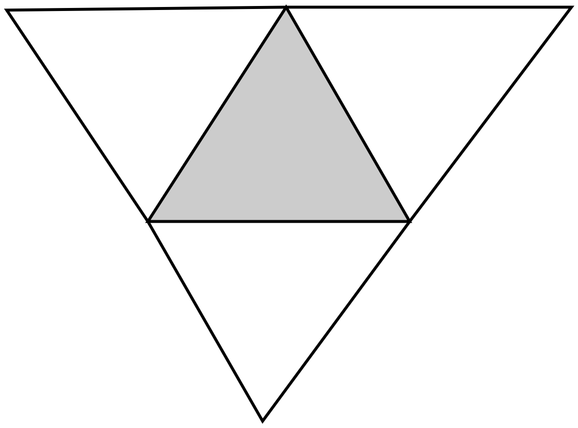

Fig. 3. Face-based filter for the shaded triangle is determined through a weighted combination of the function value within the shaded triangle and the function values within the triangles (tetrahedra in 3D) which share a face with it (the 3 white triangles (4 tetrahedra in 3D)).

where $f_i^e$ is the function value at the centroid of the element on the other side of the $i^{th}$ face (there are $n_f$ such faces for each element), and $w_i$ are filter weights. The process is more easily visualized using a 2-D simplification. For example, triangles have three faces (actually edges) , therefore, the filtering operation of a given function for a particular triangle involves the function value within the triangle and the function values within the three other triangles which surround it as illustrated in Figure 3.

This method is not optimal for most finite element methods because it requires an additional data structure to determine the elements which lie on the other side of each face of a given element. This data structure is not immediately available from existing finite element data structures in most codes. It could be pre-processed and stored, but the additional memory overhead is unattractive. This method may prove effective for finite volume methods using a face-based data structure.

### *4.2 Derivative-based filter*

An alternative approach is to approximate the the filtering operator by

$$\widehat{f}(\mathbf{x}) = f(\mathbf{x}) + \frac{1}{m}\tilde{\nabla}^2 f(\mathbf{x}) + O(\overline{\Delta}^4)$$

which extends the work of Carati [23] to non-isotropic grids. Here $\tilde{\nabla}$ is a non-dimensional gradient (i.e. the gradient in a particular direction multiplied by the discretization width in that direction).

$$\tilde{\nabla} = \Delta_x \frac{\partial}{\partial x}\mathbf{i} + \Delta_y \frac{\partial}{\partial y}\mathbf{j} + \Delta_z \frac{\partial}{\partial z}\mathbf{k}$$

It is easily verified that this procedure gives the same result as a one-dimensional top-hat filter integrated with Simpson's rule ($m = 3$). In general $m = d + 2$ where $d$ is the number of space dimensions.

This filter can be evaluated quite rapidly with the finite element method

$$\hat{f}^A = f^A - \frac{1}{d+2}\{M^{BA}\}^{-1}\left(\int_{\Omega} N^B_{,i_n} f_{,i_n} d\Omega - \int_{\Gamma} N^B f_{,i_n} n_{i_n} d\Gamma\right)$$

where $N^A$ is the basis function for node $A$ (likewise $B$ can be any node ($B = 1, \cdots, n_p$)), $\{M^{BA}\}$ is the finite element "mass matrix", $\Omega$ is the spatial domain, $\Gamma$ is the boundary of the domain, and the subscript $, i_n$ denotes differentiation in the $i^{th}$ direction multiplied by the length of the element in this direction. Note that we have included the boundary terms (there will be contributions when $f$ is the strain-rate tensor due the non-zero strain-rates at the boundaries).

While this method requires more floating point operations than the face-based filter, it requires no additional data structures (it uses existing finite element data structures) and very little communication. Algorithms of this type are already coded for the viscous terms of the formulation, consequently these operations were easily parallelized.

The filtering operation defined above requires that all quantities that need to be filtered must be defined at the nodes. The most commonly used basis functions in finite element methods are $C^0$ continuous piece-wise polynomials. Function spaces of this type yield gradient quantities (such as the strain-rates) that are discontinuous at element boundaries. Before the filtering operator can be applied it is necessary to project the strain-rates from the elements to the nodes. This is accomplished by a consistent finite element projection operator

$$S^A_{ij} = \{M^{BA}\}^{-1} \sum_{e=1}^{n_{el}} \int_{\Omega^e} N^B S^e_{ij} d\Omega_e$$

here $S^A_{ij}$ is the strain-rate tensor at node $A$ and $S^e_{ij}$ is the strain-rate tensor as defined in element $e$. Note that the sum is over all the element domains $\Omega_e$ (there are $n_{el}$ such domains).

With the strain-rate tensor globally projected to the nodes, we next interpolate $S_{ij}(\mathbf{x})$ with the basis functions,

$$S_{ij}(\mathbf{x}) = \sum_{A=1}^{n_p} N^A(\mathbf{x}) S^A_{ij}$$

The above procedure has been implemented in the parallel code. The tests thus far have used a "lumped mass" for $\{M^{BA}\}$ making inversion trivial. The cost of the dynamic model calculation of the eddy viscosity is less than one-fifth of a non-linear iteration. Therefore, even when only performing two non-linear iterations the cost of the dynamic model is less than 10 percent of the total cost. This is, relatively, as cheap or cheaper than many three-dimensional structured grid filtering operators. As mentioned before it also requires no additional memory.

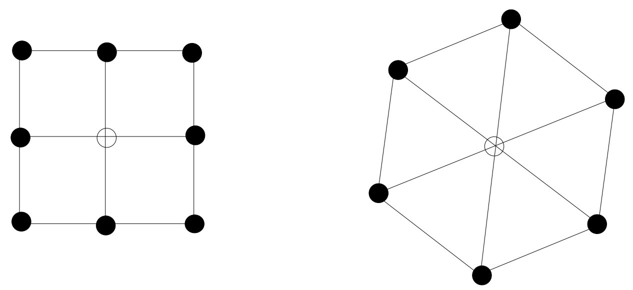

Fig. 4. Domain of filter for structured and unstructured grids. In each case the filter $G$ takes on a value of 1 on the domain created by the elements which touch the hollow node and zero elsewhere. In 3D the structured grid filter domain becomes a hexahedron whereas the unstructured grid filter domain approximates an ellipsoid.

### *4.3 Generalized top-hat filter*

The third method generalizes the notion of a top-hat or box filter to unstructured grids. Since the element domains do not necessarily form boxes it is more practical to define the filter function for node $A$ as $G(\mathbf{x}^A) = 1$ for all the elements which have node $A$ as a vertex (see Figure 4). In three dimensions this generalized box filter is an approximation to an ellipsoid (for isotropic grids it is approximately a sphere). This filter function is easily integrated against the function we desire to filter using Gauss quadrature. This method has been implemented and is slightly cheaper than derivative-based filter and has proven to be just as accurate.

### *4.4 Function based filter*

The fourth method is only appropriate when using a higher-order function space. Consider for example a one-dimensional quadratic element as shown in Figure 5 on the left. It is possible to construct a filter from a combination of two interpolations. In our one-dimensional example the filtered value of a function $f$ at the center of the element can be a weighted combination of the quadratic interpolation (which involves all three points) and linear interpolation of the endpoints viz.

$$\widehat{f} = \beta(f_{lin} + \alpha f_{quadr})$$

$\alpha$ and $\beta$ can be determined to represent the filter of choice. For example $\alpha = \frac{1}{4}, \beta = \frac{2}{3}$ is equivalent to a top-hat filter integrated with Simpson's rule. Quadrilaterals (in two-dimensions) and hexahedra (in three-dimensions) pose no additional difficulty as they are constructed from tensor products of these one-dimensional functions. Triangles and tetrahedra are not quite so trivial but can none the less be constructed (see Figure 5, right).

Implicit in this method is the assumption that it is sufficient to calculate the dynamic model coefficient once in each element. This filtering method is only meaningful on the interior of the element since at the endpoints (or corners in multi-dimensions) the different functional representations yield the same value. At these positions no filtering would be accomplished. This is of little concern for tetrahedral meshes since in this case there are roughly the same number of quadratic tetrahedra

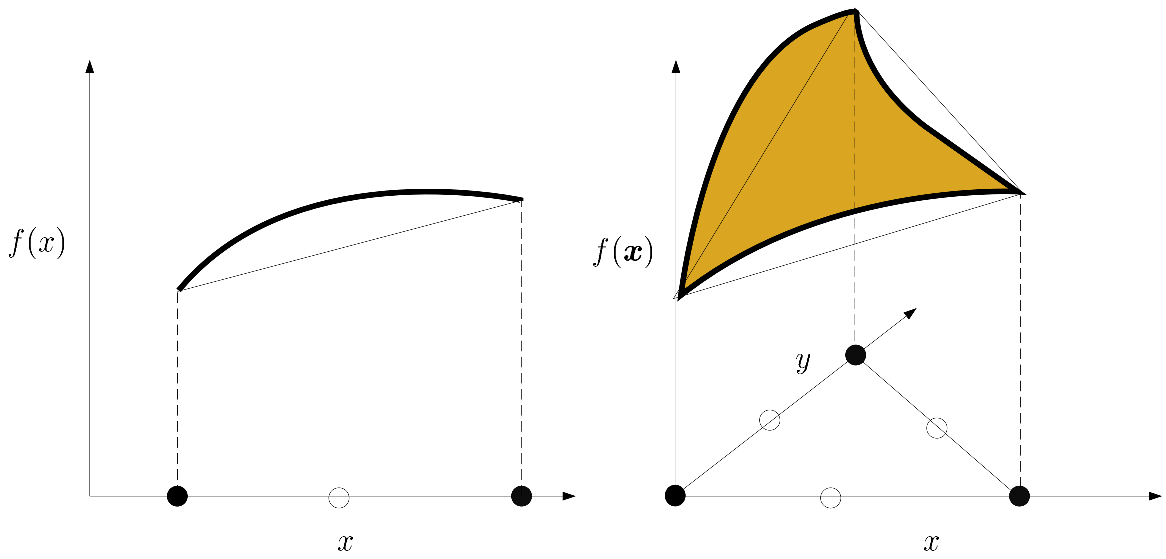


Fig. 5. On the left, a one-dimensional quadratic element is comprised of three nodes which quadratically interpolate (bold line) the nodal values of the function $f(x)$. A linear interpolant can be constructed (thin line) from the end nodes (filled circles). The concept is generalized to triangles on the right.

as there are nodal points. Therefore, the number of points where the dynamic model coefficient is evaluated is roughly the same. What has changed is the point in space where the dynamic model coefficient is calculated. We have simply moved the position where we evaluate the dynamic model coefficient from the nodes to the element centroids. The motivation for doing this is that no communication is required to determine the dynamic model coefficient at the centroid using this method. This approach promises to be far less costly than the other approaches for higher order basis functions.

In the above description we have assumed a Lagrange basis which does not have a hierarchical nature. Recently, a set of hierarchical basis functions has been developed by Shephard, Dey and Flaherty [24] which look very promising for this type of filtering.

It should be pointed out that all four filtering procedures drew insight from mimicking well understood, structured-grid-based filters. This approach is somewhat flawed since the structured-grid-based filters are of finite accuracy. Instead, future work will study the filter transfer function implied by these methods directly.

## 5 Model validation

To verify that both the numerical method and the subgrid-scale model are appropriate for large eddy simulation two canonical flows have been simulated; decay of isotropic turbulence and turbulent channel flow. In each case we have not done a careful refinement study as would be appropriate for a physical study of these flows. Instead we have used very coarse grids to get an indication of the model's performance with the resolution expected to be attainable in the more complex flows for

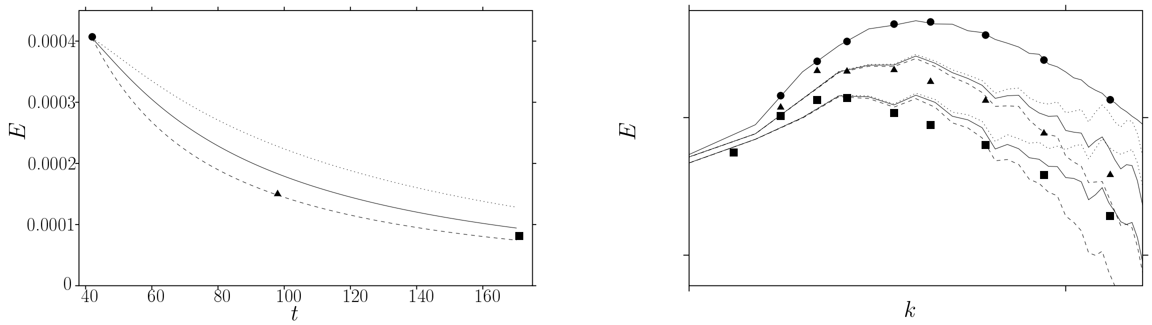


Fig. 6. Decay of isotropic turbulence on a $32^3$ grid compared to the experimental data of [25]. On the left is the decay of resolved kinetic turbulent energy; on the right the filtered velocity spectra where the wavenumber ranges from 1 to 16 (normalization by $2\pi/L$, where $L$ is the size of the computational box). —— Present dynamic model; ---- fixed coefficient Smagorinsky model ($C_s = 0.17$); ······ no model. •, ▴, ■ Experiment at $t = 42$, 98 and 171 respectively.

which these methods are directed. It is worth pointing out that these flows do not have adequate length scale variation to give unstructured grids the point reduction advantage that we see in more complex flows.

### *5.1 Decay of isotropic turbulence*

A common validation test for large-eddy simulation codes is the isotropic homogeneous turbulence decay. The simulation was run on a $32^3$ periodic grid and is designed to replicate the results of the experiment conducted by Comte-Bellot & Corrsin [25]. To accurately match the experiment, the initial velocity field of the simulation must match the *filtered* spectrum given at the first experimental station, corresponding to $t = 42$ with the conventions of the reference. Then, the actual decay is simulated and results are compared to the experimental data given at $t = 98$ and 171. During this computation, the time step is constant with 45 time steps per eddy turnover time at the first experimental station ($t = 42$).

Figure 6 shows the decay of the kinetic turbulent energy obtained with the dynamic model, with the Smagorinsky coefficient fixed at $C_s = 0.17$, and with no subgrid-scale model. The variations observed are important since they show that the numerical dissipation does not dominate the subgrid-scale model. Though the fixed-coefficient Smagorinsky model seems to provide even better results, the velocity spectra, presented on the right in Figure 6, show that this would be an errant conclusion since the dynamic model represents the shape of the spectrum much better while the fixed coefficient simply benefits from error cancellation. Finally the spectrum obtained with no model is flatter, which indicates that the numerical dissipation inherent to the scheme is not dominant.

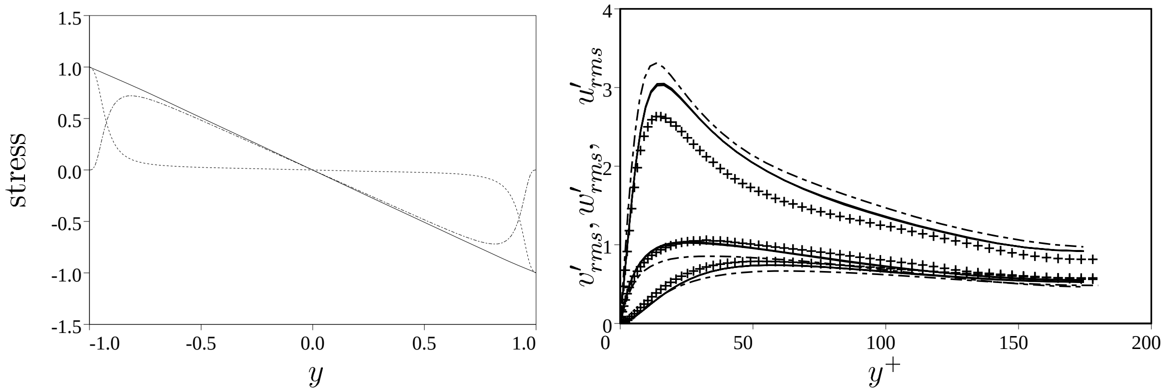


Fig. 7. On the left is the stress balance for channel flow. Note the Reynolds stress (—-— ) and the viscous stress (---- ) sum to a linear total stress (——— ) which balances the pressure gradient. On the right is the comparison of velocity $rms$ between central difference method (---- ), finite element method (——— ), and DNS (+ ). Top to bottom the curves are $u'_{rms}$, $v'_{rms}$ and $w'_{rms}$ normalized by the shear velocity ($u_\tau$).

### *5.2 Channel flow*

The Reynolds number 180 channel flow was studied using direct numerical simulation (DNS) by Kim et al.[26]. We have performed a large-eddy simulation of the same flow conditions but with a coarse grid of 32 × 64 × 32 elements. In Figure 7 we show the stress balance attained from averaging the flow over 10 time units (one time unit is the time it takes a particle to flow though the channel along the centerline). We have also included the root mean square ($rms$) of the velocity components in Figure 7. Here the results are compared to a $2^{nd}$ order central difference scheme. Note the improvement in all $rms$ quantities, especially the $v'_{rms}$ and $w'_{rms}$ components.

## 6 Airfoil simulation

As mentioned earlier, the real promise of this method is its application to more complex flows where there is a large variation in the length scales requiring resolution. Here, unstructured grids can refine or coarsen to match the physical length scale requirements of a particular region rather than letting the length scale of one region determine the grid requirements for the entire problem. While the same can be done, to some extent by overset or Chimera grids, for many flows there is a gradual and continuous change in length scales making this approach impractical.

We have chosen the NACA 4412 airfoil at maximum lift as the demonstration simulation for this approach for a variety of reasons. First, it is a problem of significant interest since it would be the first LES of an aircraft component. Second, this flow has been the subject of three experimental studies (Coles and Wadcock [27], Hastings and Williams [28], and Wadcock [29]). The first study found the maximum lift angle to be $13.87^o$. The later studies found the angle to be $12^o$. Wadcock reports in the later study that the early data agree very well with his new data at $12^o$, suggesting that the early experiment suffered from a non-parallel mean flow in the Caltech wind tunnel. It

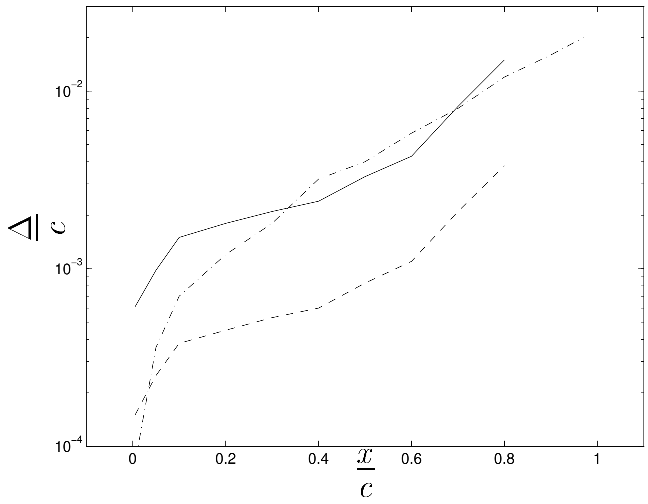


Fig. 8. Comparison of length scales over the airfoil surface. 200 wall units (streamwise near-wall spacing) ——— , $\delta_{99}/5$ (outer-layer spacing) —·— , 50 wall units (spanwise near-wall spacing) - - - - . Here, $\delta_{99}$, is the thickness of boundary layer as determined by the velocity reaching 99% of its maximum.

should be pointed out that the Reynolds-averaged simulations are usually run at $13.87^o$ and do not agree with the data when run at $12^o$. It is hoped that LES can clarify this controversy. The third reason for considering this flow is the variety of flow features which provide an important test of the dynamic model. Starting from the nose where the flow stagnates, thin laminar boundary layers are formed in a very favorable pressure gradient. This pressure gradient soon turns adverse, driving the flow toward a leading edge separation. Only the onset of turbulence can cause the flow to remain attached or to re-attach if it did separate. The persistent adverse pressure gradient eventually drives the turbulent flow to separate in the last 20 percent of chord. The separation bubble is closed near the trailing edge as the retarded upper surface boundary layer interacts with the very thin lower surface boundary layer. The large difference in boundary layers creates a challenging wake to simulate. Only the dynamic model can be expected to perform satisfactorily in this variety of situations: from the laminar regions where it must not modify the flow at all to the turbulent boundary layers and wake where it must represent a wide variety of subgrid-scale structures. The flow configuration we have chosen is that of Wadcock [29] at Reynolds number based on chord $Re_c = u_\infty c/\nu = 1.64 \times 10^6$, Mach number $M = 0.2$, and $12^o$ angle of attack.

### *6.1 Mesh design*

The design of the mesh followed the suggestion of Moin and Jimenéz [9] and Chapman [30] with the spanwise spacing set at 50 wall units and the streamwise spacing set at 200 wall units in the near wall region. With the experimental values of $C_f$ available, these spacings have been plotted as a function of streamwise position in Figure 8. Away from the wall the eddies no longer scale on wall units but instead scale on boundary layer thickness ($\delta$). A third curve has been included on Figure 8 to illustrate how this spacing varies with streamwise position based on Wadcock's measured boundary layer thickness, assuming 5 elements per boundary layer thickness.

Several points can be made in this figure. First, all three curves change by over an order of magnitude from the tip to the tail region. This illustrates how an unstructured grid saves points by matching resolution to the local changes in the length scales in the streamwise direction. For example, a structured grid would be forced to carry the fine spanwise resolution required near the nose throughout the entire domain. Secondly, when comparing the near-wall spanwise resolution to the outer-layer resolution, it is clear that coarsening the spanwise resolution as the distance from the wall increases is justified. The final point, apparent from this figure, is that coarsening of the streamwise resolution in the outer layer is not justified. In fact, over much of the airfoil surface the outer-layer grid resolution is more restrictive than the inner-layer resolution. The choices of 200 wall units and 5 points per boundary layer thickness are somewhat arbitrary, but they are believed to be comparable in their degree of coarseness. It is interesting to observe that the crossover between these two curves corresponds to $Re_\tau = (u_\tau \delta_{99}/\nu) = 1000$. Therefore, when above 1000, the inner-layer resolution is the most restrictive. Otherwise, the outer-layer resolution is the most restrictive. Only at higher Reynolds numbers will coarsening in the streamwise direction be justified.

Early on, it became apparent that existing automatic mesh generators could not produce grids to match the above spacing requirements in a fashion smooth enough to retain accuracy. A significant amount of effort was expended writing a custom mesh generator for this effort. Figure 9 shows the fruit of this labor. This, rather complex figure, shows a cut through the airfoil cross section connected to the surface mesh. The surface mesh appears nearly black because of the very fine near-wall resolution. The additional surfaces parallel to the surface mesh are offset so that the coarsening of the grid with increasing distance from the wall can be observed. The final grid is outside of the boundary layer and nearly all of the spanwise resolution has been shed, since it is not needed to compute a two-dimensional inviscid region. It is worth noting here that the equivalent structured grid, which would carry the finest $z$ spacing and the given $x$ spacing out to the boundary, would require 27 times as many points. The contours of the instantaneous streamwise velocity are shown on the same planes as the grid to confirm that the fine grid is in regions of fine-scale turbulence structures (very near the wall) while the structures become larger with increasing distance from the wall. It is important to note that this is just one instant of the flow field which is rapidly changing in time as well as space.

The mesh generator which created the grid shown in Figure 9 has been referred to as a "hand-crafted" mesh generator with good cause. Several geometrical features of the problem have been built into the mesh generation process. First, since the 3-D model is actually a prismatic extension of a 2-D model, the mesh generation process is broken into two steps. In the first step a 2-D triangular mesh is generated. In the second step, the 2-D triangular mesh was converted into a 3-D wedge mesh. This wedge mesh is in turn used to create a tetrahedral mesh. The details of this procedure are described in the following paragraphs.

To gain further control of the smoothness in the variation of the mesh spacing, the 2-D mesh generation is itself broken into successive layers of quasi-1-D mesh generation or what is sometimes called paving. The paving process requires the creation of generalized level sets of increasing distance from the solid boundaries. We say generalized here because the thickness of the layers are contracted or expanded to reflect increasing or decreasing shear stress and increasing or decreasing boundary layer thickness. In this way we can carefully control both the near wall spacing *and* the number of

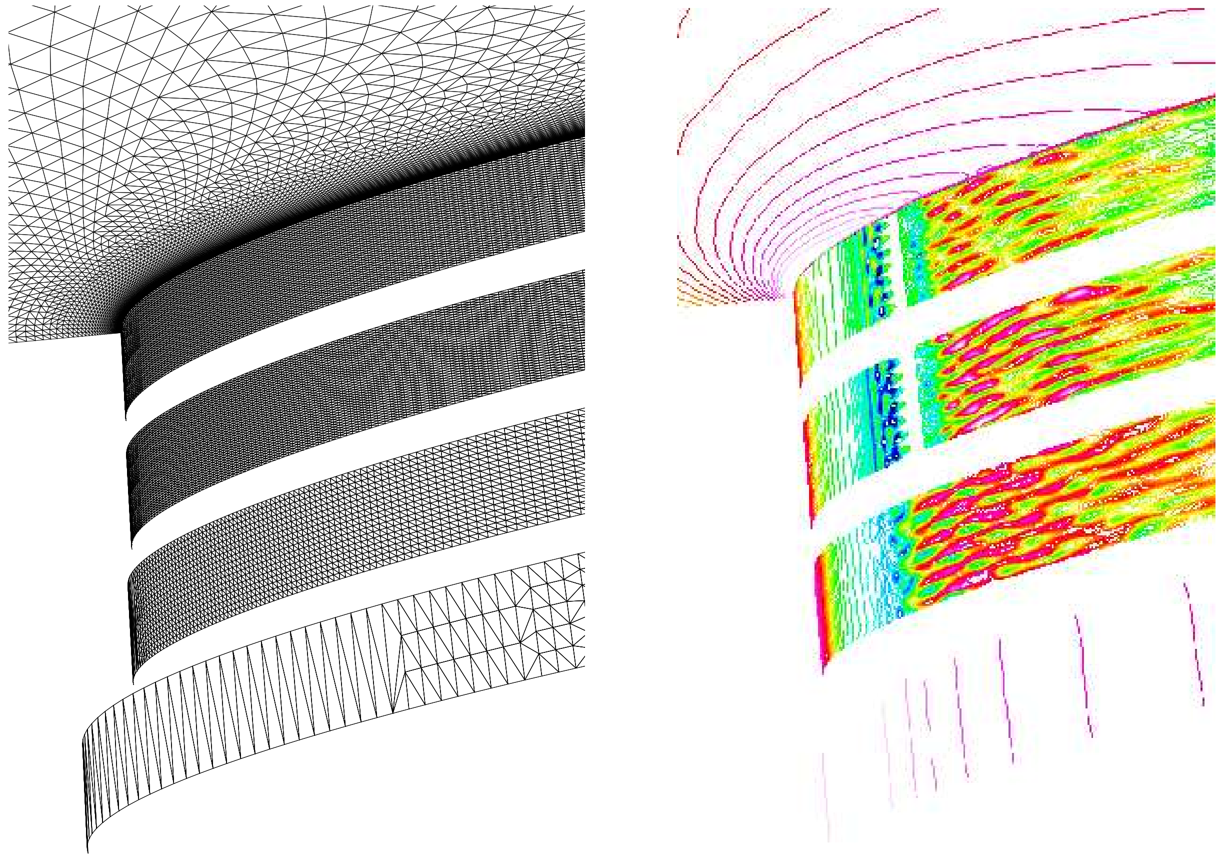

Fig. 9. On the left, an example of a "hand-crafted" mesh which meets the smoothness requirements in the turbulent regions is shown. On the right, the contours of the streamwise velocity are shown on the same planes. The cross-sectional plane is shown intersecting a plane about 10 layers off of the the wall. The next 3 planes are at 15 20 and 50 layers off of the wall. Here, light (red) contours denote high speed while dark (blue) contours denote low speed.

points in the boundary layer. This ability has been observed to be critical to the optimal performance of the method. Once the level sets are created, a point distribution is placed on the airfoil cross-section. Again this point distribution is carefully controlled to provide an even resolution of the shear stress (finer in regions of high shear where the near-wall turbulent structures will be small and coarser in regions of low shear where the same structures will be larger). Recall also that the spacing along the surface is typically 200 times larger than the spacing between the generalized level sets to reflect the much larger gradient normal to the surface compared to the gradient along the surface (a property of high Reynolds number flows). Next, the point distribution on the next layer of the level set is likewise carried out. Then, a layer of elements is paved between these two point distributions. This process insures that a smooth variation of the element areas is achieved. The process is repeated by distributing points on the next level set and paving between this layer and the previous layer. Very gradually, the strict near-wall resolution requirements are relaxed as the level sets move toward the free stream where the resolution requirements are much less. The experimental data mentioned above, provided the information needed concerning the rate of this relaxation. There were some additional challenges presented by the wake region. Space limitations prevent a full explanation here.

With the triangle mesh of the cross-section created the next step was to generate wedge elements to the desired width of the domain. Recall, from Figure 8, that the resolution in the third dimension is highly variable. Therefore, to achieve the point reduction that is desired, each point in the 2-D mesh must be allowed to have a different number of points in the third dimension. That is to say that, near the wall and especially near the leading edge, as many as 200 points are placed along a line extending normal to the point in the 2-D grid into the third dimension (creating extremely fine $z$ resolution). This is contrasted by a point from the 2-D grid in the freestream which will have only one other point along the same length line into the $z$ direction. Experimental data and physical insight provide for the smooth variation in the number points in the third dimension. With the points distributed along lines from the 2-D grid the algorithm is ready to create the tetrahedral mesh. To preserve the high quality in the element shape, tetrahedral elements are created by marching from the 2-D face through the wedge, ending at the triangular face on the end of the wedge. This marching process insures high quality tetrahedra since, for each interior face, the closest of three candidate tessellation points is always chosen so long as it does not create a conflict with an existing free face (from an already tetrahedronized wedge). Due to the structure of our 2-D grid, the free-face list never exceeds 500 making this process extremely fast. The 3-D mesh generator creates elements at over 1 million elements per minute on a single SGI R10000 processor.

As mentioned above, the benefits of this technique are completely controlled element size and anisotropy. The obvious drawback with this approach is that the *techniques* are not easily extendible to arbitrarily complex geometries. This was not the purpose of the work. The purpose was to demonstrate that careful grid generation could not only lead to precise matching of mesh length scales to physical length scales but also that the savings obtained could allow the method to be applied to problems whose computational complexity was outside the capability of existing structured grid methods. Furthermore, the smoothness of the grid (which is really dictated by the smoothly varying physical length scales) allows the method to realize its capacity for high accuracy.

Currently, work is underway to bring the experience base obtained with these "hand-crafted" meshes to bear on automatic mesh generation. It is believed that recent advances in Beall et al.[31] show great promise through the ability to assign surface and volume attributes to the solid model. An example of an important lesson learned from our "hand-crafted" meshes is that high-aspect-ratio tetrahedra, even those with very large angles, present no problem; so long as they are used in regions where the gradient has an orthogonal orientation to the grid anisotropy. This experience is a counter argument to the Babuška and Aziz [32] finding, though, in that case, the gradient was chosen to be aligned with the grid anisotropy (i.e. elements were stretched in the direction of the gradient which is exactly counter to the purpose of stretching elements in the first place). Clearly, Babuška-Aziz condition is critical when the orientation of the gradient is not predictable *a priori*. However when it is known, as is the case in wall bounded flows, the condition is unnecessarily restrictive. The high anisotropy in the near-wall of turbulent flows, combined with a coarsening of the elements, absolutely requires the existence of very large angle elements. Existing automatic mesh generators which insist on elimination of very large angle elements are forced to carry structured grid layers all the way through the boundary layer, forfeiting the point reduction which was achieved by our "hand-crafted" method.

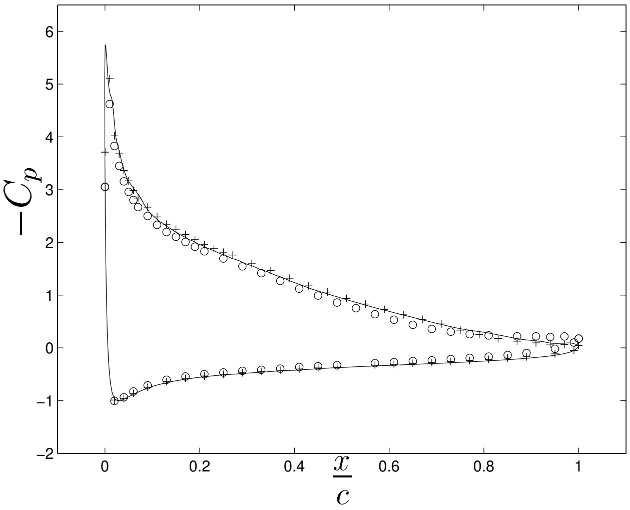


Fig. 10. Coefficient of pressure on the airfoil surface comparing calculation (—— ) to Hastings' experiment with transition strip (○ ) and without transition strip (+ ).

*6.2 Airfoil simulations*

The first simulations of this problem assumed free transition and a far-field boundary condition to approximate real flight conditions. Despite grid refinement studies, described in Jansen [3], it became clear that the flow was not converging to the experimental results, especially the velocity profiles. Figure 10 offers some explanation of the lack of convergence. Here we compare to two experimental data sets from Hastings. Note the good agreement with the pressure data taken with the transition strip removed. However, all the remaining experimental data (velocity profiles, boundary layer parameters, and *rms* profiles) were taken with the transition strip in place, since they wished their study to be representative of higher Reynolds number flows. Though the shift in pressure data may appear small it seems to have a profound effect on the velocity profiles, since the flat region near the trailing edge signals a massive separation that is much milder in the un-tripped case.

This finding motivated a second study of this flow where the wind tunnel walls and the transition strip were accounted for in the calculation. Hastings' transition strip was composed of Ballotini sprinkled randomly over the leading edge region of the flow. This was deemed more difficult to simulate than Wadcock's trip. Wadcock used a piece of tape with serrations cut into the windward side. A mesh was constructed to represent the exact, position, height and shape of this tape which can be seen in Figure 11. The results of these simulations as well as the mesh used to model the wind tunnel walls can be found in Jansen [4]. An additional figure is provided here (Figure 12) to illustrate the evolution of small-scale activity near the leading edge to very large vortical structures near the trailing edge. The boundary layer growth is also clearly shown. Note that at this Reynolds number the boundary layer on the lower surface remains laminar.

Current simulations employ approximately 8.5 million tetrahedra (1.5 million nodes). With 5 variables per node, each time step requires the solution of 7.5 million non-linear equations. Just as the spatial scales vary by two orders of magnitude from the tip to the tail, the temporal scales do as

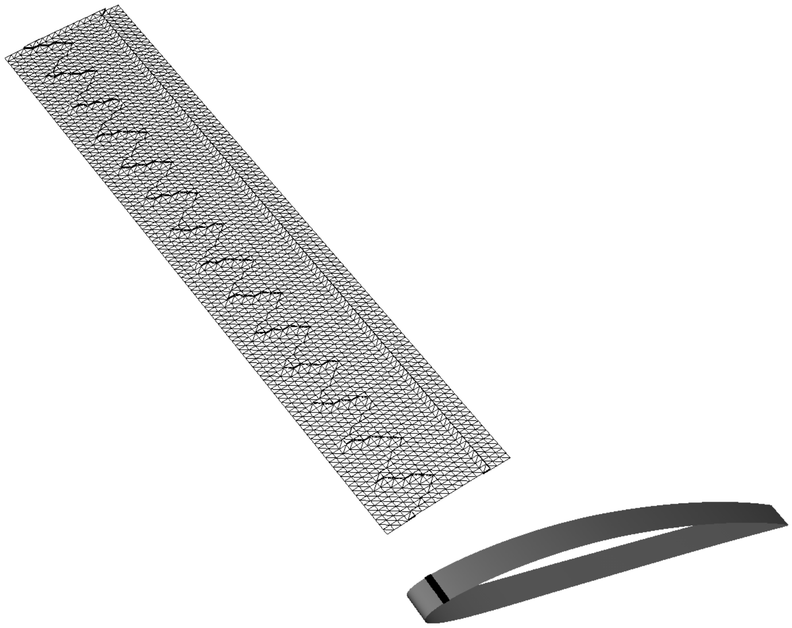

Fig. 11. Geometric model of the exact shape, height and position (shown in black on the airfoil surface) of Wadcock's transition strip.

well. It is the time accurate resolution of the entire domain that makes this problem among the largest CFD simulations undertaken to date. The difficulty arises from the need to choose a time step small enough to accurately represent the physics of the small scales in the thin boundary layer near the nose. To be certain that the flow is in a statistically stationary state this time accuracy must be carried out for a vary long time. Typically, one measures this time to statistically stationary state in "flow-over-chord" time units. Simply put, this is the time it takes for a Lagrangian particle to travel from the nose to the tail at the freestream speed. When starting from a new grid, a new angle of attack, or other significant modification to the simulation 5-10 flow over chord times are needed to flush out the transient. Then, with the flow in a statistically stationary state the flow is usually allowed to continue for another 5 flow-over-chord times to collect statistics for the mean flow and its higher moments (such as Reynolds stresses).

The fine resolution at the nose requires a time step that is $1/10,000^{th}$ of a flow-over-chord time. Such a time step leads to at least $100,000$ time steps per simulation. Therefore, for a typical simulation, $7.5 \times 10^{11}$ non-linear equations are solved, making these among the largest unstructured-grid simulations undertaken thus far. To give an estimate of the computational time required for such a simulation one needs to examine the computational time per step. On the 512 processor CM5 at the Army High Performance Computing Research Center our code requires a little under a minute per time step. At this rate, 1666 hours are required per simulation or about 70 full days. Clearly, these calculations require an enormous commitment of resources and are far outside of practical engineering calculations with the current computing algorithms and platforms. However, it is worth noting that the salient features of this flow ($C_p$ and point of separation) were in good agreement with the experimental data on a grid that was an order of magnitude smaller. In our current calculations we are doing careful refinement studies to get more detailed information such as velocity profiles in the separated region and Reynolds stresses. This detailed information will also be

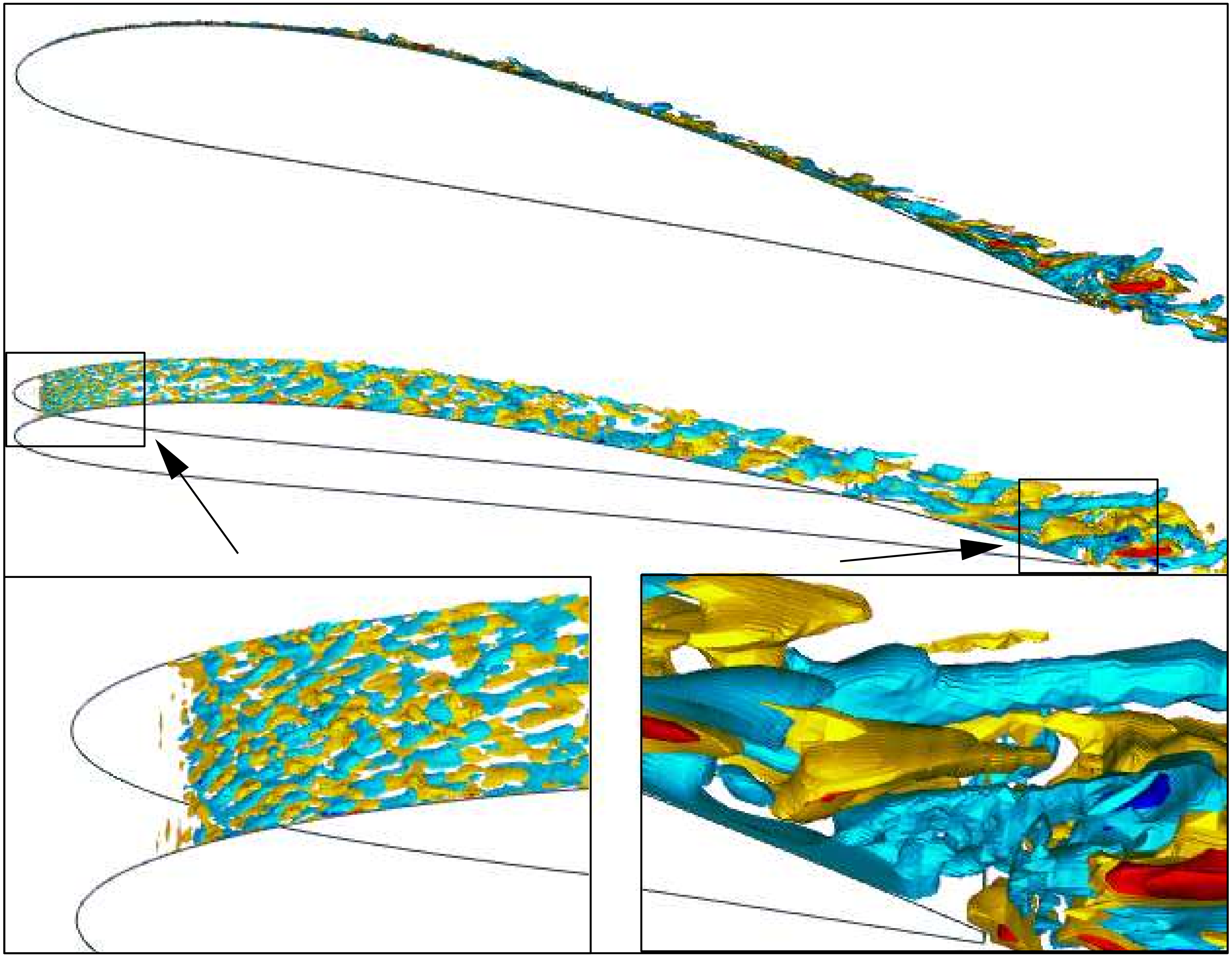

Fig. 12. Tip-to-tail development of the turbulent structures identified by spanwise velocity isosurfaces in the boundary layer over the upper surface of the airfoil. Top figure provides a side view which illustrates the development of the boundary layer thickness. The middle figure tilts slightly to show the spanwise structures. Lower left figure zooms in on the early boundary layer while the lower right zooms on the tail region illustrating the dramatic variation in spatial scales.

used to understand why RANS models are unable to predict this flow by comparing these models, term-by-term, with statistics collected from the full simulation. Such detailed physical simulations always require a level of effort far beyond that typically used in industrial practice.

## 7 Conclusions and future directions

While much progress has been made, the NACA 4412 airfoil simulation has still not yielded complete agreement between LES and experiment. Along the way it has served as a vehicle for the development and severe testing of both the accuracy and efficiency of our numerical method. To this end we have begun development of new, higher-order methods which employ hierarchical basis functions to add $p$ adaptivity to the already present $h$ adaptivity. These basis functions have already been tested on canonical problems (Helmholtz, Poisson, and linear elasticity; see Dey [33]). Their implementation into the Navier-Stokes code is currently being tested on simple flows. Their application to turbulent flows is expected shortly. It is hoped that the higher accuracy of these approaches will reduce the computational effort required in the current simulations.

Other simulations are also being carried out with this code though the space remaining here allows only a brief mention of them. Bastin [18] is using the code to compute the noise from a supersonic jet ($M = 1.4$, $Re_D = 1.5 \times 10^6$). We have also recently started a simulation of turbulent flow over a cavity ($Re_H = 27000$, $Re_\theta = 5100$) with the goal of predicting/altering the acoustic resonance which sometimes occurs in this geometry. The computational effort associated with these simulations is substantially less than that of the airfoil (the jet problem is about a factor of 4 smaller, the cavity about a factor of 20 smaller).

## 8 Acknowledgments

The author wishes to acknowledge the collaboration of Francois Bastin on the decay of isotropic turbulence problem and on the development of the MPI version of the code, Farzin Shakib on the channel flow problem, S. Scott Collis, Juin Kim, and Christian Whiting on the small disturbance problem, and many helpful suggestions by Parviz Moin, Thomas Lund, Hans Kaltenbach, Rajat Mittal, and Massimiliano Fatica and others at the Center for Turbulence Research (CTR) on the airfoil simulation.

This work has been supported by the AFOSR and by CTR (a joint NASA-Stanford University program). It has also enjoyed generous computer resources from the Army High Performance Computing Research Center (AHPCRC) and the Naval Research Lab (NRL).

## References


[1] K. E. Jansen, "Unstructured grid large eddy simulation of flow over an airfoil", in *Annual Research Briefs*, Center for Turbulence Research, NASA Ames / Stanford University, (1994) 161–173.

[2] K. E. Jansen, "Large-eddy simulation using unstructured grids", in *Advances in DNS/LES*, Greyden Press, Columbus, (1997) 117–128.

[3] K. E. Jansen, "Preliminary large-eddy simulations of flow around a NACA 4412 airfoil using unstructured grids", in *Annual Research Briefs*, Center for Turbulence Research, NASA Ames / Stanford University, (1995) 61–72.

[4] K. E. Jansen, "Large-eddy simulation of flow around a NACA 4412 airfoil using unstructured grids", in *Annual Research Briefs*, Center for Turbulence Research, NASA Ames / Stanford University, (1996) 225–232.

[5] T. J. R. Hughes, L. P. Franca, and M. Mallet, "A new finite element formulation for fluid dynamics: VI. Convergence analysis of the generalized SUPG formulation for linear time–dependent multidimensional advective–diffusive systems", *Computer Methods in Applied Mechanics and Engineering*, **63** (1987) 97–112.

[6] T. J. R. Hughes, L. P. Franca, and G. M. Hulbert, "A new finite element formulation for fluid dynamics: VIII. The Galerkin / least–squares method for advective–diffusive equations", *Computer Methods in Applied Mechanics and Engineering*, **73** (1989) 173–189.

[7] L. P. Franca and T. J. R. Hughes, "Convergence analyses of Galerkin least–squares methods for symmetric advective-diffusive forms of the Stokes and incompressible Navier-Stokes equations", *Computer Methods in Applied Mechanics and Engineering*, **105** (1993) 285–298.

[8] P. Moin, K. Squires, W. H. Cabot, and S. Lee, "A dynamic subgrid–scale model for compressible turbulence and scalar transport", *Physics of Fluids*, **3** (1991) 2746–2757.

[9] P. Moin and J. Jimenéz, "Large–eddy simulation of complex turbulent flows", in *AIAA 24th Fluid Dynamics Conference*. AIAA-93-3099, (1993).

[10] M. Germano, U. Piomelli, P. Moin, and W. H. Cabot, "A dynamic subgrid–scale eddy viscosity model", *Physics of Fluids*, **3** (1991) 1760.

[11] Guillermo Hauke, *A unified approach to compressible and incompressible flows and a new entropy-consistent formulation of the k-epsilon model*, Ph.D. thesis, Stanford University, 1995.

[12] F. Shakib, *Finite element analysis of the compressible Euler and Navier-Stokes equations*, Ph.D. thesis, Stanford University, 1989.

[13] L. P. Franca and S. Frey, "Stabilized finite element methods: II. The incompressible Navier-Stokes equations", *Computer Methods in Applied Mechanics and Engineering*, **99** (1992) 209–233.

[14] G. Hauke and T. J. R. Hughes, "A unified approach to compressible and incompressible flows", *Computer Methods in Applied Mechanics and Engineering*, **113** (1994) 389–396.

[15] J. Chung and G. M. Hulbert, "A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized-$\alpha$ method", *Journal of Applied Mechanics*, **60** (1993) 371–75.

[16] Z. Johan, T. J. R. Hughes, and F. Shakib, "A globally convergent matrix-free algorithm for implicit time marching schemes arising in finite element analysis", *Computer Methods in Applied Mechanics and Engineering*, **87** (1991) 281–304.

[17] M. R. Malik, T. A. Zang, and M. Y. Hussaini, "A spectral collocation method for the Navier-Stokes equations", *Journal of Computational Physics*, **61** (1985) 64–88.

[18] F. Bastin, "Jet noise using large eddy simulation", in *Annual Research Briefs*, Center for Turbulence Research, NASA Ames / Stanford University, (1996) 115–132.

[19] J. Smagorinsky, "General circulation experiments with the primitive equations, I. The basic experiment", *Monthly Weather Review*, **91** (1963) 99–152.

[20] S. Ghosal, T. S. Lund, P. Moin, and K. Aksevoll, "The dynamic localization model for large eddy simulation of turbulent flows", *Journal of Fluid Mechanics*, **286** (1995) 229–255.

[21] D. K. Lilly, "A proposed modification of the Germano subgrid–scale closure", *Physics of Fluids*, **3** (1992) 2746–2757.

[22] D. Carati, K. E. Jansen, and T. S. Lund, "A family of dynamic models for large-eddy simulation", in *Annual Research Briefs*, Center for Turbulence Research, NASA Ames / Stanford University, (1995) 35–40.

[23] D. Carati. Personal Communication, 1994.

[24] M. S. Shephard, S. Dey, and J. E. Flaherty, "A straight forward structure to construct shape functions for variable p-order meshes", *Computer Methods in Applied Mechanics and Engineering*, **147** (1997) 209–233.

[25] G. Comte-Bellot and S. Corrsin, "Simple Eulerian time-correlation of full and narrow-band velocity signals in grid-generated 'isotropic' turbulence", *Journal of Fluid Mechanics*, **48** (1971) 273–337.

[26] J. Kim, P. Moin, and R. Moser, "Turbulence statistics in fully developed channel flow at low Reynolds number", *Journal of Fluid Mechanics*, **177** (1987) 133.

[27] D. Coles and A. J. Wadcock, "A flying–hot–wire study of two–dimensional mean flow past an NACA 4412 airfoil at maximum lift", *AIAA Journal*, **17** (1979) 321.

[28] R. C. Hastings and B. R. Williams, "Studies of the flow field near a NACA 4412 aerofoil at nearly maximum lift", *Aero J.*, **91** (1987) 29.

[29] A. J. Wadcock, "Investigation of low–speed turbulent separated flow around airfoils", Technical Report 177450, NACA CR, 1987.

[30] D. R. Chapman, "Computational aerodynamics development and outlook", *AIAA Journal*, **17** (1979) 1293.

[31] M. W. Beall, H. L. de Cougny, S. Dey, R. Garimella, R. M. Obara, and M. S. Shephard, "Meshing environment for geometry-based analysis", Technical Report SCOREC #17-1997, Scientific Computation Research Center, RPI, Troy, NY, 1997.

[32] I. Babuška and A. K. Aziz, "On the angle condition in the finite element method", *SIAM Journal of Numerical Analysis*, **13** (1976).

[33] S. Dey, *Geometry-based three-dimensional hp-finite element modelling and computations*, Ph.D. thesis, Rensselaer Polytechnic Institute, 1997.