Three-Dimensional Simulations of Compressible Turbulence on High-Performance Computing Systems*

A. A. Mirin† R. H. Cohen† W. P. Dannevik† A. M. Dimits†
D. E. Eliason† D. H. Porter‡ O. Schilling† P. R. Woodward‡

Abstract

A three-dimensional hydrodynamics code based on the Piecewise Parabolic Method (PPM) is used to examine compressible fluid turbulence in three dimensions. The code runs on a number of parallel architectures, including $MPPs$ and SMP clusters. We consider problems of current interest, such as Rayleigh-Taylor and Richtmyer-Meshkov instability and turbulent mixing, and interactions of a shock with pre-existing turbulence. We present performance results on leading-edge platforms, including those supported under the DOE Accelerated Strategic Computing Initiative ($ASCI$).

1 Introduction

In many hydrodynamics applications, the relevant length scales range over several orders of magnitude, so that finite-difference direct numerical simulations (DNS) are computationally not feasible for the driving parameters of interest. To simulate the dynamically important range of scales, large-eddy simulations (LES) are performed instead, in which the dynamical effects of the unresolved scales are modeled by a subgrid-scale (SGS) parameterization, and the resolved scales are calculated explicitly. These parameterizations allow the use of fewer gridpoints than would be necessary for a direct numerical simulation. A principal goal of our LES research is to develop three-dimensional SGS parameterizations from hydrodynamic theory and experiments and to validate them against fully resolved direct numerical simulations and available experimental data.

This research depends on the availability of high-performance computing resources. To resolve some of the smallest spatial scales requires a thousand or more meshpoints in each of the three directions, which translates to billions of computational zones. As the mesh becomes finer, the simulation timestep often becomes commensurately smaller, so that a doubling of the mesh resolution in each direction translates to a factor-of-16 increase in computer time. With the increase in computational requirements comes a commensurate increase in data assimilation requirements, with data files often in the tens of gigabytes range.

2 Three-Dimensional Compressible Turbulence Simulations

Our focus is on three-dimensional fluid dynamics computations. A number of theories have already been developed for two-dimensional turbulence. However, turbulence in two

*This is LLNL Report Number UCRL-JC-125949. Work performed by LLNL and UMN under U.S.D.O.E. Contract W-7405-ENG-48, and by UMN under U.S.N.S.F. Grand Challenge Grant ASC-9217394.

†Lawrence Livermore National Laboratory, Livermore, CA.

‡University of Minnesota, Minneapolis, MN.

and three dimensions is profoundly different. For example, in a two-dimensional flow, small-scale fluctuations can often coalesce into larger-scale structures, so that energy is transferred from small scales to larger scales. In three dimensions, large-scale structures tend to break up into smaller structures, with a transfer of energy from the large scales to smaller scales. This has important consequences for both simulations and physics. In two-dimensional simulations of two-dimensional physics, fluctuations on resolved scales can often be accurately captured as long as there is some dissipation at the smallest scales. However, in two-dimensional simulations of three-dimensional physics, the average effects of three-dimensionality are calculated separately and incorporated as a transport process in two-dimensional space.

In three-dimensional DNS, 3-D turbulence can be directly simulated, but only over a limited range of length scales. Even if the initial fluctuations are restricted to resolved scales, the turbulence dynamics can induce transfer of energy to length scales that are too small to resolve. It is thus essential for physics reasons, and can also be important for numerical reasons, to incorporate a subgrid-scale parameterization that adequately approximates the effect of these smaller unresolved length scales on the larger resolved scales.

Another important consequence of the difference between two- and three-dimensional turbulence dynamics is that parameterizations developed to characterize the former are likely to be inadequate for the latter. The parameterizations that are accurate for three-dimensional LES will in general be different from those that are accurate for two-dimensional transport models, as they must represent the effects of a different degree of averaging in the third direction.

3 Numerical Approach

We are using a numerical simulation code based on the Piecewise Parabolic Method (PPM), which is a higher-order accurate Godunov method developed by Colella and Woodward [1]. The Godunov approach is typical of standard numerical techniques in regions where the solution is smooth. However, in regions with discontinuities, such as strong shocks, the Godunov method approximates the solution well by analytically solving an associated Riemann problem. This is an idealized problem describing the evolution of a simple jump into shocks and/or rarefactions, with a contact discontinuity in between. Monotonicity constraints ensure that these discontinuities remain sharp and accurate as they traverse the computational grid. The higher-order spatial interpolation in the PPM allows steeper representation of discontinuities, allowing a more accurate solution to a wider class of problems. The particular PPM implementation that we are invoking uses a Lagrangian time-advance followed by a remap onto the original grid, making the calculation effectively Eulerian. Multidimensional aspects are handled through operator-splitting in the coordinate directions, with the splitting order alternating on consecutive timesteps to attain a higher order of accuracy. For some of our simulations, molecular dissipation processes are explicitly modeled, in which case the simulations are of the Navier-Stokes equations rather than the Euler equations.

4 Programming Model

We are using a version of the PPM code that is targetted toward distributed memory, message-passing massively parallel processors. (A version written for SMP clusters has recently been developed.) Parallelization is accomplished through domain decomposition with message-passing. The decomposition is fully three-dimensional and logically rectangular. The subdomains contain extra border cells to allow intermediate (and redundant) computation so as to reduce the required interprocessor communication. The calculation is computationally intensive, with over 2700 floating point operations per grid point per timestep being carried out. Variables are ordered in memory to effect optimal usage of cache. Because the difference scheme is fundamentally explicit, the only required communications are across cell borders. All of the communications are decomposed into one-dimensional shifts and are thus easily adaptable to new architectures.

5 Data Assimilation

Given a well-resolved flow (i.e., a 512^3 mesh or larger), this code has demanding I/O and data management requirements. Several types of data files are produced. In addition to restart dumps, which must represent the data in its entirety, we work with several levels of compression; most commonly, each data element is packed into a 2-byte integer portable binary format. Each node produces its own data file, thereby facilitating parallel I/O. The files corresponding to the various nodes are tagged according to the coordinates of the corresponding subdomain in the decomposition.

Data is analyzed using the *PPM* toolkit developed by the University of Minnesota and collaborators [2]. The *a3d* program is used to compute physically relevant quantities, such as vorticity or power spectra, from the individual nodal files. These quantities can be further reduced, for example, into profile or spectral data, and converted to ascii output. Alternatively, they can be mapped into a single-byte representation, where the mapping, which is typically nonlinear, is designed to result in an informative raster image. This single-byte data representation is known as *Bricks of Bytes*, or *BOB* format. Any of several visualization tools may then be used to represent the *BOB* data. For example, the *Perpath* program performs volume rendering and is useful for producing animations. A newer tool, known as *Bob*, is designed more for interactive viewing on SGI systems.

6 Code Performance

We have run the *PPM* code on a number of parallel processing platforms, including the *ASCI* Blue-Pacific *IBM-SP* at Lawrence Livermore National Laboratory (LLNL), the *Cray-T3D* at LLNL, and the Intel Paragon at Sandia National Laboratories. We first consider a triply periodic decay problem having 128 meshpoints in each direction. A $4\times4\times2$ domain decomposition is invoked, so that the local mesh has resolution $32 \times 32 \times 64$. The SP and *T3D* computations each use 64-bit arithmetic, whereas the Paragon case uses 32-bit arithmetic. A comparison of throughput is shown in Table 1.

TABLE 1

Intermachine Comparison

LLNL *IBM-SP* (32 nodes)	3.3 μs/Δt/point	25.7 Mflops/node
LLNL *Cray-T3D* (32 processors)	6.4 μs/Δt/point	13.2 Mflops/proc
Sandia Intel Paragon (32 nodes)	21.9 μs/Δt/point	3.9 Mflops/node

It is to be emphasized that normalizing with respect to the number of processing elements is arbitrary. Some machines were designed to have slower nodes, but in greater numbers. For example, the LLNL *IBM-SP* has 256 nodes, each with a peak throughput of 266 Mflops, whereas the Sandia Paragon has 1840 nodes, each with a peak throughput of

75 Mflops. Also, this case produced no large data files, and hence did not test the relative performance of the I/O subsystems. One would expect that writing large amounts of data to disk would cause a greater degradation of performance on the $T3D$ than on the other two platforms because of having to go through the front end on that architecture. Indeed, in a separate animation computation we observed almost no throughput degradation on the IBM-SP, whereas the $T3D$ slowed by close to 40 percent.

We also ran this calculation on the Sandia Intel Tinyflop machine, a preliminary version of their $ASCI$ Red computer. We have observed continual improvement of that new platform but choose not to cite results here because of the rapidly evolving performance.

We next consider parallel efficiency on the Blue-Pacific system. A comparison with respect to domain decomposition is shown in Table 2.

TABLE 2
Parallel Efficiency on IBM-SP

8 nodes (2x2x2 decomp; local 64x64x64 mesh)	10.5 μs/Δt/point
32 nodes (4x4x2 decomp; local 32x32x64 mesh)	3.3 μs/Δt/point
128 nodes (4x4x8 decomp; local 32x32x16 mesh)	1.2 μs/Δt/point

We see that, for this relatively coarse problem (128^3 zones), the relative parallel efficiency in going from 8 to 32 nodes is 79 percent, and from 32 to 128 nodes is 71 percent. The drop in parallel performance with respect to the size of the domain decomposition is due almost entirely to the redundant border computations. Recall that each subdomain is surrounded by seven border rows, with redundant computations being performed in lieu of additional communications. We expect higher-resolution cases to execute with greater parallel efficiency.

7 Rayleigh-Taylor Instability and Turbulent Mixing

We consider two applications. The first of these involves using the PPM code to simulate the Rayleigh-Taylor instability and turbulent mixing in a compressible fluid. The Rayleigh-Taylor instability occurs, for example, when a light fluid is trying to support a heavier fluid. Applications of the Rayleigh-Taylor instability include the overturn of the outer portion of the collapsed core of a massive star, and the laser implosion of deuterium-tritium fusion targets [3]. The instability ensues as soon as a small perturbation occurs at the fluid interface. Those portions of the lighter fluid that are higher than average will accelerate further into the heavier fluid, and vice-versa, causing the perturbation to grow. The heavier fluid, as it drops into the lighter fluid, will form spikes, and the lighter fluid, as it rises into the heavier fluid, will form bubbles.

This application is a direct numerical simulation where the domain is a unit cube spanned by a grid containing 512 points in each of the three directions. High resolution is needed in order to validate potential SGS closure models. This case was run on the $ASCI$ Blue-Pacific System at LLNL using 128 nodes. The initial equilibrium state consists of a $\gamma = 5/3$ gas, in which each of the subvolumes above and below the midplane ($z = 0.5$) are in hydrostatic equilibrium. The internal energies are piecewise constant, while the density and pressure decrease exponentially with height, but have different scale heights above and below the midplane. The density has a jump from 1 just below, to 2 just above the midplane, corresponding to an Atwood number of 1/3, and the pressure is continuous across the midplane. The sound speed corresponding to the equilibrium state below the midplane is 1.0. Other parameters are Prandtl number = 1.0 and viscosity (normalized to the sound speed and box size) = 0.00004. The boundaries are periodic in the horizontal directions and impenetrable in the vertical direction. Time is in units of sound-wave transit times (below the interface).

A random spectrum of low-level velocity perturbations away from the equilibrium state is initially imposed. Figure 1 shows the temperature field at times $t = 1.0$, 2.0, 3.0, and 4.0, respectively. After the initial linear mixing phase, bubbles (rising from below) and spikes (falling from above) begin to form. Afterward, the horizontal fluctuation scales grow in size and the physical system evolves toward a stably stratified equilibrium.

A 301-frame animation of this turbulence simulation was produced and first shown at the *ASCI* Blue-Pacific dedication cermoney on 25 October 1996. The case ran for 5 time units and required roughly 84 hours of wall time. Each frame (stored in *BOB* format) required 134 MB of data, for a total of over 40 GByte. We produced 33 restart dumps as well, each requiring 8.3 GByte. In all, over 300 GByte of data were written to disk.

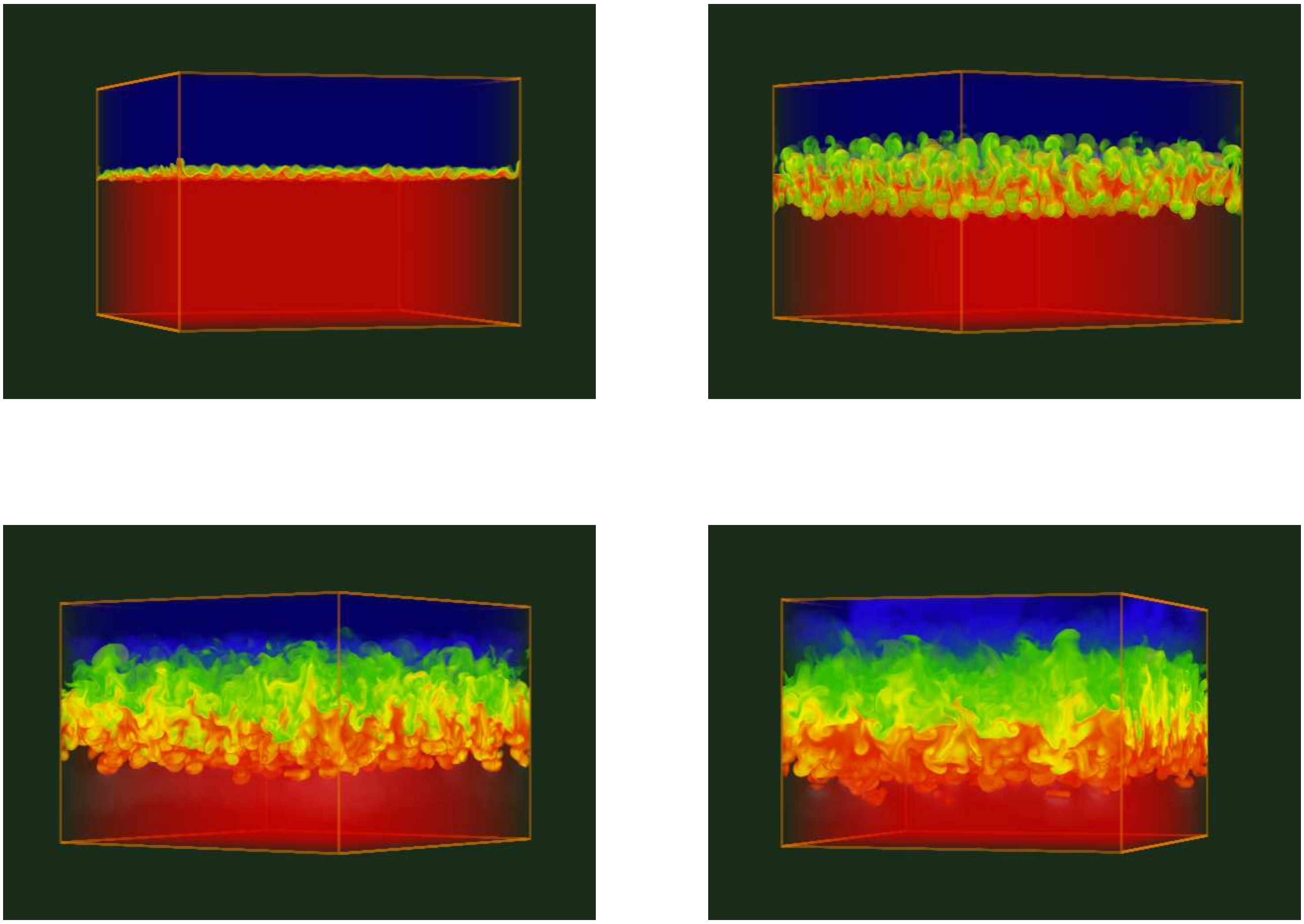

FIG. 1. *Temperature at t=1.0 (upper left), t=2.0 (upper right), t=3.0 (lower left), and t=4.0 (lower right)*

8 Shock-Turbulence Interaction

We have also used the *PPM* code to study the interaction of shock waves with a pre-existing three-dimensional turbulent field in a compressible fluid. Such a situation can arise, for example, when multiple shocks pass through an interface of different-density materials. The first shock can produce a Richtmyer-Meshkov instability; the resultant turbulence interacts with subsequent shocks. Of interest is how the shock affects the turbulence — in particular the turbulence strength, spectrum, anisotropy, and rate of shock propagation.

We first run a 3-D decay problem with triply periodic boundary conditions, starting from an initial Gaussian spectrum of 8 modes per direction with a width of 2 modes [4]. The initial turbulent Mach number is 0.7, and the initial pressure and density perturbations are zero. This initial state is allowed to decay to a turbulent Mach number of about 0.2, at which time inflow boundary conditions are invoked. At one end ($z = 0$), we specify a density, pressure, and z velocity consistent with downstream conditions for a shock of specified Mach number moving into a fluid at rest (the final state of the decay portion of the simulation). At the other end ($z = 1$), reflecting boundary conditions are imposed. The boundary conditions in the x and y directions remain periodic. As the simulation proceeds, the shock forms, moves from $z = 0$ to $z = 1$, and reflects. At any instant prior to reflection, there is an unshocked turbulent region between the shock and $z = 1$, a shocked turbulent region extending downstream from the shock, and finally a region filled with quiescent fluid that moves in from the inflow boundary. All results described below are for 256^3 resolution and are for pure *PPM* Euler (no explicit dissipation) simulations.

Figure 2 shows a 3-D rendering of the v_z field (with the x-y average subtracted off). The shock location, and the post-shock amplification, are evident. Figure 3 shows z-profiles of the root-mean-square x-y averages of each vorticity component, for Mach 2 and Mach 6 shocks. From this figure we note that (a) the x and y vorticities increase immediately behind the shock, (b) the z vorticity initially decreases behind the shock, but subsequently recovers and increases, and (c) the Mach 6 effects are appreciably stronger. Comparison with 256^3 and 128^3 resolution simulations (not shown) indicates that the simulations resolve a portion of the inertial range. The upstream data is consistent with a $k^{-5/3}$ inertial-range spectrum. Finally, there is the question of the influence of the turbulence on the shock; we find, in agreement with the analysis of [5], that the shock is (slightly) sped up by the presence of the turbulence, and the shock front is broadened by the turbulence.

9 Conclusions and Future Directions

We have carried out medium-to-high resolution *PPM* compressible turbulence simulations on present-day high-performance computing platforms. A main thrust area of this research will be to derive subgrid-scale closure models, and to validate those models with direct numerical simulations that will require 1000 or more gridpoints in each direction. Thus, a second main thrust area will be to make effective use of the most advanced computing resources at our disposal. This will undoubtedly involve heirarchical architectures, such as *SMP* clusters, as well as massively parallel processors. It will also require, in addition to high-speed processing, advanced storage and visualization capabilities.

References

[1] P. Colella and P. R. Woodward, *The Piecewise Parabolic Method (PPM) for Gas-Dynamical Simulations*, J. Comput. Phys., 54 (1984), pp. 174–201.

[2] D. H. Porter, *Perspective Volume Rendering*, U. Minnesota Supercomputer Institute Research Report UMSI 91/149, (1991).

[3] D. H. Sharp, *An Overview of Rayleigh-Taylor Instability* , Physica, 12D (1984), pp. 3–18.

[4] D. H. Porter, A. Pouquet and P. R. Woodward, *Kolmogorov-Like Spectra in Decaying Three-Dimensional Supersonic Flows*, Phys. Fluids A, 6 (1994), pp. 2133–2142.

[5] S. K. Lele, *Shock-jump Relations in a Turbulent Flow* , Phys. Fluids A, 4 (1992), pp. 2900–2905.

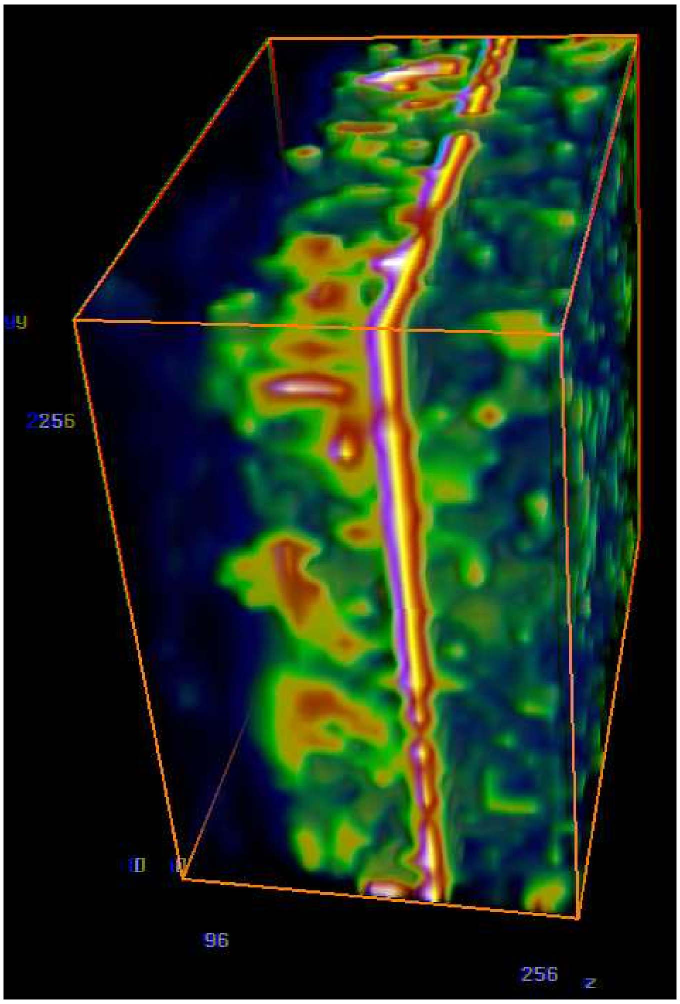

Fig. 2. *Velocity field in the z direction*

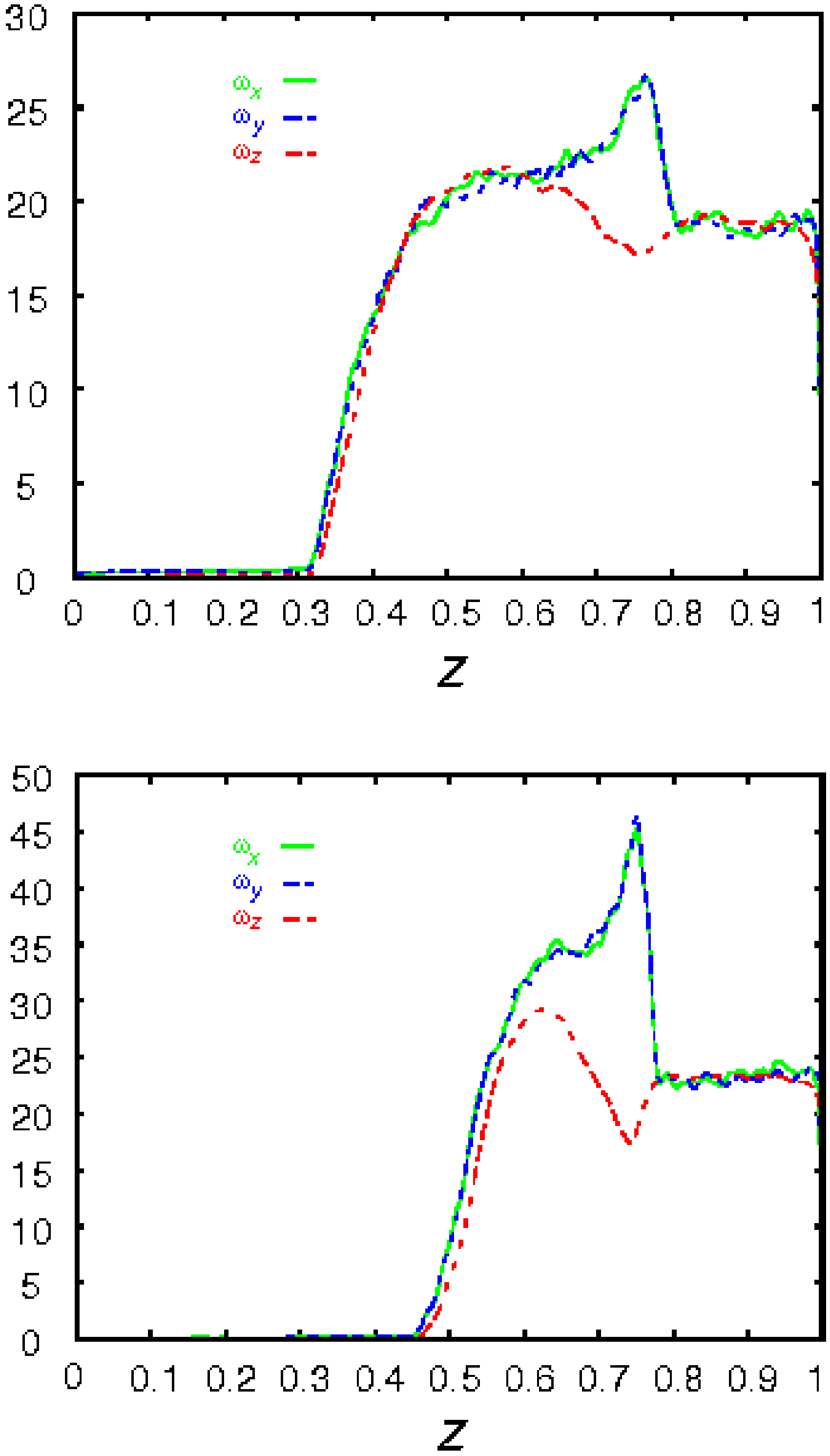

FIG. 3. *Vorticity for Mach 2 (top) and Mach 6 (bottom) cases*