Parallel Computation of Turbulent Fluid Flows with the Piecewise-Parabolic Method

Paul R. Woodward, a David H. Porter, a Sarah E. Anderson, b
B. Kevin Edgar, a Amitkumar Puthenveetil, a and Tyler Fuchs a

a Laboratory for Computational Science & Engineering,
University of Minnesota, 499 Walter, 117 Pleasant St. S. E.,
Minneapolis, Minnesota 55455, USA

b Cray Research, Eagan, Minnesota, USA

parallel computation; turbulence

1. Introduction

We have used the PPM gas dynamics code [1-4] on large NSF TeraGrid parallel systems as well as on SMP and cluster systems in our lab to simulate turbulent flows at grid resolutions of $1024^3$ and $2048^3$ cells [5-9]. We have studied both single- and two-fluid flows with statistically homogeneous and unstable shear layer initial conditions. The purpose of these simulations has been to capture detailed datasets that allow design and validation of subgrid-scale turbulence models. The parallel code implementation manages a set of shared data objects, each describing a subdomain of the problem. These data objects can be instantiated as either disk files or in-memory objects on a designated set of network nodes that serve them to computational processes over a cluster network. When implemented as disk files, large problems can be run, at no loss in performance, on systems that do not have sufficient memory to hold a complete description of the problem state. Such runs are also fault tolerant, because of the persistent disk image of the problem that is updated by the worker processes. Using either disk files or in-memory data objects, the number of participating CPUs can be dynamically adjusted during the run with no need to halt or pause the computation. This parallel computing method and the implications of our simulation results for subgrid-scale turbulence models is described below. Implications of our simulations for subgrid-scale turbulence models are also discussed.
2. New features of PPM used in this work.

Although several new features have been in use in our PPM codes that have not appeared in the published literature, a full description of the present version of the algorithm can be found on the LCSE Web site at www.lcse.umn.edu [4]. Such features that play particularly important roles in the simulations to be presented include the following: (1) estimation of the cell-averaged kinetic energy per unit mass from constrained linear distributions of all velocity components in all dimensions that are generated at the outset in each 1-D pass, and (2) estimation of the local smoothness of each variable to be interpolated with application of monotonicity constraints only when a variable is judged not to be smooth.

3. SHMOD parallel code framework.

The problem domain is decomposed into grid bricks in such a way that there are roughly 4 times as many such bricks as the maximum expected number of computing processes that will be available during the simulation. An order for updating these subdomains is established which minimizes the chance that any update process will have to wait upon the completion of previous updates in order to obtain necessary domain boundary data. This minimization is made under the assumption that updates require roughly the same amounts of time. Substantial variation in update time can occur with essentially no impact on the efficiency of code execution if there are sufficiently many subdomains. We find that 4 times the number of updating processes works well in practice, with no processes ever having to wait upon data, even if the previous time step has not yet been completed.

The subdomain updates are coordinated through a simple mySQL database. This database contains all necessary global knowledge, including the value of the time step, the status of each grid brick update, and the location of its data on the network. The database serializes access to this global data and makes it available to all participating processes, but it is otherwise passive. Processes can be started at any time. Once told the location of the database, they will check in, grab work assignments, and assist in the computation until either it is completed or they are killed. Processes can be killed at any time, in which case their assigned work is not completed. After a time-out interval, other processes take up this uncompleted work, so that the computation can proceed. If the killed process somehow wakes up, it will be aware from examination of its own clock that it has timed out, it will then abort the work it had begun, go to the database, and find new work to perform.

The key to this type of parallel computation is network bandwidth and the overlapping of data communication with computation. There are only two types of data communication used. First, of course, is database access. This is very simple. The grid brick update process simply reads the entire contents of the database, locking it, altering it, returning it, and unlocking it. Some accesses do not require locks, but
the database is so small that its entire contents, a single “unstructured glob,” can be transmitted over the network in a time comparable to that required to contact the database at all. Even with hundreds of participating processes, the database easily handles the traffic and is not a bottleneck. The second kind of communication is bulk data access. Here grid brick update processes either read or write back large data records (files) that describe the complete states of subdomains. This data is specially structured to permit transmission in a small number of large serial reads or writes, so that the transmission is very efficient. These grid brick data objects are implemented either in the memories of specially designated memory-server nodes or on the disks of specially designated disk server nodes (which may also participate in the computation). Large shared file systems attached to the network can also store and serve up these files very efficiently.

4. Homogeneous, Mach 1 turbulence simulation on a 2048³ grid.

As friendly users of the TeraGrid cluster at NCSA, we used from 80 to 250 dual-CPU nodes over a 2½ month period in the fall of 2003 to simulate compressible, homogeneous turbulence on a 2048³ grid with our PPM code (see [9] and www.lcse.umn.edu). We initialized this flow with smooth velocity disturbances centered on an energy-containing scale of half the width of our cubical, periodic problem domain. We put 10% of the initial kinetic energy into the compressional modes and the remaining 90% into the solenoidal velocity field. The initial disturbances were of course extremely well resolved. This flow was allowed to decay over a time interval equal to 2 flow-through times for the entire box, which in this Mach-1 flow was also 2 sound crossing times of the box. By the end of this time interval the decaying velocity power spectrum was no longer changing shape.

We have analyzed the data from this flow to assess the quality of a correlation we observed in earlier flows [6-8] between the rate of generation of subgrid-scale kinetic energy, which we write as $F_{SGS}$ below, and the determinant of the rate of strain tensor for the larger-scale flow. We follow the classic Reynolds averaging approach, in which we apply a Gaussian filter with a prescribed full width at half maximum, $L_f$, to our simulation data to arrive at a set of filtered, or “resolved,” variables and a set of “unresolved” state variables that fluctuate rapidly in space. For any particular state variable, $Q$, we define the filtered (or “resolved”) value, $\bar{Q}$, by

$$\bar{Q}(x) = \frac{\int e^{-(k_f(x-x))}\,Q(x)\,d^3x}{\int e^{-(k_f(x-x))}\,d^3x},$$

where the wavenumber of the filter, $k_f$, is related to the full width at half maximum, $L_f$, by $L_f = 1.6688/k_f$, and where the integral in the denominator is, of course, equal to $(2\pi)^{3/2}/(2k_f)^3$. The mass-weighted, or Favre, average of a state variable, $Q$, is denoted using a tilde. Manipulating the Euler equations using these
definitions in order to arrive at the time rate of change of the kinetic energy in a frame moving with the filtered flow velocity, we get:

\[
\frac{\partial k_{SGS}}{\partial t} + \mathcal{D}_j(\tilde{u}_j k_{SGS}) = \frac{Dk_{SGS}}{Dt} + k_{SGS} \mathcal{D}_j(\tilde{u}_j) = \left( p \frac{\partial}{\partial i} \tilde{u}_i - \tilde{\rho} \frac{\partial}{\partial i} \tilde{u}_i \right) - \tau_{ij} \frac{\partial}{\partial j} \tilde{u}_i - \frac{\partial}{\partial j} \left( \left( u_j p - \tilde{u}_j \tilde{\rho} - \tilde{\rho} \tau_{ij} + \frac{1}{2} \frac{\partial}{\partial j} \tilde{u}_i \right) + \frac{1}{2} \rho \tilde{u}_i^2 - \frac{1}{2} \rho \tilde{u}_i \tilde{u}_j \right)
\]

Here $k_{SGS}$ is the subgrid-scale kinetic energy, $D/Dt$ denotes the co-moving time derivative, and $\tau_{ij}$ is the subgrid-scale stress (SGS) tensor,

\[
\tau_{ij} = \rho u_i u_j - \tilde{\rho} \tilde{u}_i \tilde{u}_j
\]

Using our simulation data we can establish the relative importance of the various terms grouped on the right in the above expression for the time rate of increase of subgrid-scale kinetic energy per unit mass in the co-moving frame. This analysis indicates that, statistically, the divergence terms can be modeled by a diffusion of $k_{SGS}$ and that the first terms in brackets on the right, the $pDV$ work terms, tend to have little effect on the average. However, the term $-\tau_{ij} \frac{\partial}{\partial j} \tilde{u}_i$ has systematic behavior that tends to make it dominant over space and over time. We will refer to this term as the forward energy transfer to subgrid scales, or $F_{SGS}$. By analysis of

Figure 1. A diagonal slice through the computational domain with a thickness of 200 cells is shown here at time 1.25. The logarithm of the magnitude of the vorticity is visualized, with the highest values white, and then smaller values yellow, red, purple, blue, and black.
this and other detailed data sets, we have correlated this $F_{SGS}$ term to the topology of the filtered flow field, expressed in terms of the determinant of the deviatoric symmetric rate of strain tensor, given by:

$$
(S_D)_{ij} = \frac{1}{2} \left( \frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \nabla \cdot \tilde{u} \right)
$$

There is of course also a correlation with the divergence of this velocity field.

$$
FT_{MODEL} = A L_f^2 \bar{\rho} \det(S_D) + C k_{SGS} \nabla \cdot \tilde{u}
$$
This model equation is intended for use in a large eddy simulation in which the subgrid-scale kinetic energy, \( k_{SGS} \), is carried as an additional independent variable, so that it is available for use in the second, compressional term above. We find that outside of shock fronts the best fits for the coefficients \( A \) and \( C \) in the model are:

\[
A = -0.75, \quad C = -0.67
\]

The best fit coefficient for a term in the norm of the rate of strain tensor is zero.

The results of the above set of coefficients are shown on the previous page for data from our 2048\(^3\) PPM simulation at 2 times during the run and for 3 choices of the filter width (the model values are along the x-axes and the actual ones along the y-axes). The fits are very good in all cases, so that we are encouraged to construct a subgrid-scale turbulence model using this model equation. We note that if we combine the term in \( FT_{MODEL} \) involving the divergence of the velocity with the term \( k_{SGS} \nabla \cdot \vec{u} \) on the left-hand side of the evolution equation for \( k_{SGS} \), the combination of these terms indicates that this form of internal energy of the gas resulting from unresolved turbulent motions acts as if it had a gamma-law equation of state, with a gamma of 5/3. This is precisely the behavior under compression or expansion that results from the conservation of angular momentum in the case of an isolated line vortex, and consequently the value \(-0.67\) for our coefficient \( C \) is no real surprise. Dissipation of this turbulent kinetic energy, \( k_{SGS} \), into heat is not described by the Euler equations and therefore does not appear in the evolution equation for \( k_{SGS} \) derived from them. In the Euler simulation of decaying turbulence with PPM, this dissipation is provided by numerical error terms, but the rate of this dissipation is determined by the rate at which the well-resolved, inviscid turbulent cascade produces very-small-scale motions for PPM to damp. This dissipation rate can be extracted from analysis of the simulation data. Reasoning that a local value of \( k_{SGS} \) should decrease by a factor \( C_{decay} \) in one local SGS eddy turn-over time, we are led to the model that the contribution to \( Dk_{SGS}/Dt \) from this dissipation is, approximately,

\[
- C_{decay} (k_{SGS} / L_f) (2 k_{SGS} / \bar{\rho})^{1/2}
\]

A reasonably good fit to the data from the PPM simulation of decaying Mach 1 turbulence on the 2048\(^3\) grid is then obtained using a value of 0.25 for \( C_{decay} \). Using similar reasoning, we may suppose that local variations of \( k_{SGS} \) over one filter width should be reduced substantially by diffusion in one local SGS eddy turn-over time. This leads us to model the contribution to \( Dk_{SGS}/Dt \) from this diffusion, represented in our evolution equation for \( k_{SGS} \) by the bracketed group of divergence terms on the right, by

\[
- C_{diffuse} L_f (2 k_{SGS} / \bar{\rho})^{1/2} \nabla^2 k_{SGS}
\]

Our data shows that this model works fairly well with \( C_{diffuse} = 0.07 \) for all three of our filter widths.

To assess the value of these modeling ideas, we have implemented them in our PPM code. In an attempt to keep the model simple, we use an eddy viscosity mechanism, with a coefficient set from our model for \( F_{SGS} \) above, to bring about the exchange of energy, in either direction, between the resolved motions and the
unresolved SGS turbulence. We of course conserve total energy to machine accuracy, and therefore do not allow more energy to go into the resolved motions from the unresolved turbulence than is stored locally (in space and time) in $k_{SGS}$. As we have briefly described it here, the model has only a single adjustable parameter, the effective filter width, $L_f$, of our PPM gas dynamics scheme on a given mesh. Experimentation shows that setting this to 3 or 4 times the grid cell width produces noticeable effects upon the simulated flow. At these settings, the smallest eddies produced by PPM without such a turbulence model are absent, and a distribution of $k_{SGS}$ appears roughly in their place. Beyond this, dynamical effects of the model remain to be tested in detail.

Although a thorough analysis of the effects of the turbulence model briefly described above remains to be carried out, integration of this model’s evolution equation for $k_{SGS}$ along with the rest of the decaying turbulent flow from its initial conditions can reveal whether or not the ideas and statistical correlations that motivate this model are consistent with the data. We can determine $k_{SGS}$ from the original Euler simulation data on the 2048$^3$ mesh using filtering techniques, and then we can compare that spatial distribution with the result of a PPM computation with the turbulence model using a 256$^3$ mesh. The 256$^3$ mesh is sufficient to capture the energy containing scales in this flow, and therefore we might hope that using the turbulence model on this mesh we could arrive at a good approximation to the result for $k_{SGS}$ from the 2048$^3$ Euler simulation. These two spatial distributions are compared in Figure 3 in a thin slice through the simulation volume at a time of 1.1, which is about half way to the time when the turbulence is fully developed and its spectrum is not changing in shape. The comparison seems to be quite good.
The turbulence model described here in initial experiments appears to damp the smallest scales of motion that the PPM Euler scheme tends to produce, and then to replace them with a relatively smooth distribution of modeled subfilter-scale turbulent energy. It remains to be established that this combination of resolved and modeled motions is more dynamically faithful to the very high Reynolds number limit of viscous flows in any particular situation. In future work we will investigate this issue. It should also be interesting to compare this model with results of different modeling approaches, such as the dynamical models of Germano et al. [10], Moin et al. [11] or the subgrid-scale model of Misra and Pullin [12], all of which begin with difference schemes that, unlike PPM, have vanishing formal dissipation.

References