Preprint
UCRL-JC-144973

Detached Eddy Simulations of Incompressible Turbulent Flows Using the Finite Element Method

G. Laskowski

This article was submitted to
54th Annual Meeting of the American Physical Society Division of Fluid Dynamics, San Diego, CA,, November 18-20, 2001

U.S. Department of Energy

Lawrence Livermore National Laboratory

August 1, 2001

This work was performed under the auspices of the United States Department of Energy by the University of California, Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

Detached Eddy Simulations of Incompressible Turbulent Flows Using the Finite Element Method

Greg Laskowski[1]

New Technologies Engineering Division
Lawerence Livermore National Laboratory
Livermore, CA 94550

Abstract

An explicit Galerkin finite-element formulation of the Spalart-Allmaras (SA) 1-equation turbulent transport model was implemented into the incompressible flow module of a parallel, multi-domain, Galerkin finite-element, multi-physics code, using both a RANS formulation and a DES formulation. DES is a new technique for simulating/modeling turbulence using a hybrid RANS/LES formulation. The turbulent viscosity is constructed from an intermediate viscosity obtained from the transport equation which is spatially discretized using Q1 elements and integrated in time via forward Euler time integration. Three simulations of plane channel flow on a RANS-type grid, using different turbulence models, were conducted in order to validate the implementation of the SA model: SA-RANS, SA-DES and Smagorinksy (without wall correction). Very good agreement was observed between the SA-RANS results and theory, namely the Log Law of the Wall (LLW), especially in the viscous sublayer region and, to a lesser extent, in the log-layer region. The results obtained using the SA-DES model did not agree as well with the LLW, and it is believed that this poor agreement can be attributed to using a DES model on a RANS grid, namely using an incorrect length-scale. It was observed that near the wall, the SA-DES model acted as an RANS model, and away from the wall it acted as an LES model.

INTRODUCTION

The turbulent motion of fluids is a three-dimensional, unsteady physical phenomena observed at high Reynolds numbers (Re), the correct and accurate prediction of which is required for accurate drag and heat transfer, and particle dispersion predictions. Most internal and external aerodynamic problems of interest are at Reynolds numbers high enough to be either transitional or fully turbulent, and thus must be accounted for when developing a Computational Fluid Dynamics (CFD) package, and when conducting CFD simulations. However, this can prove to be a frustrating task, and as pointed out by Moin (1998) this "frustration results largely from the mixture of chaos and order and the wide range of time scales that turbulent flows possess".

[1] Summer Intern 00/01, Lawerence Livermore National Laboratory, Dept. of Mechanical Engineering PhD Candidate, Stanford University, Dept. of Aeronautics and Astronautics

Turbulence consists of a continuous spectrum of scales ranging form large scales, on the order of the geometry of the flow, down to the smallest scale, known as the Kolmogorov scale. Whereas the large scales are largely geometry, or flow dependant, the small scales are very nearly homogenous. Kolmogorov introduced the idea of an energy cascade which assumes that the turbulent energy produced by the large turbulent structures is transferred via the intermediate eddies towards the small eddies where this energy is dissipated by viscous effects. The chief difficulty in simulating and/or modeling turbulence is the fact that the scales have a strong non-linear dependence on the Reynolds number of the flow.

In the present investigation, a relatively new approach known in the literature as DES is investigated in order to more accurately simulate and model turbulence. Detached Eddy Simulation (DES) is similar to Large Eddy Simulation (LES) in that it is a modeling/simulation technique whereby the large, non-homogenius, energy-carrying scales are solved for directly on the computational grid while the small, homogenius, primarily dissipative scales are modeled. This is accomplished through the use of a model for an artificial viscosity used to represent the effect of the unresolved scales on the flow. Whereas LES uses a simple algebraic model, which has a tendency to break down near solid walls, to compute the SGS viscosity, DES uses a more complex 1-equation transport model which has been developed to more accurately predict the turbulent behavior of fluids near such walls. An explicit Galerkin finite-element formulation of the Spalart-Allmaras model was implemented into the incompressible flow module of a parallel, multi-domain, Galerkin finite-element, multi-physics code (Dunn, 2000), using both a RANS formulation and a DES formulation.

One of the most frequently investigated problems in the literature for turbulence modeling and simulation is the flow between two infinite parallel plates. The reasons are quite apparent upon inspection. While the geometry is simple, the flow is not. In the streamwise and transverse directions the turbulence is homogenous. However, in the wall normal direction this is not the case. The presence of the walls has a profound impact on the turbulence. In the present investigation, channel flow simulations were conducted for plane channel flow at shear Reynolds number $Re_\tau = 200$, where:

$$Re_\tau = \frac{u_\tau \delta}{\nu}$$

$u_\tau = \sqrt{\frac{\tau_w}{\rho}}$ is the friction velocity, τ_w is the wall shear stress, δ is the channel half-width, ν is the kinematic viscosity, and ρ is the fluid density.

GOVERNING EQUATIONS AND TURBULENCE MODELING

The governing equations of incompressible flows are the Navier-Stokes equations, which take the (PDE) form:

$$\frac{\partial u_i}{\partial x_i} = 0$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{\mathrm{Re}} \frac{\partial^2 u_i}{\partial x_i \partial x_j} + f_i$$

where use is made of the Einstein summation convention and u_i is the velocity vector, $p = P/\rho$ is the modified pressure, the pressure divided by the constant density, and f_i is the external force vector. The equations are written here in nondimensional form in order to demonstrate the effect of the Reynolds number on the overall system. The equations are a set of coupled, non-linear partial differential equations which, along with the correct specification of boundary and initial conditions, govern any fluid in a continuum. It is generally agreed that the equations fully govern turbulent flow as the Knudsen number (ratio of mean free path to a suitable length scale) is much smaller than the Kolmogorov scale of turbulence (Härtel, 2000).

Presently there are three approaches to the simulation and/or modeling of turbulence: Direct Numerical Simulation, Reynolds Averaged Simulations, and Large Eddy Simulations. A brief description of the benefits and limitions is now given.

The first approach, and most accurate, is known as Direct Numerical Simulation (DNS) whereby the Navier-Stokes equations are solved exactly using high-order spatial discretization methods and temporal integration algorithms in a computational domain sufficiently resolved to capture all of the spatial scales. Unfortunately, however, this requires tremendous computing resources. Consider the channel test case mentioned above. For $\mathrm{Re}_\tau = 200$, the number of grid points required is on the order of (Wilcox, 1993):

$$N_{DNS} \cup (3\,\mathrm{Re}_\tau)^{9/4} \cup 1{,}750{,}000$$

the time step required to accurately represent the temporal scales is on the order of:

$$\Delta t \cup \frac{0.006}{\sqrt{\mathrm{Re}_\tau}} \frac{\delta}{u_\tau} = \frac{4.25e^{-4}}{u_\tau}$$

and the number of timesteps required to reach a statistical steady state is on the order of 25,000. Thus, for such a "simple" problem, enormous computational resources are required and the problem intensifies as the geometric complexity, and hence Reynolds number, increases.

The second method to simulating/modeling turbulence is to solve the "Reynolds-Averaged Navier-Stokes" equations. The Reynolds Averaged Navier Stokes (RANS) approach is to simulate the effect of the turbulent scales on the large scales of the problem. The velocity in the governing equation is replaced s.t. $u_i \blacklozenge \langle u_i \rangle + u_i'$ where the

<> denotes a temporally averaged quantity, and the prime denotes some perturbation from the mean. The equations are then averaged in time resulting in the RANS equations:

$$\frac{f\langle u_i\rangle}{fx_i} = 0$$

$$\frac{f\langle u_i\rangle}{ft} + \frac{f\langle u_i u_j\rangle}{fx_j} = -\frac{f\langle p\rangle}{fx_i} + \frac{f^2\langle u_i\rangle}{fx_i fx_j} - \frac{f\langle u_i' u_j'\rangle}{fx_j} + \langle f_i\rangle$$

where the additional term on the RHS of the momentum equation is termed the Reynolds stress tensor. A plethora of models exist, in varying degrees of complexity, for the Reynolds stress tensor. The RANS approach is currently the only viable option for simulating flows at high Reynolds numbers. Unfortunately, however, the degree of accuracy of the model can vary sharply for one simple reason: the Reynolds stress tensor is flow dependant, as opposed to a fluid dependant, property. As such, a model that was developed and calibrated for one flow may not be all that accurate when used in another. More to the point, the large-scale turbulent structures that would have existed in a time-dependant simulation are strongly dependant on the geometry and boundary conditions. When these scales are averaged and modeled, the model is only as good as the problem for which the model was originally developed.

The third method used for simulating/modeling turbulence is really a hybrid approach combining the strengths of DNS and RANS while attempting to discard their weaknesses. Large Eddy Simulation (LES) is an approach where one solves for the large, non-homogenous (i.e. geometry dependant) scales and models the small homogenous scales. In order to distinguish between large scales and small scales, the Navier-Stokes equations are filtered:

$$\bar{u}_i(\bar{x}, t) = \int G(\bar{x} - \bar{\xi}; \Delta) u_i(\bar{\xi}, t) d^3\bar{\xi}$$

where G is the filter function. The most common filter function found in the literature is the volume-average box filter:

$$G(\bar{x} - \bar{\xi}; \Delta) = \begin{cases} 1/\Delta^3, & |x_i - \xi_i| < \Delta x_i/2 \\ 0, & \text{otherwise} \end{cases}$$

Applying the filter to the Navier-Stokes equations results in:

$$\frac{f\bar{u}_i}{fx_i} = 0$$

$$\frac{f\bar{u}_i}{ft} + \frac{f\overline{u_i u_j}}{fx_j} = -\frac{f\bar{p}}{fx_i} + \frac{1}{\mathrm{Re}}\frac{f^2\bar{u}_i}{fx_i fx_j} + \bar{f}_i$$

where the convective flux is generally written as:

$$\overline{u_i u_j} = \bar{u}_i \bar{u}_j + L_{ij} + C_{ij} + R_{ij}$$

$$L_{ij} = \overline{\bar{u}_i \bar{u}_j} - \bar{u}_i \bar{u}_j$$

$$C_{ij} = \overline{\bar{u}_i u_j'} + \overline{\bar{u}_j u_i'}$$

$$R_{ij} = \overline{u_i' u_j'}$$

L_{ij}, C_{ij}, and R_{ij} are the Leonard, cross, and Reynolds stresses respectively. The governing equations for LES can be re-written in a form strikingly similar to the RANS equations:

$$\frac{f\bar{u}_i}{fx_i} = 0$$

$$\frac{f\bar{u}_i}{ft} + \frac{f\bar{u}_i \bar{u}_j}{fx_j} = -\frac{f\bar{p}}{fx_i} + \frac{1}{\mathrm{Re}} \frac{f^2 \bar{u}_i}{fx_i fx_j} - \frac{f\tau_{ij}}{fx_j} + \bar{f}_i$$

where

$$\tau_{ij} = \overline{\bar{u}_i \bar{u}_j} - \bar{u}_i \bar{u}_j$$

and, as with RANS, the sub-grid shear stress (SGS) term needs to be modeled. Unlike RANS, however, this model needs only to model the small, primarily dissipative, nearly homogenous scales of turbulence. The most commonly used SGS model is the simple, yet effective Smagorinsky model:

$$\tau_{ij} = -2\nu_T \bar{S}_{ij}$$

where S is the filtered strain rate and eddy viscosity (or SGS viscosity) ν_T is:

$$\nu_T = C\Delta^2 |\bar{S}|$$

where

$$|\bar{S}| = \sqrt{2\bar{S}_{ij}\bar{S}_{ij}}; \quad \bar{S}_{ij} = \frac{1}{2}\left(\frac{f\bar{u}_i}{fx_j} + \frac{f\bar{u}_j}{fx_i}\right)$$

Δ is the cell volume and C is the Smagorinsky constant which is set to 0.11 as determined from analysis of homogeneous turbulence.

The Smagorinsky model is a simple model that works well for homogenous turbulence. However, in wall-bounded flow it is severely limited and breaks down due to

the fact that it doesn't account well for the diminishing length scale near solid walls (Laskowski, 1998). Furthermore, in flows with large separation regions, it has been shown that the accuracy of the model is suspect.

Spalart (2000) has suggested yet another approach to the modeling of turbulence: Detached Eddy Simulation (DES). Whereas LES is really a hybrid of DNS and RANS, DES is a hybrid of LES and RANS. Instead of using the Smagorinsky model as the SGS model, he proposed using the Spalart-Allmaras (SA) 1-equation turbulence model (Spalart, 1992), originally developed for turbulent boundary layers, shear layers, and separated flows. Away from the wall, the model is essentially the Smagorinsky model whereas near the wall, the model has been calibrated to account for both attached and detached flows. The model is a transport model, which solves for the SGS viscosity:

SGS VISCOSITY

$$\nu_{SGS} = \tilde{\nu} f_{v1}$$

TRANSPORT EQUATION

$$\frac{D\tilde{\nu}}{Dt} = c_{b1}\tilde{S}\tilde{\nu} - c_{w1}f_w\left(\frac{\tilde{\nu}}{d}\right)^2 + \frac{1}{\sigma}\frac{f}{fx_k}(\nu+\tilde{\nu})\frac{f\tilde{\nu}}{fx_k} + \frac{c_{b2}}{\sigma}\frac{f\tilde{\nu}}{fx_k}\frac{f\tilde{\nu}}{fx_k}$$

ADDITIONAL RELATIONS

$$f_{v1} = \frac{X^3}{X^3 + c_{v1}^3}; \quad f_{v2} = 1 - \frac{X}{1 + Xf_{v1}}; \quad f_w = g\left(\frac{1+c_{w3}^6}{g^6 + c_{w3}^6}\right)^{1/6}$$

$$X = \frac{\tilde{\nu}}{\nu}; \quad g = r + c_{w2}\left(r^6 - r\right) \quad r = \frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}; \quad \tilde{S} = S + \frac{\tilde{\nu}}{\kappa^2 d^2}f_{v2}; \quad S = \sqrt{2\Omega_{ij}\Omega_{ij}}; \quad \Omega_{ij} = \frac{1}{2}\left(\frac{fU_i}{fx_j} - \frac{fU_j}{fx_i}\right)$$

CLOSURE COEFFICIENTS

$$c_{b1} = 0.1355; \quad c_{b2} = 0.622; \quad c_{v1} = 7.1; \quad \sigma = 2/3;$$
$$c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{(1+c_{b2})}{\sigma}; \quad c_{w2} = 0.3; \quad c_{w3} = 2; \quad \kappa = 0.41$$

Finally, in the above expressions 'd' is the length scale, which for a RANS simulation is the minimum distance to a wall, and for DES is:

$$d = \min\left(C_{DES}\Delta^{1/3}, d_{wall}\right)$$

where C_{DES} is a constant and is usually set to 0.61.

NUMERICAL FORMULATION

ALE3D is an unstructured, multi-domain, multi-physics finite element code. The incompressible flow module uses the Galerkin Finite-Element Method (FEM) as presented in Gresho (1984). Prior to this effort, the governing equations were integrated in time using the first-order forward Euler scheme and the turbulence was simulated using the Smagorinsky model to compute the SGS stress without a wall correction term (i.e. van Driest). In order to implement the SA model, the transport equation for the eddy viscosity was first discretized using a finite-element approach similar to that used for the momentum transport equation. The residual function is formed and multiplied by an appropriate test function, which for the GFEM method is the basis function. The resulting expression is then integrated by parts to eliminate any second order derivatives thus arriving at the weak form:

$$\int_\Omega \phi_i \frac{f\tilde{\nu}}{ft} + \int_\Omega \phi_i (\bar{V} ? \tilde{\nu}) =$$

$$\int_\Omega \phi_i c_{b1} \tilde{S}\tilde{\nu} - \int_\Omega \phi_i c_{w1} f_w \left(\frac{\tilde{\nu}}{d}\right)^2 + \int_\Omega \phi_i \frac{c_{b2}}{\sigma} \frac{f\tilde{\nu}}{fx_k} \frac{f\tilde{\nu}}{fx_k} - \int_\Omega \frac{1}{\sigma} (\nu + \tilde{\nu}) \frac{f\tilde{\nu}}{fx_k} \frac{f\phi_i}{fx_k} + \int_\Gamma \frac{1}{\sigma} \hat{n}\phi_i (\nu + \tilde{\nu}) \frac{f\tilde{\nu}}{fx_k}$$

where the last term is the natural boundary condition. The source terms on the RHS are evaluated at the element level whereas the transport variable in the advection and diffusion term are replaced with the expansion:

$$\tilde{\nu} = \sum_{j=1}^{8} \nu_j \phi_j$$

to yield:

$$\dot{\tilde{\nu}} \int_\Omega \phi_i \phi_j + \bar{V}_k \tilde{\nu}_j \int_\Omega \phi_i \frac{f\phi_j}{fx_k} + \frac{\tilde{\nu}_j}{\sigma} (\nu + \tilde{\nu}) \int_\Omega \frac{f\phi_i}{fx_k} \frac{f\phi_j}{fx_k} = S_1 + S_2 + S_3$$

where S_1, S_2, and S_3 are the three source terms. The above expression is integrated in time using first order forward Euler, and the expression can be recast in terms of element matrices as:

$$M_{ij}\tilde{\nu}_j^{n+1} = M_{ij}\tilde{\nu}_j^n + \Delta t \left[N_{ij}\tilde{\nu}_j^n - K_{ij}\tilde{\nu}_j^n + S_1 + S_2 + S_3 \right]$$

Finally, multiplying through by the inverse of the (lumped) mass matrix, M_{ij}, the value of 'nutilda' at each node for a single element takes the form:

$$\tilde{\nu}_j^{n+1} = \tilde{\nu}_j^n + \Delta t\left[M_{ij}^{-1}N_{ij}\tilde{\nu}_j^n - M_{ij}^{-1}K_{ij}\tilde{\nu}_j^n + M_{ij}^{-1}(S_1 + S_2 + S_3)\right]$$

Now, the SGS viscosity is an element quantity, as opposed to a nodal quantity like 'nutilda'. In order to assemble the global value of 'nutilda' the contribution from the source terms at the element level had to be averaged in such a way that the corresponding contribution from the RHS (i.e. source terms) was specified at each node. This was accomplished with a simple arithmetic average, with no weighting from the neighboring element volumes. Once the source terms are changed from being an element quantity to a nodal quantity, the global nodal value of 'nutilda' is assembled, and updated in time. Since we are using hexahedral elements, the element value of the SGS viscosity is then found from another arithmetic average:

$$\nu_{SGS_{element\ j}} = \frac{1}{N}\sum_{i=1}^{N}\tilde{\nu}_{i,element\ j}; \quad N = 8$$

Initialization and Boundary Conditions

Since the governing equation for the SGS viscosity is a first-order (in time) PDE for 'nutilda', an initial condition is required. This proved to be somewhat challenging in that 'nutilda' is a non-linear function of the SGS viscosity, namely:

$$\nu_{SGS} = \tilde{\nu}\frac{(\tilde{\nu}/\nu)^3}{(\tilde{\nu}/\nu)^3 + c_{v1}^3}$$

Ideally the algebraic Smagorinsky model would be used for several hundred iterations in order to determine a good initial guess for the SGS viscosity from which the initial value of 'nutilda' could be determined. This requires the use of a Newton solver, which was not implemented due to time restrictions. Instead, the value of nutilda was initialized to the molecular viscosity, which, while successful, is ultimately a poor choice requiring far too many iterations to arrive at a realistic value since, typically, $\nu << \nu_{SGS}$ (except, of course, in the viscous sublayer).

Only two boundary conditions for 'nutilda' were implemented, namely: no-slip, where the turbulent viscosity is set to zero along walls, and natural boundary conditions. The later is quite questionable as it is extremely desirable to specify inflow values of turbulence in order to conduct realistic numerical simulations. While the natural boundary condition seems to function at inflows, it might be advantageous to implement an inflow boundary condition to fix the value of the SGS viscosity, and thus 'nutilda' at an inflow! For example, it would not be possible to investigate the effect of freestream turbulence intensity on the skin friction for turbulent flow past a flat plate!

Time Step Control

Since the formulation outlined early is an explicit formulation, the SGS model has its' own stability limits for Δt. Unfortunately, however, the stability limits are not clear due to the extreme non-linearity of the model. In the current investigation, it is assumed that the Δt computed from the stability limit of the Navier-Stokes equations is sufficiently small to ensure stability of the SGS model. This is definitely not a hard and fast rule, and this assumption is extremely suspect. One possible way in determining the time step is to set (Wilcox, 1993, page 297):

$$\Delta t \leq \frac{CFL_{SGS}}{|S_1| + |S_2| + |S_3|}$$

It should be noted however, that this stability restriction is an absolute restriction and is, in fact, far too restrictive. The actual Δt required for the SGS model to be stable probably lies somewhere in between the value computed for the Navier-Stokes stability limit and the limit just described.

TIME INTEGRATION OF THE NAVIER-STOKES EQUATIONS

As mentioned previously, in order to conduct unsteady DES simulations several ingredients are necessary:

1. Accurate spatial resolution (i.e. grid, algorithm)...at least 2nd order
2. Accurate temporal integration...at least 2nd order accurate
3. Robust SGS model

Having focused much attention on requirement number 3, the integration of the Navier-Stokes equations was investigated. The original version of ALE3D integrated in time using the 1st order forward Euler method:

$$u_j^{n+1} = u_j^n + \Delta t\left(RHS^n\right)$$

This method, however, is not sufficiently accurate for temporal integration and therefore the 2nd order AB2 method was implemented (Gresho, 2000):

$$u_j^{n+1} = u_j^n + \frac{\Delta t}{2}\left(3RHS^n - RHS^{n-1}\right)$$

RESULTS

Results are presented for plane channel flow with $Re_\tau = 200$. The geometry and flow conditions were taken from Piomelli (1993) and Laskowski (1998). However, in order to validate the implementation of the SA model, a scaled down 'RANS' grid of the channel was constructed and investigated. Table 1 presents the geometry and grid parameters used for the simulation validations case.

Table 1. Validation Case for Plane Channel Flow Simulations

Lx	Ly	Lz	Nx	Ny	Nz	Δx^+	Δy^+_{min}	Δz^+
4π	2	π	60	80	1	41.9	0.5	628.3

Figure 1 shows the grid in the xy plane. Note that the grid lines are clustered towards the walls in order to more accurately predict the regions of strong gradients.

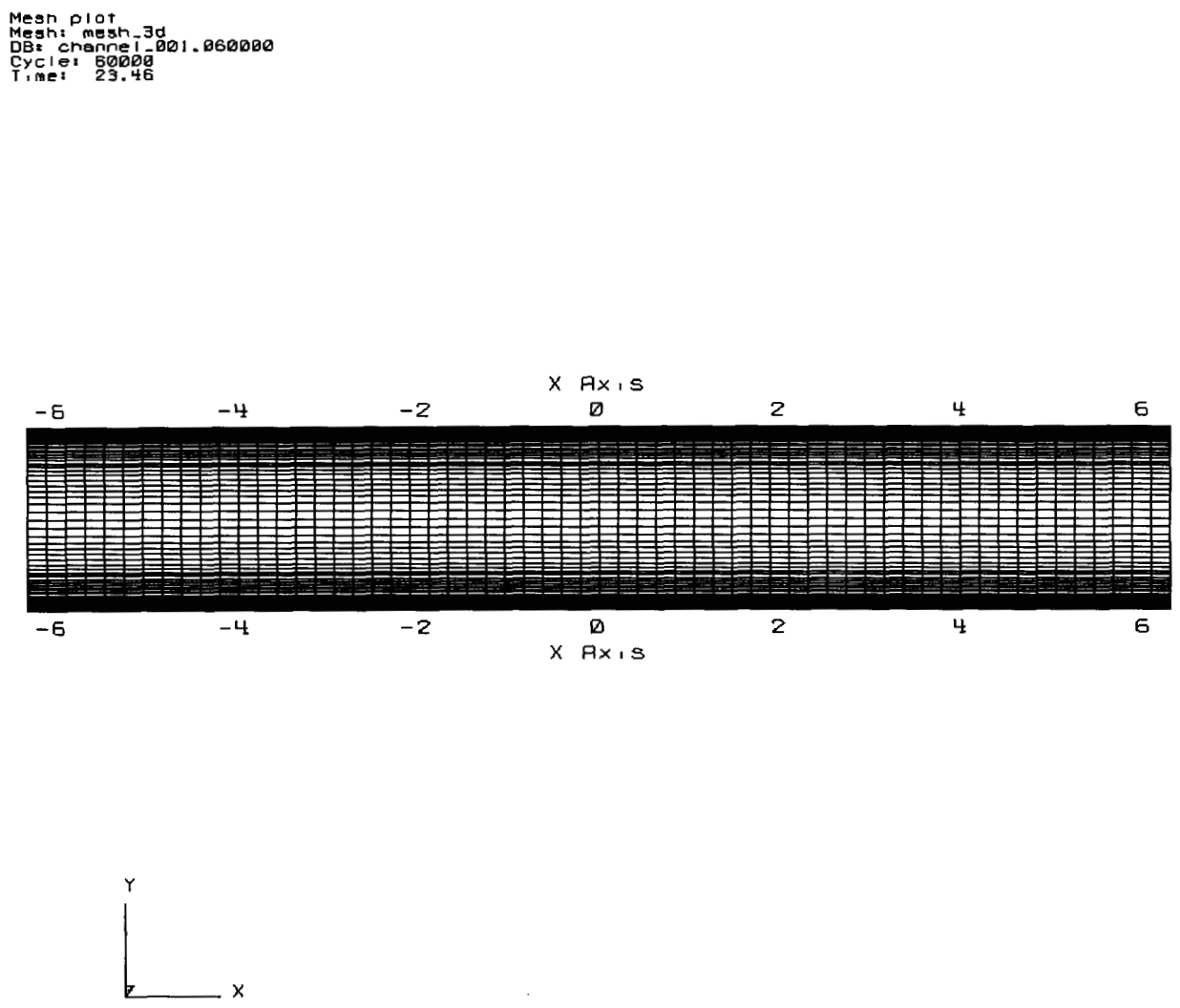

Figure 1. Computational grid in the xy plane.

A parabolic velocity profile, with a maximum value of 20, was specified at the inflow, whereas a zero natural boundary condition was set at outflow. No slip walls were

set at the upper and lower walls, and the symmetry boundary condition was invoked for the single element out of plane.

Three separate cases were conducted in order to investigate the effect of the turbulence model on the overall flow-field: 1) Smagorinsky, 2) SA-RANS, 3) SA-DES. Each case uses the same grid and the same initial and boundary conditions. Thus, any differences seen in the results can be attributed to the turbulence model.

Figure 2 depicts the eddy viscosity for cases 1, 2 and 3. The first thing to note is that while the turbulent viscosity predicted by the SA model goes to zero near the wall, as desired, the value predicted by the Smagorinsky does not. Thus, there really is no viscous sub-layer for case 1. Furthermore, it is interesting to note that the SA-RANS model predicts a maximum value of ν_T which is 6 times that of the Smagorinsky predictions. This might be attributed to the fact that the Smagorinsky model is primarily used to dissipate the small scales of turbulence whereas the grid being used is essentially a RANS grid thus resulting in a lower predicted value of the SGS viscosity. The length-scale which is used for the RANS simulation is correct ($d = d_{wall}$) whereas the length scale being used by the Smagorinsky model is not ($d = C_{DES}\Delta^{1/3}$). Finally, note that the DES and Smagorinsky results agree well in the core of the flow where the DES model is using the LES length scale, and thus the model breaks down to a Smagorinsky-type model.

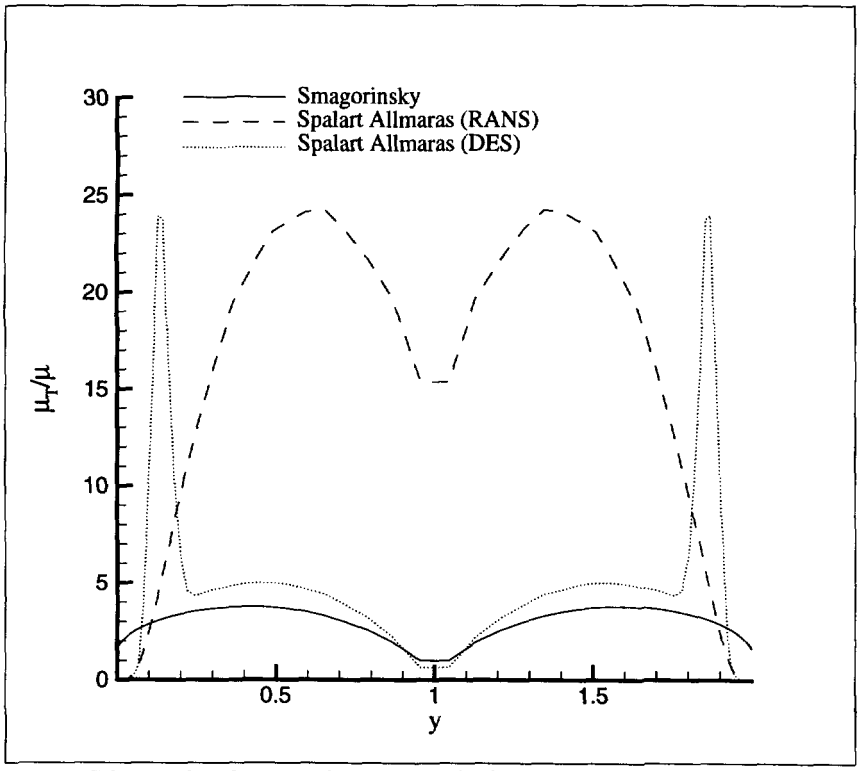

Figure 2. Comparison of turbulent viscosity profiles.

Figure 3 presents the velocity profiles in global coordinates. It is immediately evident that the SA-RANS simulation predicts a much fuller velocity profile than does either the SA-DES simulation or the Smagorinsky simulation. Also, the point where the SA-DES model switches from RANS to LES mode is evident around y = 0.1 and y = 1.9. In order to determine if the predicted shear stress is correct, the velocity profile will be plotted against the Log-Law of the Wall.

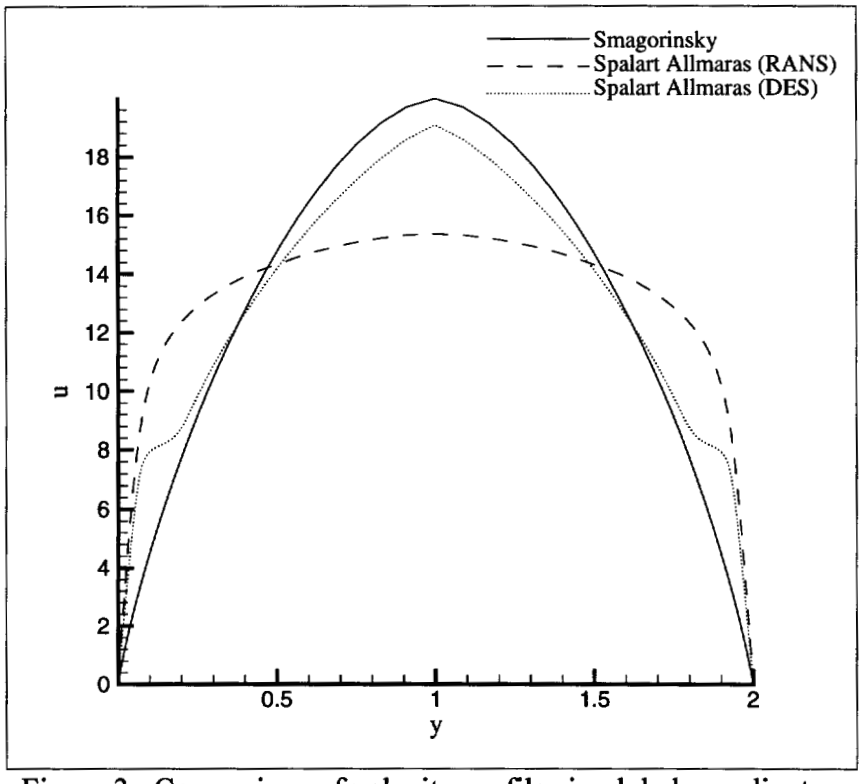

Figure 3. Comparison of velocity profiles in global coordinates.

Figures 4 and 5 present the sternest test yet for the model implementation: comparison against theory. The exact profile for channel flow is known and termed the 'Log Law of the Wall' (LLW) which consists of 3 regions: the viscous sub-layer $0 < y^+ < 10$:

$$u^+ = y^+; \quad u^+ = \frac{u}{u_\tau}, \quad y^+ = \frac{u_\tau y}{\nu}$$

the buffer layer $10 < y^+ < 15$, and the log-layer $15 < y^+ < \delta$:

$$u^+ = \frac{1}{\kappa}\ln(y^+) + B$$

where κ is the von Karman constant, 0.41 and B is the van Driest constant, 5.1-5.5, both of which were empirically determined. It should be noted that the above cutoffs for the different turbulent regions are hard and fast.

Referring to Figures 4 it can be seen that the SA-RANS model is in very good agreement with the LLW solution. The viscous sublayer is in excellent agreement with theory. In the buffer layer, it is evident that the model is transitioning from the viscous sublayer to the log-layer, albeit at a more gradual rate than predicted by theory. Futhermore, it is evident that while the SA-RANS simulation results in a log-layer with the correct slope (i.e. the von Karman constant, κ), the y-intercept (i.e B) is over predicted by 4-10% (depending on the value of B).

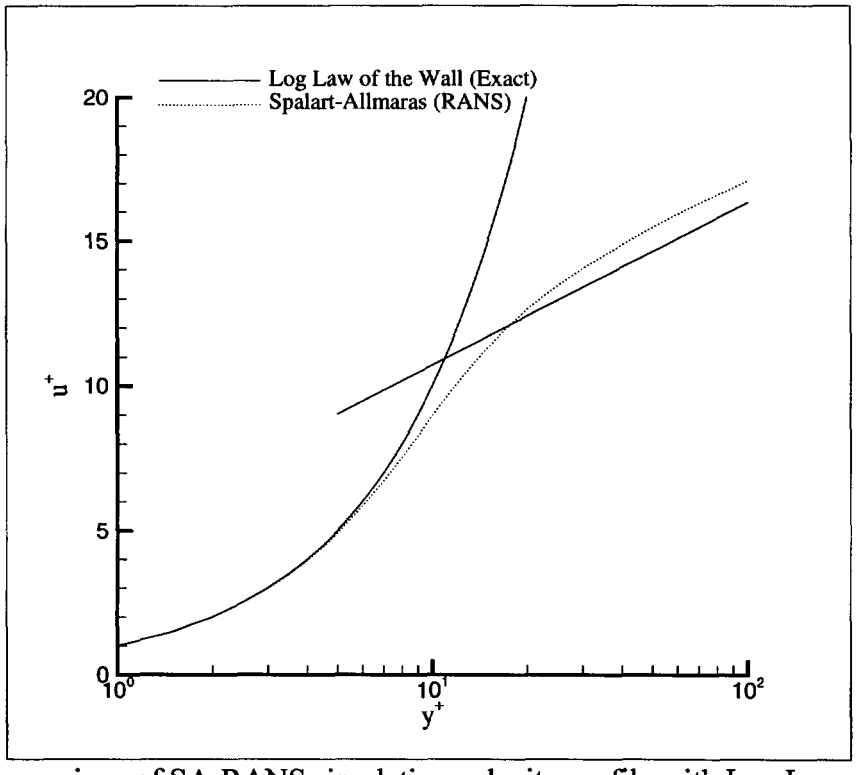

Figure 4. Comparison of SA-RANS simulation velocity profile with Log Law of the Wall, κ= 0.41 and B = 5.1.

Figure 5 presents the results of all three cases when compared to a curve fit for the LLW. This figure is also is a good indicator of where the SA-DES model switches from RANS mode to DES mode. The SA-DES model is in good agreement near the wall, where it is using the correct length scale, but is in poor agreement in the log-layer where it is using the wrong length scale. However, it is interesting to note that the agreement between the Smagorinsky model and the SA-DES model in the core of the flow is in excellent agreement, as expected. This can be viewed as an indication that the SA-DES model is ready to be run on a DES grid, as opposed to the RANS grid used in the current investigation. Finally, looking at the Smagorinsky results, it is interesting to note that a viscous sublayer does seem to be evident which never transitions into a log-layer.

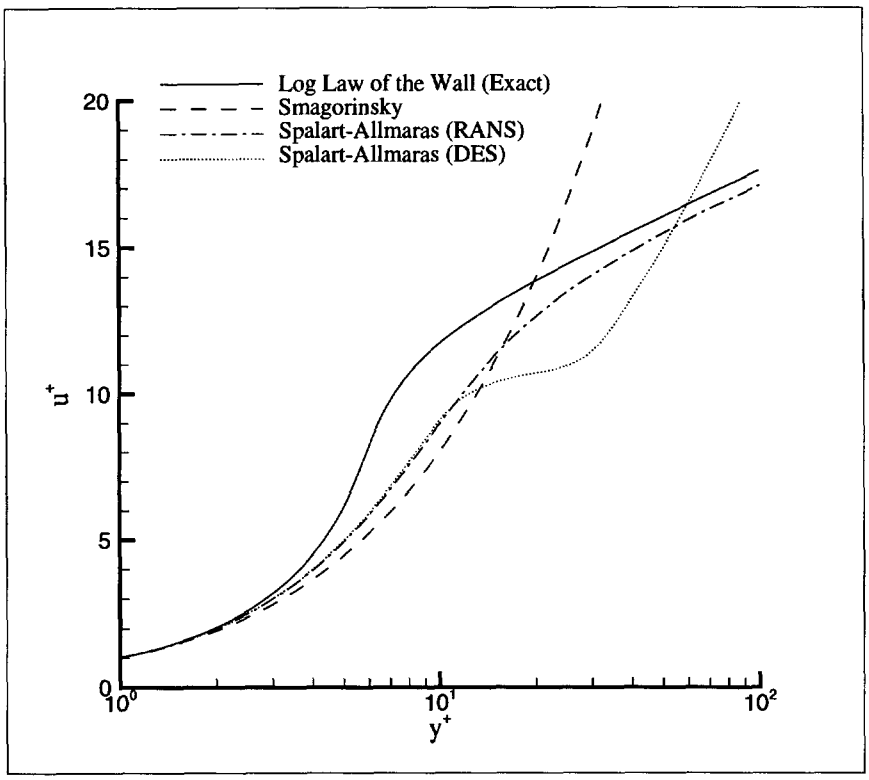

Figure 5. Comparison of Case1, Case2, and Case3 velocity profiles with LLW.

CONCLUSIONS AND RECOMMENDATIONS

An explicit GFEM formulation of the Spalart-Allmaras 1-equation turbulent transport model was implemented in ALE3D as both a RANS model and as a DES model. Simulations of plane channel flow using the model as a RANS model and as a DES model, on a RANS grid were, conducted in order to validate the implementation. Excellent agreement was observed between the SA-RANS results and theory, namely the Log Law of the Wall. The results obtained using the SA-DES model did not agree as well, and it is believed that this poor agreement can be attributed to using a DES model on a RANS grid. In short, the length scale that the model uses in the LES region is incorrect as the grid is far too coarse. However, it should be pointed out that near the wall, the SA-DES model acted as a RANS model, and away from the wall it acted as an LES model, as can be seen by comparing the turbulent viscosity in the core of the flow with results obtained using the Smagorinsky model. The natural next simulation to investigate is DES of channel flow on an LES grid, namely the simulations conducted by Piomelli (1989) and reproduced by Laskowski (1998).

Finally, there are several issues, which must be addressed in both the near and distant future, listed in order of priority in Table 2.

Table 2. Problems with current implementation and suggested fixes.

Problem	Possible Solution
SA model incapable of restarting.	Bug in code. Find and fix.
Element→Node source term treatment treated with algebraic average	Should be weighted based on element volumes comprising a node. In code...commented out, but buggy.
$\tilde{\nu} \blacklozenge \nu$ at first timestep (initialization).	Run Smagorinksy for several iterations and use Newton's method to determine 'nutilda' based on Smagorinsky SGS viscosity.
Stability of model questionable.	Make Δt dependent on stability restriction of SA model. OR Go implicit.
Too many time steps required.	Make model/governing equations implicit rather than explicit.

ACKNOWLEDGEMENTS

This work was performed under the auspices of the U.S. Department of Energy by Lawerence Livermore National Laboratory under Contract W-7405-ENG-48. The author would like to thank Dr. Rose McCallen for arranging for this work to be done at LLNL, Tim Dunn for all of his extremely helpful input and advice regarding the implementation of the model in ALE3D, and Rob Neely for his help with certain parallel issues.

REFERENCES

Dunn, T. A, McCallen, R. C. (2000) "Parallel Computations of Natural Convection Flow in a Tall Cavity Using an Explicitly Finite Element Method", UCRL-JC-141027.

Gresho, P., Sani, R., (2000) *Incompressible Flow and the Finite Element Method: Volume 1*, John Wiley and Sons, LTD.

Härtel, T., (2000) *Handbook of Computational Fluid Mechanics*, Ed. Peyret, Academic Press, pp. 283-415.

Laskowski, G., (1998) "Wall Resolved Large Eddy Simulation (LES): A First Step Toward Direct Numerical Simulation (DNS)", VKI PR 1999-0019, 1999.

Moin, P., Mahesh, K., (1998) "Direct Numerical Simulation: A Tool in Turbulence Research," Annu. Rev. Fluid Mech., p. 539-578.

N.V. Nikitin, F. Nicoud, B. Wasistho, K.D. Squires and P.R. Spalart, (2000) "An Approach to Wall Modeling in Large Eddy Simulation", Phys. Fluids 12(8), pp 1629-1632.

Piomelli, U., Ferziger, J., Moin, P., (1989) "New Approximate Boundary Conditions for Large Eddy Simulations of Wall-Bounded Flows," Phys. Fluids A, Vol. 1, No. 6, pp 1061-1068.

Spalart, P. R., Allmaras, S. R., (1992) "A One-Equation Turbulence Model for Aerodynamic Flows", AIAA-92-0439.

Wilcox, D. C., (1993) *Turbulence Modeling for CFD*, DCW Industries Inc., La Cañada, CA, 1993.

APPENDIX

The ALE3D and TRUEGRID input files used for the simulations are attached for the sake of completeness. Note that Case1, Case2 and Case3 use the same two files making sure to switch 'itrb' in channel.in to the appropriate flag.

Channel.in

```
# 2d poiseuille flow


DECOMP
# metis
  partitions 1
END

CONTROL
 stopcycle 60000
  dtinit 1.e-8
  dtmax 1.0
  dtmin 1.e-8
END

OUTPUT
 plotac 0
 dumpcycle 20 1e9
 plotcycle 20 1e9
 numcycdigits 6

 plotvar deleteall
 plotvar add p u1 u2 u3
END

BOUNDARY
 table const_table 0.0 1.0
 spacetable inst 1.0   y 0.0   poly 1.0 0.0 -1.0

 incvel_loadcurve pz1                    0.0   0.0 0.0 1.0   1   table
const_table
 incvel_loadcurve pz2                    0.0   0.0 0.0 1.0   1   table
const_table
 incvel_loadcurve py1                    1.0   0.0 0.0 0.0   0   table
const_table
 incvel_loadcurve py2                    1.0   0.0 0.0 0.0   0   table
const_table
 turbwall py1     1.0 #  0.0 0.0 0.0   0   table const_table
 turbwall py2     1.0 #  0.0 0.0 0.0   0   table const_table
# incvel_loadcurve px1   spacetable inst   1.0   1.0 0.0 0.0   0
table const_table
# inctrac_loadcurve px1       12.56636              table const_table
  incvel_loadcurve ppx1  spacetable inst   1.0  20.0 0.0 0.0 0
table const_table
 incvelocity      nall   spacetable inst                      20.0 0.0
0.0
END
```

```
INCFLOW
 incdivfree 1
 inctime 1
 irke 1
 solverparams
   solverlib hypre
   solver superlu
   schurreduction 1
   outputlevel 1
END

MATERIAL a 1
 incinput rho 1.0 p0 0.0 rnu 0.005 itrb 2 smagc 0.1
END
```

Channel.tgi

```
title truegrid file
ale3d
partmode i

c
c parameters
c
para nx 60;
para ny 80;
para nz 1;

para l 12.566371;
para h 2.0;
para w 3.14159;
para x1 [ -%l/2.] ;
para x2 [ +%l/2.] ;
para y1 [ -%h/2.] ;
para y2 [ +%h/2.] ;
para z1 [ -%w/2.] ;
para z2 [ +%w/2.] ;

c
c create the fluid block
c
c using partmode i
block 1;1;1;
      [ %x1] [ %x2] ;
      [ %y1] [ %y2] ;
      [ %z1] [ %z2] ;

c
c set gridsize
c
mseq i [ %nx-1]
mseq j [ %ny-1]
mseq k [ %nz-1]
das 1 1 1 2 2 2 j 0.0025 0.0025

c
c create nodesets
c
nseti ;;;= nall
nseti -1;;;= px1
nseti -2;;;= px2
nseti ;-1;;= py1
nseti ;-2;;= py2
nseti ;;-1;= pz1
nseti ;;-2;= pz2

nseti -1;;; = ppx1
nseti ;-1;; - ppx1
nseti ;-2;; - ppx1
nseti -2;;; = ppx2
nseti ;-1;; - ppx2
nseti ;-2;; - ppx2
```

```
mate 1
endpart

c
c merge
c
tp 1.e-7
merge

c
c create nodesets
c
nsetc nall nall
nsetc px1 px1
nsetc px2 px2
nsetc py1 py1
nsetc py2 py2
nsetc pz1 pz1
nsetc pz2 pz2

nsetc ppx1 ppx1
nsetc ppx2 ppx2
nsetc ppz1 ppz1
nsetc ppz2 ppz2

write
```

Work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract W-7405-ENG-48.