A Parallel 2-D Explicit-Implicit Compressible Navier–Stokes Solver

C. B. Velkur, J. M. McDonough
Department of Mechanical Engineering
University of Kentucky, Lexington, KY 40506-0503
Email: *vcbabu@uky.edu*

Key Words: Hybrid methods, Shock-tube problem, OpenMP

1 Introduction

Time-dependent PDEs are generally solved using time-marching procedures. Typical time-marching procedures fall into one or the other of two different approaches, explicit or implicit. The time step in an explicit scheme is governed by the Courant number for high Reynolds number problems, which must not be greater than unity for stable calculations. The stability limit for explicit schemes is set by regions in the domain where wave speeds are high. These regions drastically reduce the time step possible for explicit schemes. Implicit schemes, on the other hand, can maintain stability with much larger time steps when compared to their explicit counterparts. For coupled nonlinear PDEs such as the compressible Navier–Stokes (N.–S.) equations, the use of implicit schemes results in having to iteratively solve a coupled system of linearized equations at each time step. Hence, a reduction in the number of time steps may be outweighed by increase in the number of arithmetic operations at each time step. Hybrid schemes containing both implicit and explicit approaches have also been developed to abate the disadvantages of the above mentioned approaches. In 1981 MacCormack [1] presented an explicit-implicit predictor-corrector method that involved inversion of bidiagonal matrices in an effort to reduce computer time. An implicit-explicit hybrid scheme was developed by Fryxell *et al.* [2] which extended Godonov-type schemes to the implicit regime. An iterative implementation of the scheme in [2] was done by Dai *et al.* [3] for solving the Euler equations.

Our solver is based on a predictor-corrector methodology. The predictor part of the scheme consists of a half time step explicit foward Euler time integration of the Euler equations followed by a full time step implicit backward Euler time integration of the complete N.–S. equations. Spatial discretization is second-order centered for both predictor and corrector parts of the scheme, with dependent variables being evaluated at cell centers and fluxes evaluated at cell walls. The nonlinearites in the N.–S. equations are handled iteratively by δ-form quasilinearization, and δ-form Douglas–Gunn time splitting [4] is used to solve the linearized equations leading to an easily parallelizable algorithm. Shocks are captured by using MacCormack and Baldwin higher-order artificial viscosity as given in [5].

Our ultimate goal is to have 3-D capability including turbulence effects in the context of large-eddy simulation methodology, which will be highlighted at the time of the oral presentation at the conference. Moreover, in calculations of high-Re viscous flows, where changes in the flow fields occur close to a surface, finer gridding is required to capture these effects, especially the boundary layer. Though LES significantly reduces the amount of arithmetic when compared with DNS, the required arithmetic still can scale as badly as Re^2. This in turn results in very long run times, and hence the need for using parallelizable alogorithms for such simulations.

We introduce the governing equations in the next section of the paper, followed by a brief description of the standard test problem, the shock tube employed to validate the solver. Finally we present numerical solutions and the speedups obtained by parallelization.

2 Governing Equations

Since our solver is of a hybrid type having a predictor step solving the Euler equations and a corrector step solving the full compressible N.–S. equations, we will write both the systems of equations in their generic form given by

$$\frac{\partial U}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}=0. \tag{1}$$

Equation (1) represents the entire system of governing equations in conservation form if *U, F, G* are interpreted as column vectors given by,

$$U=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho\left(e+\frac{\mathbf{V}^2}{2}\right)\end{bmatrix}, \tag{2}$$

$$F=\begin{bmatrix}\rho u\\ \rho u^2+p-\tau_{xx}\\ \rho vu-\tau_{xy}\\ \rho\left(e+\frac{\mathbf{V}^2}{2}\right)u+pu-k\frac{\partial T}{\partial x}-u\tau_{xx}-v\tau_{xy}\end{bmatrix}, \tag{3}$$

$$G=\begin{bmatrix}\rho v\\ \rho uv-\tau_{yx}\\ \rho v^2+p-\tau_{yy}\\ \rho\left(e+\frac{\mathbf{V}^2}{2}\right)v+pv-k\frac{\partial T}{\partial y}-u\tau_{yx}-v\tau_{yy}\end{bmatrix}, \tag{4}$$

where ρ, p and T are the ususal density, pressure and temperature; $\mathbf{V}\equiv(u,v)^T$ is the velocity vector; the quantity $\left(e+\frac{\mathbf{V}^2}{2}\right)$, which will be represented as E in the remainder of the report, corresponds to total energy. Elements of the stress tensor are given by:

$$\tau_{ij}=\mu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)+\delta_{ij}\lambda\frac{\partial u_k}{\partial x_k}, \tag{5}$$

Finally, k is thermal conductivity; μ is dynamic viscosity; λ is second viscosity, and δ_{ij} is the Kronecker delta. The corresponding Euler equations are obtained by setting $\mu, \lambda, k\equiv 0$.

3 Test Case

We validate our scheme using Sods shock-tube problem [6]. The shock-tube problem is an interesting test case because the exact time-dependent solution is known for the Euler equations in 1-D, and hence we can compare our computed viscous solution at least qualitatively to the exact inviscid solution. The initial data for the shock-tube problem are composed of two uniform states (generally known as left and right states) separated by a discontinuity, physically a diaphragm. When the diaphragm is broken, two pressure waves appear, being either a shock wave or expansion fan, which start to run into the initial fluid states as shown in Fig. 1, resulting in two uniform states 2 and 3. In addition 1 represents the shock wave and 4 represents the expansion fan. The final states 2 and 3 are separated by a contact surface (discontinuity in first derivatives), which means that the pressure and the velocity of these states are equal, but density jumps across the discontinuity. The initial conditions in the left and right sections of the shock tube are given in Table 1 with all entries in SI units. The governing equations (1) are solved on a domain $\Omega=[0,L]\times[0,W]\equiv(0,1.0m)\times(0,0.2m)$, with boundary conditions; the no-slip condition is imposed at $y=0$ and $y=W$, and an inflow condition $\partial U/\partial n=0$ and outflow condition $\partial U/\partial n=0$ are applied at $x=0$ and $x=L$, respectively.

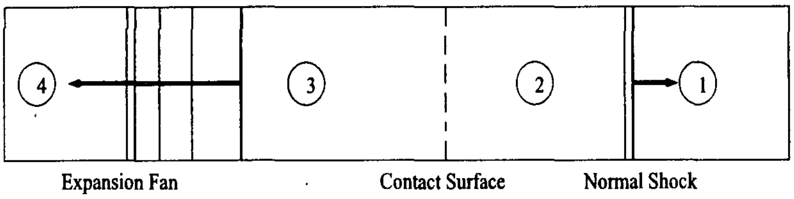

Figure 1: Shock tube.

Parameters	Left	Right
Velocity	0	0
Density	1	0.125
Pressure	10000	1000
Total Energy	2.5e5	2.5e4

Table 1: Initial conditions

4 Results

To compare our 2-D results with the 1-D case the density and velocity profiles at the horizontal centerline of the domain were used. Calculations reported here were performed on a 401 × 401 grid and a time step Δt= $1 \times 10^{-6} s$. The following figures suggest this is adequate, but further grid function convergence tests are being conducted. In Fig. 2 we present comparisons between computed and exact solutions. Part (a) of the figure shows the density profile, and part (b) displays the velocity profile. Both computed profiles are in good agreement with the exact inviscid solution. It should be noted that the inviscid discontinuities are now transformed to sharp but continuous variations due to physical viscosity and heat conduction effects, but these cannot be resolved using typical gridding; hence, arificial dissipation still must be used to capture these discontinuities.

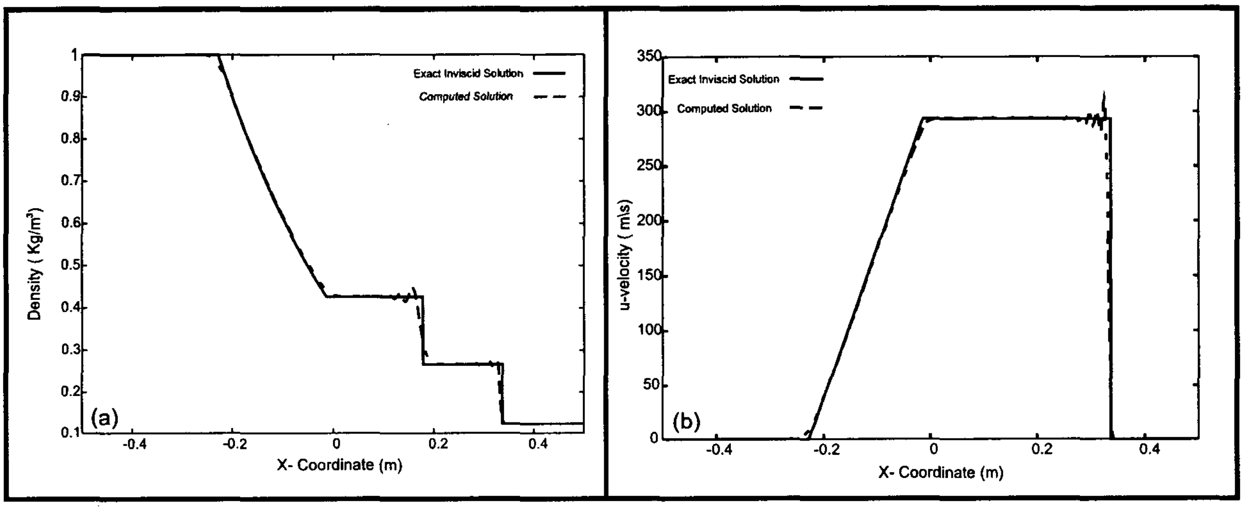

Figure 2: Exact vs computed results

Figure 3a presents a contour plot of velocity magnitude variation in the x-direction showing the final states at t = 0.61ms. In Fig. 3b we display a zoomed vector plot of the computed boundary-layer velocity profile. It is clear that the vertical grid spacing is sufficient to provide reasonably good resolution of the boundary-layer.

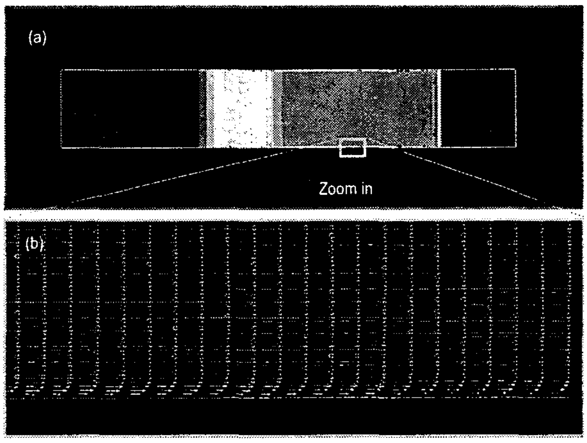

Figure 3: (3a) Velocity contour in x- coordinate direction: (3b) Zoomed boundary-layer velocity profile

The code was parallelized using OpenMP on the HP Superdome SMP at the University of Kentucky. The Do loops of the Douglas–Gunn time splitting were parallelized by issuing compiler directives such that the line solves were distrubuted among the processors. To study the speedup obtained, we used a range of number of processors from 2 to 32 to execute the parallel algorithm. Results of the speedup are presented in Fig. 4. It is should be noted that the speedups are sub-linear and not especially good; moreover, no improvement was seen in going to 32 processors, so these results are not shown. These results are fairly consistent with those obtained from several other pieces of software parallelized with OpenMP on Hewlett Packard hardware.

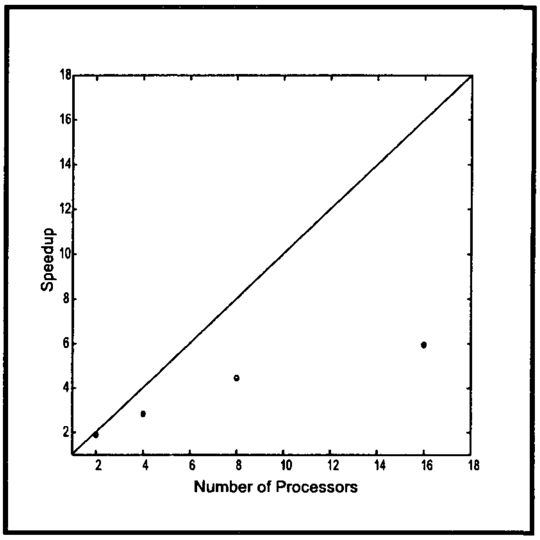

Figure 4: Speedup

5 Summary

A hybrid N.–S. solver with an explicit Euler equations predictor and an implicit corrector solving the full viscous N.–S. equations was introduced. We demonstrate relatively good accuracy between computed and exact inviscid solutions for Sods shock tube problem. We are able to resolve the boundary-layer profile upon using sufficiently fine grids. We remark that employing time-splitting methods such as Douglas–Gunn to solve multi-dimensional problems results in solution algorithms that are easily parallelizable. Finally, we presented the speedups obtained by parallelization using OpenMP. We note that the speedups are sublinear and ongoing tests are expected to produce some improvement, but as noted above the current speedups are not very different from earlier ones obtained with OpenMP.

References

[1] R. W. MacCormack, A numerical method for solving the equations of compressible viscous flows, *AIAA Paper.* **81-0110**, 1981.

[2] B. A. Fryxell, P. R. Woodward, P. Colella, and K. H. Winkler, An implicit-explicit hybrid method for Lagrangian hydrodynamics, *J. Comput. Phys.* **63**, 283–318, 1986.

[3] Wenglong Dai and Paul Woodward, Iterative implementation of an implicit-explicit hybrid scheme for hydrodynamics, *J. Comput. Phys.* **124**, 217–229, 1996.

[4] J. Douglas, Jr. and J. E. Gunn, A general formulation of alternating direction implicit methods, part 1. parabolic and hyperbolic problems, *Numer. Math.* **6**, 428–453, 1964.

[5] Charles Hirsch, *Numerical Computation of Internal and External Flows, Vol.* **2**, *John Wiley & Sons*, 1988.

[6] G. A. Sod, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, *J. Comput. Phys.* **27**, 1–31, 1978.