Parallel Simulation Of Unsteady Turbulent Flames

Suresh Menon

School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332-0150, USA

1. INTRODUCTION:

Time-accurate simulation of turbulent flames in high Reynolds number flows is a challenging task since both fluid dynamics and combustion must be modeled accurately. To numerically simulate this phenomenon, very large computer resources (both time and memory) are required. Although current vector supercomputers are capable of providing adequate resources for simulations of this nature, the high cost and their limited availability, makes practical use of such machines less than satisfactory. At the same time, the explicit time integration algorithms used in unsteady flow simulations often possess a very high degree of parallelism, making them very amenable to efficient implementation on large-scale parallel computers. Under these circumstances, distributed memory parallel computers offer an excellent near-term solution for greatly increased computational speed and memory, at a cost that may render the unsteady simulations of the type discussed above more feasible and affordable.This paper discusses the study of unsteady turbulent flames using a simulation algorithm that is capable of retaining high parallel efficiency on distributed memory parallel architectures.

Numerical studies are carried out using large-eddy simulation (LES). In LES, the scales larger than the grid are computed using a time- and space-accurate scheme, while the unresolved small scales are modeled using eddy viscosity based subgrid models. This is acceptable for the momentum/energy closure since the small scales primarily provide a dissipative mechanism for the energy transferred from the large scales. However, for combustion to occur, the species must first undergo mixing at the small scales and then come into molecular contact. Therefore, global models cannot be used. Recently, a new model for turbulent combustion was developed [1,2], in which the combustion is modeled within the subgrid (small-scales) using a methodology that simulates the mixing and the molecular transport and the chemical kinetics within each LES grid cell. Finite-rate kinetics can be included without any closure and this approach actually provides a means to predict the turbulent rates and the turbulent flame speed.

The subgrid combustion model requires resolution of the local time scales associated with small-scale mixing, molecular diffusion and chemical kinetics and, therefore, within each grid cell, a significant amount of computations must be carried out before the large-scale (LES resolved) effects are incorporated. Therefore, this approach is uniquely suited for parallel processing and has been implemented on various systems such as: Intel Paragon, IBM SP-2, Cray T3D and SGI Power Challenge (PC) using the system independent Message Passing Interface (MPI) compiler. In this paper, timing data on these machines is reported along with some characteristic results.

2. THE SIMULATION MODEL

The formulation of the simulation model is briefly summarized below. More details are given elsewhere [1-4] and, therefore, avoided here for brevity.

2.1 Fluid Dynamics:

The fluid dynamic LES equations are obtained by spatially filtering the compressible Navier Stokes equations. The resulting equations contain unknown subgrid terms such as the stresses in the momentum equations: $\tau_{ij}^{sgs} = \bar{\rho}[\widetilde{u_i u_j} - \tilde{u}_i \tilde{u}_j]$, the energy flux and the viscous terms in the energy equation: $H_i^{sgs} = \bar{\rho}[\widetilde{E u_i} - \tilde{E}\tilde{u}_i] - [\overline{p u_i} - \bar{p}\tilde{u}_i]$, $\sigma_{ij}^{sgs} = [\overline{u_i \tau_{ij}} - \tilde{u}_i \bar{\tau}_{ij}]$ and the scalar correlations: $\psi_k^{sgs} = \widetilde{TY}_k - \tilde{T}\tilde{Y}_k$ in the equation of state. Here, tilde and bar indicate filtered variables, $\tilde{u}_i$, $\bar{\rho}$, $\bar{\tau}_{ij}$ and $\tilde{E}$ are, respectively, the resolved velocity components, the density, the viscous stresses and the total energy per unit volume. To close these subgrid terms, a model equation for the subgrid kinetic energy $k = \widetilde{u_i u_i} - \tilde{u}_i \tilde{u}_i$:

$$(\bar{\rho}k)_t + (\bar{\rho}\tilde{u}_i k)_i = C_k \tau_{ij}^{sgs} (\tilde{u}_i)_j - C_\varepsilon \bar{\rho} k^{3/2}/\Delta_g + (\bar{\rho}\nu_t (k)_i)_i \qquad (1)$$

is also solved along with the LES equations. Here, $(\phi)_t$ and $(\phi)_i$ indicate differentiation with respect to time and space, respectively. Also, $\nu_t = C_\nu k^{1/2} \Delta_g$ is the subgrid eddy viscosity and Δ_g is the grid size. The three constants in Eq. (1) are: $C_k = 1.0$, $C_\varepsilon = 0.916$, $C_\nu = 0.0854$; however, in the dynamic model [3,4], the coefficients C_ε and C_ν are computed locally (in space and time) during the simulation. With k known, the subgrid terms are: $\tau_{ij}^{sgs} = -2\bar{\rho}\nu_t (S_{ij} - S_{kk}\delta_{ij}/3)$, $H_i^{sgs} = -\bar{\rho}\nu_t (H)_i / Pt$, and $\sigma_{ij}^{sgs} = u_j \tau_{ij}^{sgs}$. Here, $Pr \approx 1$ is the turbulent Prandtl number, S_{ij} is the resolved rate-of-strain tensor, $S_{ij} = [(\tilde{u}_i)_j + (\tilde{u}_j)_i]/2$, and H is the resolved total enthalpy. Here, all third-order subgrid correlations are neglected in this closure for simplicity. More details are given elsewhere [2-4].

2.2 Combustion Models

In general, simulation of reacting flows (both premixed and non-premixed) requires the solution of the species conservation equations. The LES species equations can be written as:

$$(\bar{\rho}\tilde{Y}_k)_t + (\bar{\rho}\tilde{u}_i \tilde{Y}_k)_i + (\bar{\rho} S_{ik}^{sgs})_i = (\bar{\rho}\widetilde{D_k (Y_k)_i})_i + \bar{\dot{\omega}}_k \qquad (2)$$

where, Y_k and D_k are, respectively, the kth specie mass fraction and the diffusion coefficient. Closure for the species-velocity correlation $S_{ik}^{sgs} = \bar{\rho}(\widetilde{u_i Y_k} - \tilde{u}_i \tilde{Y}_k)$, the production term $\bar{\dot{\omega}}_k$ and the scalar correlations Ψ_k^{sgs} is more problematic, since, to estimate these terms, small-scale turbulent mixing and species molecular diffusion must be modeled correctly. Conventional subgrid closure cannot handle these features and, therefore, the new subgrid combustion model (discussed below) was developed to address this limitation.

For conventional LES simulations of turbulent premixed combustion, a model which circumvents the above noted closure problem can be used for simulations of thin flames (i.e., for flames thinner than the smallest eddy). For these cases, the flame is modeled as a propagating surface and a variable G is defined that is governed by the equation: $(G)_t + \vec{u} \bullet \nabla G = -u_F|\nabla G|$, where u_F is the local propagation speed. This equation describes the convection of a level surface, described by G = constant, by the velocity $\vec{u}$ while undergoing propagation normal to itself at a speed u_F. G=1 denotes the reactant and G=0 the burnt region with the thin flame identified by a level surface in the [0,1] range. The effect of heat release is included in the definition of the specific enthalpy [5]. For turbulent flames, the propagation speed u_F is the turbulent flame speed and is determined using a flame speed model [5]: $u_F/S_L = \exp[u'^2/u_F^2]$. Here, S_L is the laminar flame speed and $u' = \sqrt{2k/3}$ is the local turbulence intensity which is known. This approach avoids the problems associated with the closure of the term $\bar{\dot{\omega}}_k$ since Eq. 2 is not needed for this approach and the effect of chemical kinetics is implicitly included by the specification of S_L.

2.3 Subgrid Combustion Model

In this approach, no scalar equations (Eq. 2) are solved on the LES resolved grid. Rather, within each LES grid cell, the local scalar field is simulated in a 1D domain which is considered a statistical slice through the local three-dimensional flame brush. Resolution within the 1D domain is chosen to resolve the smallest eddies in the flow. Between each LES time step, the scalar reaction-diffusion equations (i.e., Eq. 2 without the convective term) are solved within each LES cell using a finite difference scheme as a local "direct" simulation. Therefore, no closure of the production and diffusion terms is required. Turbulent convection (caused by both small eddies) is simultaneously implemented to include the effects of small-scale turbulent transport (by eddies ranging from the smallest scale to Δ_g) and is simulated using a stochastic (Monte-Carlo type) simulation. The time scales of each of these processes are implemented independently and, therefore, there is direct coupling between the small-scale turbulent mixing, molecular diffusion and chemical kinetics. The effect of large-scale transport (caused by the larger LES-resolved eddies) is also included as a transport of the local subgrid scalar fields across LES cell surfaces based on local momentum flux in a manner that ensures conservation of mass. The heat release and volumetric expansion in the subgrid is coupled to the LES calculation by the equation of state and scalar fields $\tilde{Y}_k$ are obtained by filtering the subgrid fields. More details are given elsewhere [1-2] and, therefore, omitted here for brevity.

3. PARALLEL IMPLEMENTATION ISSUES

The technique of data concurrency (i.e., the primary data space is partitioned and distributed among the processors) rather than functional concurrency (i.e., the overall application is decomposed into several distinct parallel computational tasks) was chosen after careful review of the type and degree of parallelism inherent in the numerical algorithm. The data space is partitioned and distributed to the processors so that 1) the distribution of cells to the nodes leads to a nearly balanced load of communication and computation among all nodes, and 2) the inherent spatial data locality of the underlying cell structure is maintained so as to minimize interprocessor communication. The cell partitioning scheme decomposes the 2D (3D) compu-

tational domain into logically congruent, nearly equal-sized rectangles (cubes). Maximum concurrency is extracted to minimize the execution time on a given number of processors. The overheads associated with parallel implementation, such as, (1) load imbalance, (2) inter-processor communication, (3) data dependency delays, (4) arithmetic, and (5) memory, were analyzed. While the first four types of overheads lead to performance degradation, the memory overhead limits the size of the problem that can be executed on a fixed number of processors. In practice, simultaneously minimizing all these overheads is very difficult.

In the present implementation, the partitioning scheme results in each processor performing computations only on the cells that are held by it. For finite-differences, each domain contains extra layers of ghost cells along the processor partitions to allow the exchange of boundary cell data. This exchange is carried out using a few, relatively long messages. As a result, the high cost of latency associated with message passing is minimized, resulting in a reduced communication overhead even though this data exchange results in an increased memory overhead.

The implementation of the subgrid combustion model is relatively straightforward since this model is within the LES cells and requires no inter-cell communication for the local subgrid processes. Inter-processor communication is needed every LES time step to carry out the transport of subgrid scalar field across LES cell surfaces. These messages carry the local scalar information, however, unlike the long messages needed for the fluid dynamics part, these messages are from the nearest neighbor cells and, thus, are relatively short messages.

The current implementation on parallel systems employs double precision (64-bit) arithmetic and is based entirely on Fortran. Performance comparison with and without I/O has been carried out. However, I/O overhead is unavoidable since the data generated on the spatio-temporal evolution of the flow field is needed for analysis. The type, form and frequency of data varies with the problem and, thus, cannot be standardized. In general, the 3D flow field is needed for flow visualization and for restart files. The present approach combines both these requirements by making all processors to write the required data into one file. The location of this file depends on system architecture: on the Paragon, the file can reside on the local file system while on the SP2 it has to reside on one processor. To optimize I/O time on *all* systems, the flow variables from all processors are written one at a time into a temporary buffer array which always resides on one processor and then this array is written to a file. This approach results in one processor writing a large amount of data instead of all processors writing small amounts of data (which was found to cause I/O bottleneck). This I/O implementation works quite well on all systems used here and is considered an optimal compromise to allow flexibility in porting the code (and data) to different systems. In addition, this approach allows the simulation to be restarted on arbitrary number of processors. This capability is very useful when the system is heavily loaded. A disadvantage of this approach is that a large buffer array is needed (on one processor) which again results in an increased memory overhead and limits the memory available for the simulation on each node.

4. RESULTS AND DISCUSSIONS:

Characteristic results for various applications using both 2-D and 3-D codes are described below to highlight the simulation capability. All the results reported here were obtained using a finite volume scheme that is fourth-order accurate in space and second-order accurate in time. Details of the numerical scheme is given elsewhere [2,5].

4.1 Performance data

Figure 1 shows the typical scaling of the axisymmetric and 3D codes on various systems. The direct simulation (DNS) timing is less than the LES timing because the latter solves an additional equation for the subgrid kinetic energy. However, LES are usually performed using much coarser grids and, therefore, are relatively computationally less intensive. On all the systems, these codes show nearly linear scaleup when the number of processors are doubled. In general, among the distributed processing systems, the SP2 is the fastest for all test cases. However, for a fixed grid size, as the number of processors are increased, the scaleup is superior on both T3D and the Paragon. Also shown is some data obtained on the mixed shared/distributed-memory SGI-PC (MIPS 8000/75 MHz) system (using the same code and MPI). Results suggest that for the same number of processors used, the SGI-PC performs the best (and is twice as fast as the SP-2). For comparison, on a single processor Cray C90, a vectorized version of the 3D code executes at around 487 MFLOPS and requires 0.95 sec per iteration (equivalent to 64-processors SP2).

Figures 2a and 2b show the timing data for the combustion model. For a fixed LES resolution, increasing the subgrid cells increases the CPU time. However, the scale-up is still very good and nearly linear (Fig. 2a). Of particular interest to this study is the slow increase in CPU time on a fixed number of processors (Fig. 2b) with increase in the subgrid resolution. Results show that when the subgrid resolution is increased by a factor of 2, the CPU time (on T3D) only goes up by around 30%; when the resolution is increased by a factor of 5, the time increases by around 2.0. This is due to the increase in local computations relative to the communication overheads. The T3D and SP2 timings are quite close while on the C90 the execution time is increased by 78% when the resolution is doubled. This clearly demonstrates that the combustion subgrid model is much more efficient on the parallel systems.

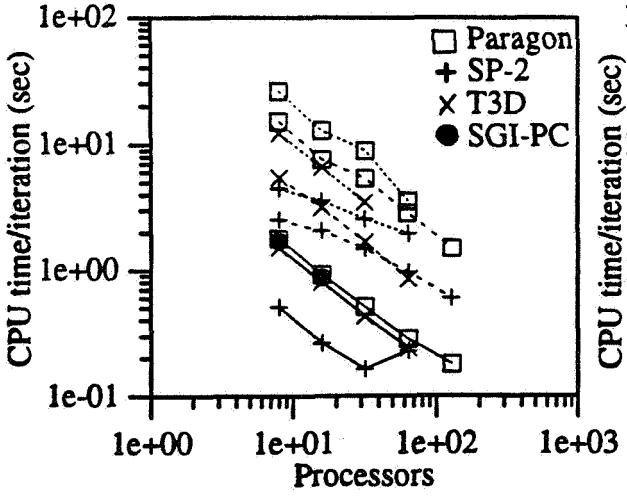


Figure 1. Run time on various platforms. —: 288x64 LES with six reactive scalars, ----: 96x64x64 DNS with one passive scalar, ...: 96x64x64 LES with one passive scalar and solid symbols: 94x64x64 DNS with one scalar.

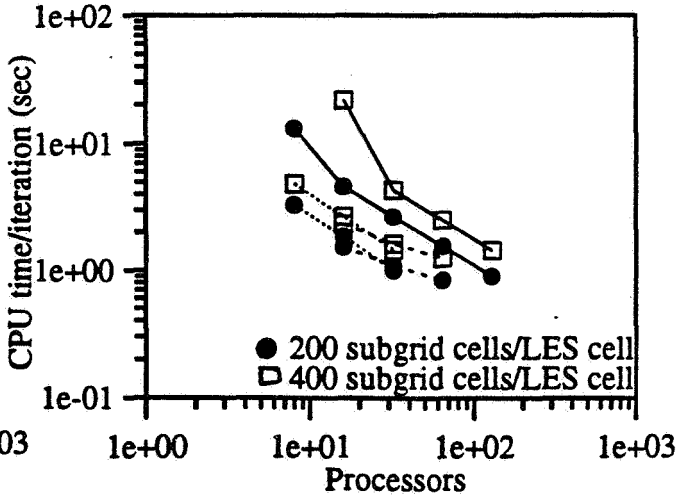


Figure 2a. Run time for the subgrid combustion model on a 32x32x32 LES grid with one passive scalar. —: Paragon, ----: SP-2, ...: T3D

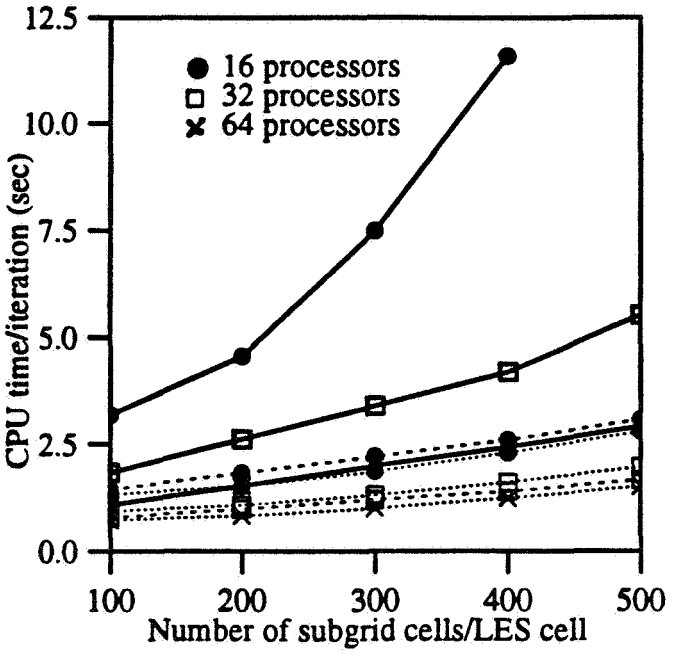


Figure 2b. Performance of the subgrid combustion model in a 32x32x32 LES with one scalar in the subgrid. —: Paragon, ----: T3D, ·····: SP2.

Figure 3. Flame propagation in isotropic turbulence. Contours in the reactants (left of flame) represents vorticity.
Test Conditions: $u'/S_L = 6.3$, $Re = 83$, domain: $2\pi \times 2\pi$, $T_p / T_f = 4$

4.2 Turbulent Premixed Combustion

Simulations of premixed flames have been carried out using a DNS code and the combustion model to validate the new approach. In DNS, all the scales of motion and the flame structure have to be resolved and, thus, only low Reynolds number simulations are possible even on parallel systems due to the memory requirements to resolve all scales. Figure 3 shows a freely propagating turbulent premixed flame in isotropic turbulence simulated on a 400x400 grid with simple single-step finite rate chemistry. The reactants are on the left side of the flame and the burnt products are on the right side. The chemical length and time scales are chosen so that the "thin" flame structure is resolved (using 10 grid points). A typical simulation of around 3-4 large-eddy turnover time required 8 CPU hours on 32 processors (Paragon). The turbulent inflow (on the left side of the flame) contains many vortical structures typical of isotropic turbulence and Fig.3 shows that due to the heat release and gas expansion, a rapid damping of the vortical structures occurs. The initially planar laminar thin flame is wrinkled due to turbulence, thereby, increasing its propagation speed and the overall reactant consumption. These phenomena have been observed in actual experimental studies.

The structure of the premixed flame depends on the competition between the thermal and molecular diffusion and is often characterized by the Lewis number ($Le = \lambda/\rho C_p D$). Figure 4 shows the flame shapes (using a specified mass fraction) at the same time of evolution for various Le. It can be clearly seen that the Le-effect is predominant in regions of high positive curvature (defined concave with respect to the product). Premixed flames are thermally unstable for $Le < 1$ and stable for $Le > 1$. The instability for $Le < 1$ increases the wrinkling of the flame and the turbulent flame speed. This is correctly captured in the DNS and is shown in Fig. 5. Also shown is the prediction by the subgrid combustion model (implemented as a stand-alone model to resolve all the scales, and, therefore, gives a statistically "steady" state value).

4.3 Premixed Flame in Turbulent Couette Flow

Flow between two walls moving in opposite direction (Couette flow) is a classical flow problem and has been studied for a long time. A key characteristic of this flow is that the flow becomes turbulent at relatively low Re and provides a test flow where all length scales are accessible (both experimentally and numerically). If the fluid is a premixed reactant and is ignited, then a turbulent flame ball propagates under the influence of a mean shear (caused by the moving walls). However, since the flow speed is very low, the turbulent advection process is overwhelmed by buoyant convection (under normal gravity). Thus, turbulent combustion in this configuration has to be studied under microgravity and is currently being experimentally studied. Here, some preliminary results obtained using both DNS and LES with the subgrid combustion model are reported which demonstrate not only the characteristic flame structure but also provide an assessment of the subgrid method. Simulations were performed on the SP2, T3D and SGI-PC primarily to take advantage of the availability of these systems.

Figures 6a and 6b show, respectively, the predicted mean and fluctuating velocity fields (obtained by statistically averaging the instantaneous flow fields). Both the LES (using the dynamic subgrid model) and the DNS predictions show reasonable agreement with experimental results. Subsequently, the fluid was modeled as a premixed mixture and ignited at the center. The flame was modeled using the G-equation in the DNS and in the LES (using the subgrid combustion model). No heat release was included for this study but can be included without any problem. Figures 7a and 7b show, respectively, a G=constant level surface representing the flame obtained from DNS (on a 96x64x64 grid) and LES. In LES, the flame is obtained by averaging the subgrid G-fields. Due to the mean shear the flame front is stretched and then torn into two smaller flame balls. Although the LES was performed on a much coarser grid (32x32x32), the results show that the flame ball gets torn at about the same time as in the DNS. Further analysis and LES with heat release are currently underway for further validation.

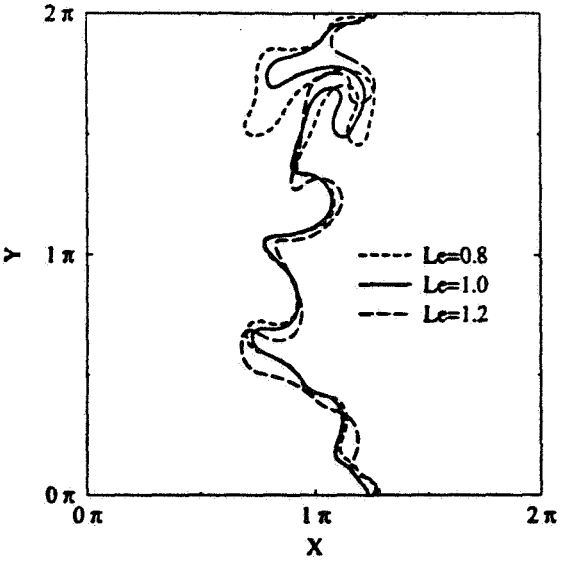


Figure 4. Flame structure for different Lewis numbers at about the same time of evolution (after three large-eddy turnover time). Flame is shown as a fixed mass fraction contour

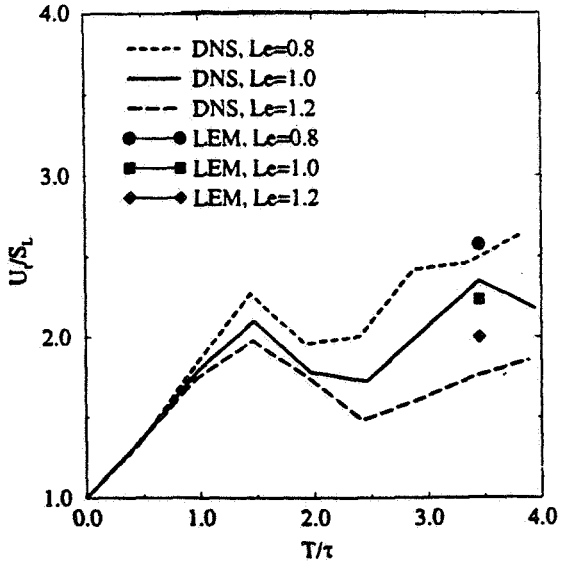
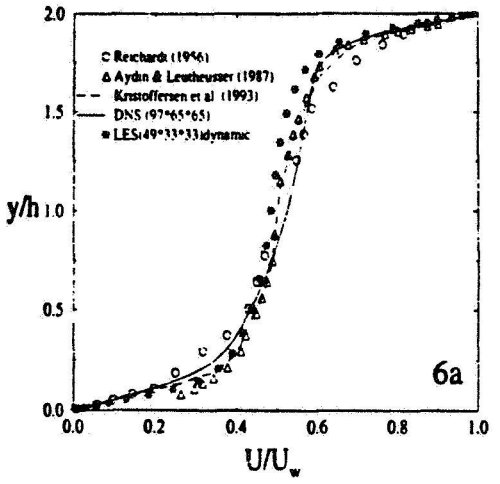
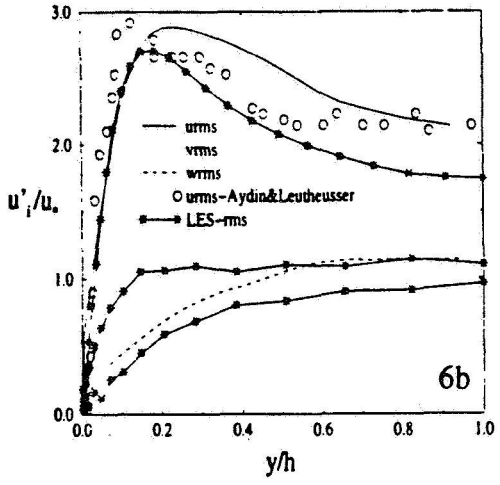
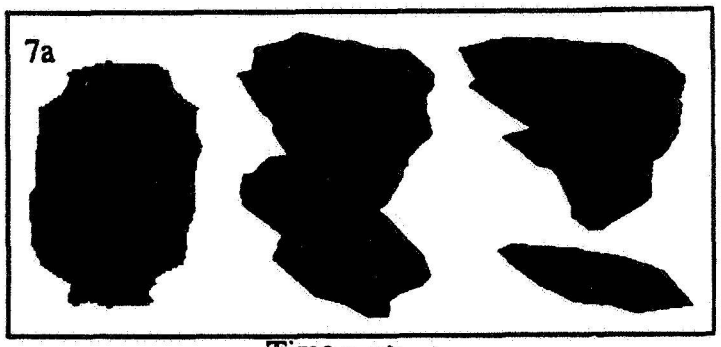


Figure 5. Evolution of turbulent flame speed for different Le. Flame speed is based on area ratio. DNS never reaches a steady state. Also, shown is the "steady" state predicted by the stand-alone combustion model (LEM).







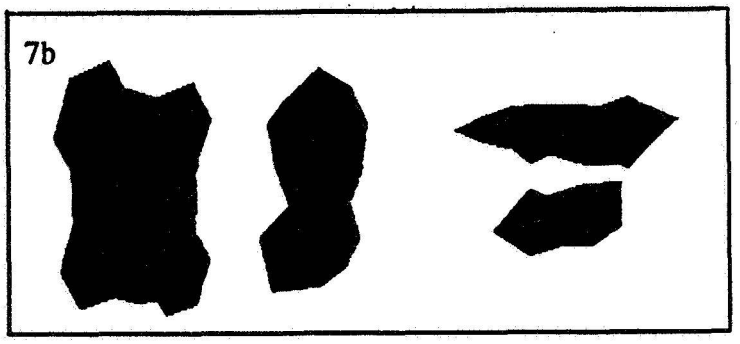


Figure 7. Propagation of a flame ball in Couette flow. Time increases from left to right. (a) G-surface from a DNS simulation, (b) G-surface using subgrid LEM.

Figure 6. Mean and root-mean-square (rms) velocity profiles in a plane Couette flow in a channel of width h. (a) mean velocity normalized by wall friction velocity, (b) rms velocity fluctuation profiles.

5. CONCLUSIONS:

A new LES code has been implemented on parallel systems to study unsteady turbulent flames. This approach models correctly the small-scale turbulent mixing and molecular transport processes. The performance of the codes on parallel systems has been evaluated and it has been demonstrated that the subgrid combustion model is optimum only on parallel systems. Some characteristic results are presented here to demonstrate the ability of the simulation model.

Acknowledgments: This work was supported in part by the Office of Naval Research grants N00014-95-1-0163 and N00014-92-J-4030, and by the NASA Lewis grant NAG3-1610. Results are contributed by graduate students: T. Smith, K. Bhagvan and A. Srinivasan. The computational support was provided by the U.S. DOD HPC Centers at Maui (SP2), Army Research Lab (SGI-PC), Wright Patterson AFB (Paragon), and Eglin AFB (T3D and C90).

References:

[1] Menon, S., McMurtry, P. and Kerstein, A. R., Large Eddy Simulations of Complex Engineering and Geophysical Flows, B. Galperin and S. Orszag (eds.), Cambridge Press (1993).
[2] Calhoon, W. and Menon, S., AIAA Paper No. 96-0516 (1996).
[3] Menon, S. and Kim, S., AIAA Paper No. 96-0425 (1996).
[4] Menon, S. and Jou, W. -H., Combustion Science and Technology, 75 (1991) 53.
[5] Weeratunga, S. and Menon, S., AIAA Paper No. 93-1914 (1993).
[6] Aydin, E. M. and Leutheusser, H. J., ASME Forum on Turbulent Flows, 51 (1987) 51.
[7] Kristoffersen, R., Bech, K. H. and Andersson, H. I., Applied Science Res. 51 (1993) 337.
[8] Reichardt, H., ZAMM Sonderneft (1956) 526.