Parallel program complex for 3D unsteady flow simulation

E.V. Shilnikov

Institute for Mathematical Modeling RAS, 125047, Miusskaya sq. 4A, Moscow, Russia
shiva@imamod.ru

Abstract. A parallel program complex for 3D viscous gas flow simulation is presented. This complex is based on explicit finite difference schemes, which are constructed as an approximation of conservation laws (control volume method) and oriented on use of locally refined grids. Special algorithm and utility for nested grid partitioning was created. The principle of program construction permits to introduce new types of boundary conditions and change as finite difference scheme as governing equation system. Introducing new face types and writing new subroutines for flux calculations may reach it. This opens wide perspectives for the further development of program complex presented. The scalability of program complex was investigated on 2D and 3D subsonic and supersonic problems. The calculations were held on different cluster type multiprocessor computer systems. The parallelization efficiency was more than 90% for 40 processors and more than 60% for 600 processors.

1 Introduction

Essentially unsteady and turbulent regimes of viscous gas flows have received increasing attention from researches, motivated in part from the importance of unsteadiness in industrial problems arising in turbomachinery and aeronautics. Unsteady flow phenomena which occur frequently behind relatively slender, bluff structures are of great practical interest. In the case of symmetric geometry at relatively small Reynolds numbers numerical simulation based on 2D unsteady Navier - Stokes equations is quite successful. At high Reynolds numbers, which are more relevant in practice, 3D stochastic turbulent fluctuations are superimposed on the quasi-periodic 2D unsteady motion. So numerical simulation must be three-dimensional even for simple flow geometry. The numerical simulation of a detailed structure of unsteady viscous compressible 3D flows with high Reynolds numbers is possible only by use of high performance parallel computer systems. This demands the development of the specialized parallel software. This software must have a good scalability (with respect as to the number of processors as to the problem size), portability and robustness.

2 Choice of Numerical Method

Use of parallel computer systems with distributed memory architecture determines the choice of numerical method. The opinion is widely spread that we have to use only implicit schemes for viscous gas flow simulation because of their good stability properties. In the case of essentially unsteady flow we have to receive detailed information about high frequency oscillations of gas dynamic parameters. This fact limits the time step acceptable by the accuracy requirements. For many interesting problems these limitations neutralize the advantages of implicit schemes. So for such problems the explicit difference schemes seem to be preferable because of their simplicity for program realization, especially for parallel implementation. Our program complex is based on explicit finite difference schemes, which are constructed as an approximation of conservation laws (control volume method). The explicit kinetically consistent finite difference (KCFD) schemes [1] were selected for realization. They have soft stability condition ($\tau \sim h$) giving the opportunity to use very fine meshes to study the fine flow structure. KCFD schemes belong to the class of kinetic schemes. Nowadays the kinetic or Boltzmann schemes are very popular in computational fluid dynamics [2] – [4]. The successful experience in solving various gas dynamic problems by means of KCFD schemes showed that they describe viscous heat conducting flows as good as schemes for Navier-Stokes equations, where the latter are applicable. In addition to this KCFD schemes permit to calculate oscillating regimes in super- and transonic gas flows, which are very difficult for modeling by means of other algorithms.

The situations frequently occur during numerical modeling a large amount of modern problems of mathematical physics, when high accuracy resolution of the solution particularities in small regions is needed. These peculiarities may be the result of as physical processes as problem geometry. This problem may be avoided by use of unstructured meshes, but the convenience and simplicity of difference schemes on regular grids enforced us to use multiblock grids where different subregions have their own grids. A variant of such approach, the use of nested (or locally refined) grids is chosen. Parallel realization of explicit finite difference schemes for the numerical solution of gas dynamic problems on nested grids was discussed in [5].

3 Parallel Implementation

The basic idea for the program complex constructing was to simplify ultimately the parallel program providing its lucidity. For this reason all complicated logical communications and other sophisticated but not time-consuming operations are addressed to sequential preprocessing utilities as far as possible. The program complex structure is defined by four relatively independent tasks:

- problem/model formulation including description of computation field geometry, initial and boundary conditions specification and grid generation;
- grid partitioning and data preprocessing;
- main computational block execution;
- data postprocessing.

As a rule only third of these tasks must be parallel. Separate sequential programs may execute all other jobs.

Parallel realization is based on geometry parallelism principle. In the case when the difference scheme is written in the form of conservation laws, the approximation of gas dynamic equations comes to approximation of conservative variables (density ρ, momentum $\rho\mathbf{U}$ and total energy E) fluxes through cell faces. In order to reach the algorithm homogeneity the boundary conditions of different types (no slip, symmetry, impermeability, inlet, outlet conditions etc.) are also written as fluxes of conservative variables through region bound. Each cell face is supplied by an attribute indicating its type: inner face, various boundary faces, face between cells of different size (result of local mesh refining), ghost face i.e. face between ghost cells. This attribute determines which subroutine must calculate fluxes through the face. The face attributes as well as description of problem geometry and grid information are contained in a special text file, which is prepared by sequential preprocessing utility. Another utility divides 3D computational region with rectangular bounds (in i-j-k space) into required number of rectangular subdomains according to multistep algorithm described in [5]. As a result this utility creates a text file describing 3D subregions in terms of grid node numbers, list of neighbors for each subregion and information needed for organizing of inter processor communications.

According to this the main computational module of parallel program calculates fluxes through all faces in a subregion addressed to each processor. Having equal number of nodes in each subdomain the homogeneity of algorithm automatically provides load balancing of processors. Note, that such program construction permits to introduce new types of boundary conditions and change as finite difference scheme as governing equation system (changing the coordinate system for example).

4 Program Package Testing

The program complex was tested on different 3D subsonic and supersonic problems. Locally refined computational grids were used. Total amount of grid cells was about 8000000. Our results are in a good agreement with the experimental and computational results of other authors. The scalability of program complex was investigated on different cluster type multiprocessor computer systems (768-processor MCS-1000M computer system equipped with 667MHz 64-bit 21164 EV67 Alpha processors and 906-processor MCS-15000 computer system equipped with 2.2GHz PPC970FX processors). The parallelization efficiency was measured for different number of processors. The results can be found in Table 1.

Because of lack of memory such a large problem can't be solved on one processor of MCS-1000M system. That's why the efficiency for this system was computed with respect to the calculations held on 2 processors.

The scaling with respect to the problem size was also inspected. Our grid was doubled in each direction, so total number of cells achieved $\sim 6 \times 10^7$. Such problem is too large even for one processor of MCS-15000 system, so the times

Table 1. Efficiency (%) dependence on the number of processors

N	1	2	10	40	160	320	600
MCS-1000M	—	100	97	92.8	79.9	70.1	61.4
MCS-15000	100	98.5	96.3	92	80.1	72.1	62.7

were compared needed for 2000 time steps of our program on these two grids. If the efficiency doesn't depend on the problem size, these times must differ by the factor of 8 from each other. The diminishing of this factor corresponds to the efficiency increase for large problem. The values of this factor for some numbers (N) of processors are presented in Table 2.

Table 2. Computational time increase for enlarged grid

N	10	40	100	200	400	600
f	7.96	7.83	7.71	7.48	7.13	6.69

These results show that for $N < 100$ our factor is really close to 8, but for greater N it diminishes. This effect is connected with the increase of computational time with respect to exchange time for each processor. So, increasing of the total grid nodes number leads to the efficiency growth. For example factor 6.69 for $N = 600$ means 75% efficiency for "large" problem in contrast with 62.7% for "small" one.

This work was supported by Russian Foundation for Basic Research (grants No. 06-01-00187, 05-07-90230, 05-01-00510)

References

1. B.N. Chetverushkin: On improvement of gas flow description via kinetically-consistent difference schemes. In B.N. Chetverushkin et al (Eds): Experimental modeling and computation in flow, turbulence and combustion, Wiley, **2** (1997) 27–37
2. B. Perthame: The kinetic approach to the system of conservation laws. Recent advances in partial differential equations, Res. Appl. Math. Masson, Paris. **30** (1992)
3. E. Oñate and M. Manzam: Stabilization techniques for finite element analysis for convective-diffusion problem. Publication CIMNE **183** (2000)
4. S. Succi: The lattice Boltzmann equations for fluid dymanics and beyond. Oxford, Clarendon press. (2001)
5. E.V. Shilnikov: Viscous gas flow simulation on nested grids using multiprocessor computer systems. In: Proceedings of Parallel Computational Fluid Dynamics Conference (Moscow, Russia, 2003), Elsevier Science BV. (2004) 110–115